A forest is more than its trees: haplotypes and ancestral recombination graphs

Halley Fritze[†], Nathaniel Pope[*], Jerome Kelleher[‡] and Peter Ralph[*,†,§]

[*]Institute of Evolution and Ecology and Department of Biology, University of Oregon, Eugene, Oregon, [†]Department of Mathematics, University of Oregon, Eugene, Oregon, [‡]Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford, [§]Department of Data Science, University of Oregon, Eugene, Oregon

ABSTRACT Foreshadowing haplotype-based methods of the genomics era, it is an old observation that the "junction" between two distinct haplotypes produced by recombination is inherited as a Mendelian marker. In a genealogical context, this recombination-mediated information reflects the persistence of ancestral haplotypes across local genealogical trees in which they do not represent coalescences. We show how these non-coalescing haplotypes ("locally-unary nodes") may be inserted into ancestral recombination graphs (ARGs), a compact but information-rich data structure describing the genealogical relationships among recombinant sequences. The resulting ARGs are smaller, faster to compute with, and the additional ancestral information that is inserted is nearly always correct where the initial ARG is correct. We provide efficient algorithms to infer locally-unary nodes within existing ARGs, and explore some consequences for ARGs inferred from real data. To do this, we introduce new metrics of agreement and disagreement between ARGs that, unlike previous methods, consider ARGs as describing relationships between haplotypes rather than just a collection of trees.

KEYWORDS genealogy, tree sequence, haplotypes, ancestral recombination graph

Introduction

Ancestral recombination graphs (ARGs) describe how a set of sampled sequences are related to each other at each position of the genome in a recombining species (Brandt *et al.* 2024; Lewanski *et al.* 2024; Nielsen *et al.* 2024; Wong *et al.* 2024), and there has been significant recent progress on inference through a range of approaches (Rasmussen *et al.* 2014; Speidel *et al.* 2019; Kelleher *et al.* 2019; Zhang *et al.* 2023; Deng *et al.* 2024; Gunnarsson *et al.* 2024). One way of viewing ARGs is as a sequence of local trees, i.e., the genealogical trees that describe how each portion of the genome was inherited by the focal genomes. This is reflected in methodology of some ARG inference methods and in metrics used to assess inference accuracy, as well as in basic terminology. For instance, the "succinct tree sequence", introduced by Kelleher *et al.* (2016), is a common format for describing these inferred ARGs, and is seeing wide use thanks in part to its efficiency and accompanying reliable toolkit, `tskit` (Kelleher *et al.* 2024; Ralph *et al.* 2020).

However, an ARG is emphatically not merely a sequence of trees: viewed another way, it describes inheritance relationships between ancestral haplotypes. These two points of view are related because a single haplotype may extend over many local trees; in other words, the internal nodes in the trees are labeled, and many of these labels are shared between adjacent trees (Wong *et al.* 2024).

Another reason we tend to focus on the trees is that much of our intuition about inference of relationships from genomic data comes from phylogenetics. Indeed, all methods might very roughly be summarized as "more similar sequences are more closely related". For instance, two sequences that share a derived mutation are probably more closely related over some span of genome surrounding the location where the mutation occurs. It has long been observed that not only mutations but also the "junctions" between distinct haplotypes, if they could be somehow identified, would be inherited as Mendelian markers (Fisher 1954; Chapman and Thompson 2003). In more modern terminology, even in the absence of new mutations, recombination between distinct haplotypes can create a novel haplotype whose relationships and origination time could be inferred.

doi: 10.1534/genetics.XXX.XXXXXX
Manuscript compiled: Tuesday 25th February, 2025

[1]Corresponding author: plr@uoregon.edu

Haplotype identity has been largely overlooked in the literature on ARG inference – most methods that have been used so far to measure accuracy of inferred ARGs depend only on the sequence of local trees, not on how ancestral haplotypes span across these trees. For instance, Kelleher *et al.* (2019) and Zhang *et al.* (2023) compared true and inferred ARGs using average Robinson-Foulds (Robinson and Foulds 1981) and Kendall-Colijn (Kendall and Colijn 2016) distances between trees across a regular sequence of genomic positions, using sampled genotypes as labels, while Brandt *et al.* (2022) compared times to most recent common ancestor between pairs of sampled genomes. Neither is affected by shared haplotype structure – two ARGs could be identical by either measure but imply completely different patterns of haplotype sharing and inheritance. Also, Deng *et al.* (2021) evaluated agreement of distributions of distances along the genome between tree topology changes, and Zhang *et al.* (2023) defined a generalization of Robinson-Foulds distance that is the total variation distance between the induced distribution on genotypes; however, neither of these measure the sharing of haplotypes between adjacent trees. An exception is Ignatieva *et al.* (2024), who compared distributions of haplotype spans in true and inferred ARGs, as well as more sophisticated summaries of edges. The additional information provided by haplotype structure can be important: for instance, haplotypes that extend over many local genealogies "tie together" those genealogies, allowing estimates of times of particular ancestors to be informed by larger portions of the genome on which there are many genealogies.

In this paper, we study various aspects of haplotype identity in ARGs. First, we describe a deterministic algorithm that extends the genomic region spanned by ancestral haplotypes using the principle that intermediate nodes in inheritance paths should remain unchanged when possible. These extended portions of ancestral haplotypes manifest as unary nodes in the local trees. To quantify how accurate the new information is, we define and describe how to compute new measures of (dis-)agreement between ARGs that are motivated by the Robinson-Foulds distance between trees but account for haplotype identity. These measures show that the vast majority of these extended haplotypes are correct if the trees are correct, and that substantial information about haplotypes is contained in these nodes in inferred trees as well.

Motivation and statement of problem

Consider the (small portion) of a hypothetical ARG in Figure 1 A. On the first portion of the genome (left-hand tree), the sample nodes (labeled 0, 1, and 2) coalesce into a small subtree: 1 and 2 find a common ancestor in ancestral node 3, which finds a common ancestor with node 0 in ancestral node 4. On the next portion of the genome (right-hand tree), sample node 2 has a different ancestor. This seems reasonable, and a method that infers trees separately on each portion of the genome could not be expected to produce anything different. However, the example becomes more complicated once we consider what these local genealogies imply about haplotype inheritance. Figure 1 B shows the implied inheritance of haplotypes, with the haplotypes carried by 4 to the left and right of the recombination breakpoint labeled L and R. Here, sample node 2 has inherited the chunk of haplotype labeled L from ancestral node 4 via 3, and the haplotype to the right of this from some other node (and so doesn't carry haplotype R). On the other hand, sample node 1 has inherited *both* haplotypes L and R from ancestral node 4, but the trees imply that only haplotype L is inherited via ancestral node 3. This implies – if taken literally – that there must have been a recombination event at some point between node 1 and node 4 that separated the L and R haplotypes, and then these two ancestral (and nonoverlapping) haplotypes coalesced together in ancestral node 4. Although this is possible, it seems unlikely – a more parsimonious explanation is depicted in Figure 1 C, in which sample node 1 inherits the entire LR haplotype from ancestral node 4 through node 3 (and there is a recombination somewhere between node 3 and node 2). This implies that ancestral node 3 inherits from node 4 on the right-hand tree as well, which is depicted in Figure 1 D – and so node 3 has become unary in this tree. Note that the more parsimonious ARG also includes fewer edges: the three distinct edges $4 \to 3$, $3 \to 1$, and $4 \to 1$ in Figure 1 B have been reduced to the two edges $4 \to 3$ and $3 \to 1$ in Figure 1 D.

So, given the ARG shown in Figure 1 A&B, it should be possible to extend the ancestral haplotype represented by node 3 to obtain the ARG shown in Figure 1 C&D, thus adding additional information to the ARG. This might be surprising, as intuition from phylogenetics suggests we can only infer information about the branching points in the tree, not intermediate (unary) nodes. The goal of this paper is to answer: How can we do this, and how accurate is the resulting inference?

Methods

Notation and terminology

We work with the *succinct tree sequence* representation of ARGs (henceforth, "tree sequence"), to take advantage of the tools available in `tskit` (Kelleher *et al.* 2024), and our terminology and notation follows Ralph *et al.* (2020). For our purposes here, a tree sequence $\mathbb{T} = (N, E)$ contains a set of *nodes* N which represent ancestral segments of genome, and *edges* E which represent relationships between nodes over different regions of the genome. Each node $n \in N$ has a *time* t_n, which is the amount of time in the past that the individual who carried that segment of genome lived. Some nodes are *samples*, meaning that they represent genome sequences available as data. Each edge $e \in E$ describes inheritance

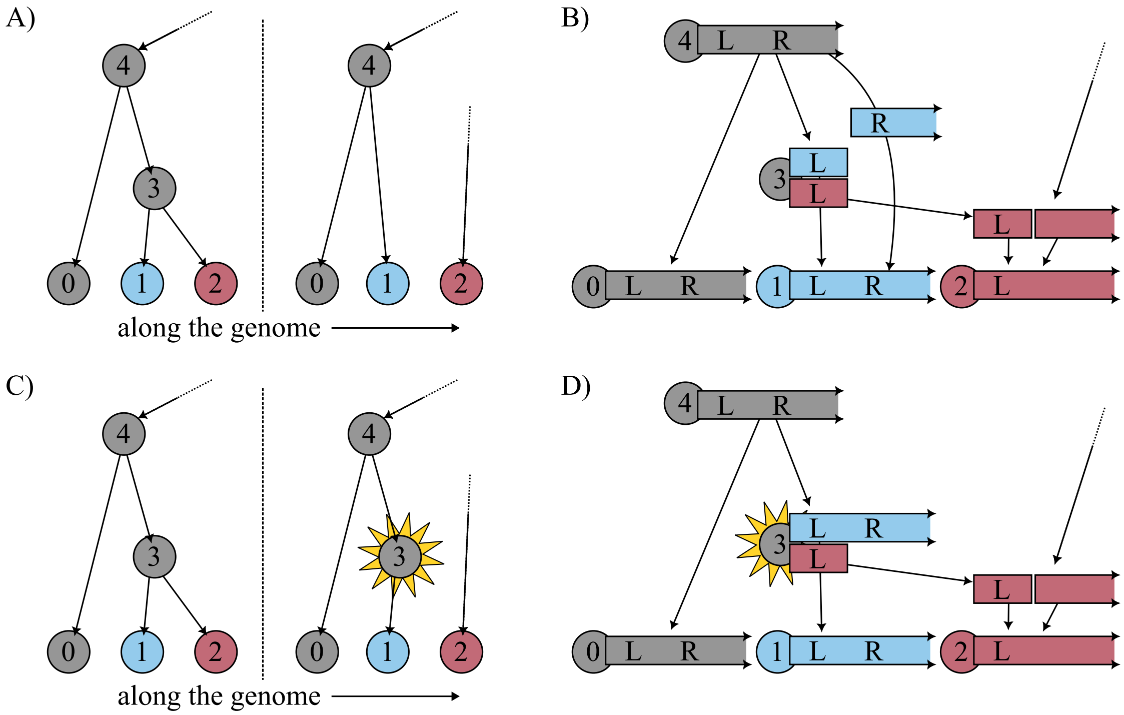

Figure 1 A simple example showing the basic idea (described in more detail in the text): **(A)** a small portion of an ARG without unary nodes; **(B)** the implied inheritance pattern of the two portions of the haplotype carried by ancestral node 4, labeled L and R; **(C)** local trees with a unary node added, which produces **(D)** a more parsimonious haplotype inheritance pattern (that also includes fewer edges).

between a parent node p_e and child node c_e, over a segment of genome $[\ell_e, r_e)$. Suppose that the unique elements of the set of left and right edge endpoints are $0 = a_0 < a_1 < \cdots < a_n = L$, where L is the length of the genome. Using this information, one can construct the sequence of trees $(T_1, ..., T_n)$ that describe how the nodes are related to each other along the genome: each T_k is a tree whose nodes are in N and that describes relationships on the half-open interval $[a_{k-1}, a_k)$. Nodes represent (portions of) ancestral haplotypes, and so we will use the terms interchangeably. Not all nodes appear in each tree, and we say $n \in T_k$ for a node n if the tree T_k describes at least one parent-child relationship for node n.

An algorithm to extend haplotypes

Given a tree sequence, our goal is to identify areas of implied inheritance of haplotypes. Generalizing from Figure 2, we do this by identifying paths of inheritance that are shared across a sequence of local trees but for which some of the intermediate nodes are missing. Concretely, suppose that if in tree T_k there is a chain of inheritance $p \to u_1 \to \cdots \to u_m \to c$ (where $a \to b$ denotes a parent-child relationship) and in tree T_{k+1} there is a chain of inheritance $p \to v_1 \to \cdots \to v_n \to c$, where $\{u_i\}_{i=1}^m$ and $\{v_j\}_{j=1}^n$ are disjoint. This situation implies that c inherited from p over the entire interval $[a_{k-1}, a_{k+1})$, so it seems reasonable to assume that c has inherited from p *along the same path* for that entire interval. In other words, the intermediate nodes $\{u_i\}$ should also lie on the path from c to p in tree T_{k+1}, and conversely the nodes $\{v_j\}$ should lie on that path in tree T_k. Of course, this does not always make sense – for instance, if u_i is already represented somewhere else in T_{k+1}, or if $t_{u_i} = t_{v_j}$, for some i and j. So, we restrict our attention to pairs of such paths in adjacent trees for which $u_i \notin T_{k+1}$ for all $1 \leq i \leq m$, $v_j \notin T_k$ for all $1 \leq j \leq n$, and the times of the nodes $\{u_i\}$ and $\{v_j\}$ are unique. Call a pair of such paths *mergeable*. So, the goal of our algorithm is to iterate over trees, identify mergeable pairs of paths, and then extend the nodes $\{u_i\}$ to T_{k+1}. (We also extend $\{v_j\}$ to T_k, but on a backwards pass.)

An efficient algorithm to do this is described in Algorithm 1. The algorithm considers each tree transition from T_k to T_{k+1} in turn, updating its internal state (which includes possibly modifying T_k) as it goes. Suppose we are at the transition from tree T_k to tree T_{k+1}, which is done by first removing a set of edges O and then adding another set of edges I. O defines a sub-forest F_O of T_k, and I defines a sub-forest F_I of T_{k+1}. The key step in the algorithm is to determine whether the pair of paths that terminate in a given node in two adjacent trees are mergeable. The algorithm we use to do this is given as Algorithm 2, and works as follows. If a pair of paths is mergeable, then the edges of the two paths

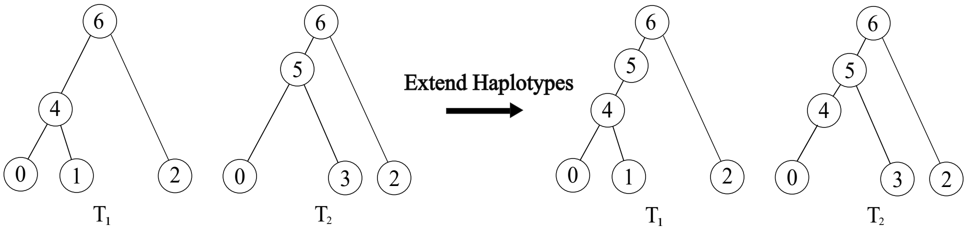

Figure 2 A visualization of the *extend haplotypes* method. In both trees T_1 and T_2, node 0 inherits from node 6, the root: T_1 contains the path $6 \to 4 \to 0$ while T_2 has path $6 \to 5 \to 0$. The intermediate nodes 4 and 5 do not appear in T_2 and T_1 respectively, and so the paths are *mergeable*. The "extend haplotypes" method joins these two paths, inserting the merged path $6 \to 5 \to 4 \to 0$ into both T_1 and T_2.

must lie in O and I, respectively. Suppose an edge in O has child c. To see if c is the base of a pair of mergeable paths, the algorithm traverses up from c in both F_O and F_I; terminating if a node in the other tree is found (i.e., if the node traversed in F_O is in T_{k+1} or if the node traversed in F_I is in T_k) or if a pair of traversed nodes have the same time. If these two traversals end in the same node p, the paths are mergeable. Iterating over all edges in O will thus find all mergeable pairs of paths. There is often more than one pair of mergeable paths in a tree transition; so, the algorithm merges pairs of mergeable paths, starting with pairs that add the smallest number of new edges, until no more are found.

Algorithm 1 simplifies the full algorithm implemented in software in several ways for the sake of clarity – for instance, the bookkeeping required to keep track of T_k and T_{k+1} is omitted. Furthermore, as described the algorithm does one left-to-right pass over the tree sequence; in practice we do repeated passes in both directions until no changes can be made. The main step that is omitted is a description of the `merge` operation, which performs the actual extending of haplotypes. This algorithm is essentially the same as `mergeable` in Algorithm 2, except with additional bookkeeping. Roughly speaking, the algorithm traverses up from the shared base node c, doing the appropriate operations to insert the nodes along the path found in T_k into the path in T_{k+1}. To do this, some edges that end at a_k will be extended to end at a_{k+1}; some edges that begin at a_k will be postponed to begin at a_{k+1}, and some entirely new edges may be added, as in Figure 2. Furthermore, the trees T_k and T_{k+1} (and corresponding forests F_O and F_I) need to be updated.

Algorithm 1: Extend haplotypes. Given an ARG $\mathbb{T}$ with N trees $T_1, \ldots, T_N$ for which edges O_k are removed to transition from T_k to T_{k+1}, identify and merge all mergeable paths (see text). Each child node c_e of each removed edge e is checked to see if it is at the base of two mergeable paths; paths that add fewer new edges are merged first. (The variables m, M, and M' are to ensure this ordering by number of new edges.)

```
def ExtendHaplotypes ($\mathbb{T}$):
    for $k$ in $1 \ldots N$:
        Set $M = 0$ and $M' = \infty$.
        while $M < \infty$:
            for $e \in O_k$:
                Set $m$ = Mergeable ($c_e$, $T_k$, $T_{k+1}$).
                if $m < M$:
                    Merge ($c_e$, $T_k$, $T_{k+1}$)
                else:
                    Set $M' = \min(m, M)$.
            Set $M = M'$ and $M' = \infty$ .
```

Because the algorithm needs to take multiple passes over the tree sequence in each direction, an important practical question for this algorithm is: how many passes do we need to do? The algorithm is monotone (spans of ancestral nodes only increase), so it is guaranteed to terminate in a finite number of passes, but it is also not hard to construct pathological cases that require an arbitrary number of passes. However, experimentation suggests that in practice at most five iterations are needed before the algorithm terminates. Indeed, for even large sequences Table S1 shows that 99% of all changes to an ARG occur after the first iteration, with the algorithm always completing after four iterations.

Dissimilarity between ARGs

If we begin with a tree sequence containing unary nodes, it is straightforward to remove the portions of each node's span on which it is unary, apply Algorithm 1, and quantify how much node span was correctly or incorrectly added. However,

Algorithm 2: Given a node c, trees T_O and T_I, and sub-forests F_O and F_I such that removing F_O and adding F_I turns T_O into T_I, check to see if the paths upwards from c in T_O and T_I are mergeable. If the paths are mergeable then this returns the number of new edges that would be added by extending the path from T_O to T_I; otherwise, this returns ∞. Let $P_O[n]$ and $P_I[n]$ be the parents of node n in the set of edges to be removed and added, respectively (i.e., in F_O and F_I). The variable m_e will record the number of new edges to be added, and m will record the number of extended haplotypes.

```
def Mergeable ($c$, $T_O$, $T_I$):
    Let $p_i = P_I[c]$, $t_i = t[p_i]$, $p_o = P_O[c]$, $t_o = t[p_o]$, and $m_e = m = 0$.
    while True:
        Set $y_i = (p_i \neq NULL)$ & $(p_i \notin T_O)$ & $(t_i < t_o)$
        and $y_o = (p_o \neq NULL)$ & $(p_o \notin T_I)$ & $(t_o < t_i)$
        if not ($y_i$ or $y_o$):
            break
        if $y_i$:
            if $P_I[c] \neq p_i$ and $P_O[c] \neq p_i$:
                Set $m_e = m_e + 1$.
            Set $c = p_i$, $p_i = P_I[p_i]$, and
            $t_i =$ if $(p_i = NULL)$ then $\infty$ else $t[p_i]$ .
        else:
            if $P_I[c] \neq p_o$ and $P_O[c] \neq p_o$:
                Set $m_e = m_e + 1$.
            Set $c = p_o$, $p_o = P_O[p_o]$,
            $t_o =$ if $(p_o = NULL)$ then $\infty$ else $t[p_o]$ ,
            and $m = m + 1$ .
    if $m = 0$ or $p_i \neq p_o$ or $p_i = NULL$:
        Set $m_e = \infty$.
    return $m_e$
```

we are also interested in whether Algorithm 1 improves *inferred* ARGs. Since we are not aware of any current methods for measuring (dis)agreement between ARGs that take into account haplotype identity, we define a measure of *matched span* to quantify this. The method is implemented in the tscompare package.

It is helpful to first describe what we compute in the simple case. We will first simulate tree sequences where nodes that are unary in local trees between coalescent haplotypes are retained. Then, each node is present in both tree sequences, and we can quantify, for each node, how much of their span is correct or incorrect by comparing to the original, true tree sequence.

Now suppose that instead of comparing two tree sequences with the same set of nodes, we wish to compare two tree sequences for which we know the sample nodes are the same but are otherwise unclear as to the equivalency of nodes across sequences. (For instance, with a simulated tree sequence and one inferred from its genotypes; nodes in the former represent actual ancestral haplotypes, and in the latter represent hypothetical ancestors which may or may not resemble the truth.) Call the two tree sequences $\mathbb{T}_1$ and $\mathbb{T}_2$, which should have the same genome length and the same set of sample nodes; in what follows we think of $\mathbb{T}_2$ as the true ARG and $\mathbb{T}_1$ as an inferred ARG. We would like to measure (a) how much of $\mathbb{T}_1$ is found in $\mathbb{T}_2$; (b) how much of $\mathbb{T}_2$ is found in $\mathbb{T}_1$; and (c) how much of $\mathbb{T}_1$ is *not* found in $\mathbb{T}_2$. (Think of these three quantities as the sizes of two relative intersections and difference between the tree sequences, thought of vaguely as sets.) Roughly speaking, we first identify matching nodes as those whose sets of descendant samples agree for the largest span along the genome, and then compute for how much of their spans do their descendant samples agree (or not). An example of our method is illustrated in Figure 3.

The method works as follows. To simplify notation suppose that the two tree sequences have the same set of breakpoints between trees, so that $T_1^{(1)}, \ldots, T_N^{(1)}$ are the trees in $\mathbb{T}_1$ and $T_1^{(2)}, \ldots, T_N^{(2)}$ are the trees in $\mathbb{T}_2$. For a node n and tree T let $S(T, n)$ denote the set of samples that inherit from n in T, and for a pair of nodes n_1 and n_2 with n_1 in $\mathbb{T}_1$ and n_2 in $\mathbb{T}_2$, define

$$\mathcal{M}(n_1, n_2) = \left\{ k : S\left(T_k^{(1)}, n_1\right) = S\left(T_k^{(2)}, n_2\right) \right\},$$

to be the indices of all trees where n_1 and n_2 are ancestral to the same sample set in both ARGs, and

$$m(n_1, n_2) = \sum_{k \in \mathcal{M}(n_1, n_2)} (a_k - a_{k-1}),$$

which is the total span over which the samples below n_1 in $\mathbb{T}_1$ matches the samples below n_2 in $\mathbb{T}_2$. The *matched span* of $\mathbb{T}_1$ in $\mathbb{T}_2$ is then defined to be

$$\underset{\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2) = \max_{\beta : N_1 \to N_2} \sum_{n \in N_1} m(n, \beta(n)),$$

where the maximum is over all mappings β of nodes in $\mathbb{T}_1$ to nodes in $\mathbb{T}_2$, and we require that samples in $\mathbb{T}_1$ are mapped to samples in $\mathbb{T}_2$. (Note that multiple nodes in $\mathbb{T}_1$ may be mapped to the same node in $\mathbb{T}_2$, and that some nodes in $\mathbb{T}_2$ may not be mapped to by any nodes in $\mathbb{T}_1$.) Since the maximum is independent over nodes, we may define for each node $n_1 \in \mathbb{T}_1$ its *best matching node* in $\mathbb{T}_2$ as

$$\alpha(n_1) = \mathrm{argmax}_{n_2 \in N_2}\, m(n_1, n_2),$$

so that

$$\underset{\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2) = \sum_{n \in N_1} m(n, \alpha(n)). \tag{1}$$

If the best-matching node is not unique, we define $\alpha(n_1)$ to be the node in T_2 out of those maximizing $m(n_1, n_2)$ that minimizes $|t^{(1)}_{n_1} - t^{(2)}_{n_2}|$ (and if *this* is not unique, we pick an arbitrary one) – however, this potential ambiguity does not affect the definition of $\underset{\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2)$. Let $s(\mathbb{T}, n)$ denote the total span that node n is present in the local trees,

$$s(\mathbb{T}, n) = \sum_{k=1}^{N} (a_k - a_{k-1}) \mathbf{1}_{n \in T_k} \quad ,$$

where $\mathbf{1}_{n \in T_k}$ is an indicator (i.e., it is 1 if $n \in T_k$ and 0 otherwise), and let $\|\mathbb{T}_1\| = \sum_{n \in N_1} s(\mathbb{T}_1, n)$ be the total span of all nodes in $\mathbb{T}_1$. We then define the *non-matched span* of $\mathbb{T}_1$ in $\mathbb{T}_2$ by

$$\underset{\not\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2) = \sum_{n \in N_1} \left(s(\mathbb{T}_1, n) - m(n, \alpha(n)) \right) = \|\mathbb{T}_1\| - \underset{\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2),$$

which is the total span for all nodes in $\mathbb{T}_1$ over which their descendant samples do *not* match those of their best match in $\mathbb{T}_2$. Contrarily, given a matching $\alpha : \mathbb{T}_1 \to \mathbb{T}_2$, we want to quantify how much of $\mathbb{T}_2$ is represented in $\mathbb{T}_1$. To do this, we define the *inverse matched span* of $\mathbb{T}_1$ in $\mathbb{T}_2$ as

$$\underset{\longleftarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2) = \sum_{n_2 \in N_2} \max_{n_1 \in \alpha^{-1}(n_2)} m(n_1, n_2) \tag{2}$$

where $\alpha^{-1}(n_2)$ is the set of all nodes $n_1 \in \mathbb{T}_1$ whose best match is n_2. This differs from the matched span of $\mathbb{T}_2$ in $\mathbb{T}_1$ because there may be more than one node in $\mathbb{T}_1$ that is mapped to the same node in $\mathbb{T}_2$ – so, if nodes n_1 and n_1' are both mapped by α to the same node n_2, then both count towards $\underset{\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2)$, but only the better match counts towards $\underset{\longleftarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2)$.

A common measure of disagreement between ARGs, first proposed by KUHNER and YAMATO (2015), is to use a weighted average Robinson-Foulds (RF) distance. This could be computed in a very similar way: instead of $m(n, \alpha(n))$ define

$$\mathcal{M}'(n) = \left\{ k : \exists\, n_2 \text{ for which } S\left(T^{(1)}_k,\, n\right) = S\left(T^{(2)}_k,\, n_2\right) \right\},$$

the indices of all trees on which there is *some* node in $\mathbb{T}_2$ whose set of descendant samples matches those of n, and

$$m'(n, \mathbb{T}_2) = \sum_{k \in \mathcal{M}'(n)} (a_k - a_{k-1})$$

the total span over which n finds a match. Then the average RF distance (averaged over locations in the genome) is

$$\frac{1}{L} \left(\sum_{n_1 \in N_1} m'(n_1, \mathbb{T}_2) + \sum_{n_2 \in N_2} m'(n_2, \mathbb{T}_1) \right).$$

In other words, we require a node in $\mathbb{T}_1$ to match the *same* node in $\mathbb{T}_2$ across all trees, but average RF distance allows a different node to match on each tree. The other differences are that *average* RF distance normalizes by sequence length, and is symmetrized. The RF distance between two trees was defined by ROBINSON and FOULDS (1981) to be the minimum number of branch contraction/expansion operations needed to move from one tree to the other (which they then show is equal to the number of edges that induce different splits on the labels). A similar metric on ARGs could be defined using the subgraph-prune-and-regraft moves used by DENG *et al.* (2024).

The matched span, $\underset{\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2)$, measures agreement between *topologies*, but not times. If the ARG is dated (e.g., as in WOHNS *et al.* 2022; DENG *et al.* 2024), we can naturally use the "best match" α to also compare times. Empirically, dating error seems to be more or less homoskedastic on a log scale, so we recommend using the weighted root-mean-squared error of log(times), computed as

$$\mathrm{wRMSE}_t(\mathbb{T}_1, \mathbb{T}_2) = \sqrt{\frac{\sum_{n \in N_1} s(\mathbb{T}_1, n) \left(\log\left(1 + t_n^{(1)}\right) - \log\left(1 + t_{\alpha(n)}^{(2)}\right)\right)^2}{\|\mathbb{T}_1\|}}, \tag{3}$$

where the transformation is $t \mapsto \log(1+t)$ to avoid $\log(0)$. The mean is computed weighting by node span, so that a dating error is more impactful for a node with a longer span.

The implementation of this method in `tscompare` additionally produces relative values: the ARG RF value (non-matched span relative to $\mathbb{T}_1$) and true proportion represented (inverse matched span relative to $\mathbb{T}_2$). The ARG RF (which we call "ARF") is defined to be the matched span proportional to the total span of nodes in $\mathbb{T}_1$,

$$\mathrm{ARF}(\mathbb{T}_1, \mathbb{T}_2) = 1 - \frac{\underset{\longrightarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2)}{\|\mathbb{T}_1\|}, \tag{4}$$

and so if $\mathbb{T}_2$ represents the truth, is analogous to a false positive rate. The *true proportion represented* (TPR) is the inverse matched span between two trees relative to the total span of nodes in $\mathbb{T}_2$,

$$\mathrm{TPR}(\mathbb{T}_1, \mathbb{T}_2) = \frac{\underset{\longleftarrow}{\mathrm{match}}(\mathbb{T}_1, \mathbb{T}_2)}{\|\mathbb{T}_2\|}, \tag{5}$$

and is analogous to statistical power. The outputs framed as proportions relative to one of the given tree sequences is more easily understood for comparing pairs of tree sequences than the original matched span and inverse matched span, whose units are length of spans. We compute the ARF and TPR of in a simple example in Figure 3.

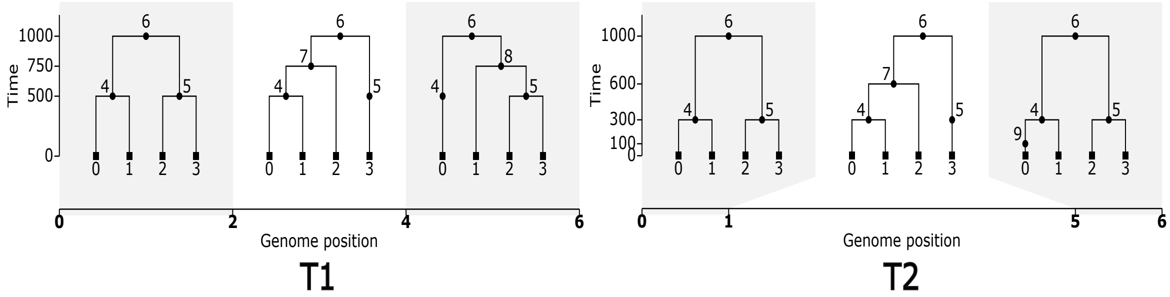

Figure 3 For two tree sequences $T1$ and $T2$ the *matched span*, $\underset{\longrightarrow}{\mathrm{match}}(T1, T2)$, matches nodes in $T1$ to nodes in $T2$ based on identical sample sets. In this example, node 8 has no match in $T2$ as there are no nodes in $T2$ with sample set $\{1, 2, 3\}$. Node 4 has no match on $[4, 5)$ and matches with node 9 on $[5, 6)$. Thus the maximal mapping for 4 should be to itself. On the rest of the genome, all of nodes match with their identical counterpart. This makes the matched span $\underset{\longrightarrow}{\mathrm{match}}(T1, T2) = 43$ and $\mathrm{ARF}(T1, T2) = \frac{3}{46}$. Given the above matching, the inverse matching will match nodes 0 through 7, and node 9 has no match since its only possible match ($4 \in T1$) was not the best match from $T1 \rightarrow T2$. This means the inverse matched span $\underset{\longleftarrow}{\mathrm{match}}(T1, T2) = 43$ and $\mathrm{TPR}(T1, T2) = \frac{43}{47}$.

Metrics on ARGs Neither the matched span or non-matched span of $\mathbb{T}_1$ in $\mathbb{T}_2$ are metrics in the mathematical sense (i.e., symmetric, nonzero distance between distinct points, and satisfying the triangle inequality). This is by design: in practice it is not possible to infer all aspects of the true ancestry of a set of samples (i.e., all their genetic ancestors who ever lived), and so we wanted to quantify "How much of the true relationships does this ARG represent?" However, it is worth noting that the symmetrized version of non-matched span

$$\overline{\underset{\not\to}{\text{match}}}(\mathbb{T}_1, \mathbb{T}_2) = \underset{\not\to}{\text{match}}(\mathbb{T}_1, \mathbb{T}_2) + \underset{\not\to}{\text{match}}(\mathbb{T}_2, \mathbb{T}_1),$$

is a metric. To see this, first suppose that we have a bijection between the nodes of $\mathbb{T}_1$ and $\mathbb{T}_2$, and view each ARG as a subset of the space $[0, L) \times N \times N$, where N is the shared set of nodes. Then, dissimilarity is the Lebesgue measure of the relative difference of the two sets: $|\mathbb{T}_1 \setminus \mathbb{T}_2|$, and so the symmetrized version is the measure of the symmetric difference $|\mathbb{T}_1 \Delta \mathbb{T}_2|$, which is well-known to be a metric (RUDIN 1976). If the two ARGs have the same number of nodes, we can consider all bijections between their nodes. The symmetric difference between $\mathbb{T}_1$ and $\mathbb{T}_2$, related through each bijection is a metric. Then the minimum over all such metrics will still be a metric since the minimum over a finite number of metrics is also a metric. This also extends to two tree sequences with different numbers of nodes, $|N_1| \neq |N_2|$, as we can take the minimum over all possible matchings $\mathbb{T}_1 \to \mathbb{T}_2$ and $\mathbb{T}_2 \to \mathbb{T}_1$.

The RF distance (ROBINSON and FOULDS 1981) essentially counts the number of differing branches between two trees; the averaged RF distance (KUHNER and YAMATO 2015) averages this distance across local trees, weighted by span along the genome. The method we present here for measuring dissimilarity between topologies of ARGs is a straightforward generalization that takes into account span along the genome of inferred ancestral haplotypes (and separates the metric into two pieces). However, the RF metric has many undesirable properties – for instance, moving a single tip can result in a tree with maximum distance to the original – and there is a substantial literature giving more robust generalizations (reviewed by LLABRÉS *et al.* 2021). Many of these generalizations (e.g., BÖCKER *et al.* 2013) relax the requirement that the match between subtended sample sets be exact, and weight matches in some way by the size of the dissimilarity. We considered such definitions as well, but kept to the simple case for computational tractability – the generalization of BÖCKER *et al.* (2013) is NP-hard to compute, even for a single tree. In the ARG literature, ZHANG *et al.* (2023) defines a metric (called "ARG total variation distance") that includes branch lengths, in a way similar to ROBINSON and FOULDS (1979) and KUHNER and FELSENSTEIN (1994); however, it is still applied to ARGs as an average over local trees, without enforcement of identity across haplotypes; it would be useful to extend our dissimilarity to include branch lengths.

Simulations

Our method for extending haplotypes is applicable to any ARG. However its accuracy depends on the overall structure of the ARG it is applied to. Thus to understand how well our methods can infer ancestral haplotypes we work with ARGs simulated across a range of parameter values. To do this, we simulate ARGs containing full haplotypes using `msprime` (KELLEHER *et al.* 2016; BAUMDICKER *et al.* 2021), with the `coalescing_segments_only` option set to False. Although `msprime` simulates many events that do not create a coalescence in some local tree, by default it only outputs information for nodes which contain a coalescence (i.e., are the MRCA of some pair of samples at some point on the genome). Furthermore, by default it only outputs those segments of the genome on which there is a coalescence. Said another way, by default all ancestral nodes in an ARG output by `msprime` are the MRCA of some pair of samples at every point in the genome on which they are represented. However, here we are interested in those segments of genome on which the nodes are *not* coalescent; i.e., where they are unary in the local trees. Setting `coalescing_segments_only` to False includes just this information: any ancestral segments for which these coalescent nodes are ancestral to any samples – so, the unary portions of their spans as well. However, this includes more information than we want: we hope to recover those portions of ancestral haplotypes on which the nodes are unary, but adjacent to a region of the genome where the node is not unary. For instance, if a lineage carrying an ancestral segment of genome that spans $[a, c)$ coalesces with another spanning $[b, d)$, with $a < b < c < d$, then the resulting node is only coalescent on $[b, c)$ but we hope using this algorithm to extend the node's span to $[a, b)$ and $[c, d)$ (on which the node is unary). However, following this example, the first lineage might also carry a segment $[x, y)$ that is disjoint from the segment $[a, d)$. We call these segments "isolated non-coalescent segments"; they have also been called "trapped unary spans" (by WONG *et al.* 2024). Such isolated segments will not be recovered by our algorithm, and would likely be unrecoverable by any other method. So, after simulation, we first remove these isolated, non-coalescent segments. To give an idea of what proportion of the full spans of ancestral nodes these isolated non-coalescent segments represent, a simulation of 1000 samples with genome length 5×10^7, recombination rate 10^{-8}, and population size 10^4 has about half the total span of all nodes in isolated, non-coalescent segments. For more discussion of these segments, see BAUMDICKER *et al.* (2021).

We used simulations of several scenarios. To include the effects of heterogeneous recombination rate, in some we used stdpopsim (ADRION *et al.* 2020) to simulate chromosome 1 of *Canis familiaris* using the CanFam3 genetic map from CAMPBELL *et al.* (2016). *"Constant dog"* simulations simulated this chromosome in a population of (constant) size 10^4. *"Expanding dog"* simulations were similar, but used a discrete-time Wright–Fisher model to simulate a small population of 100 that expanded to 1,000 individuals ten generations ago, which then doubled every generation to reach 512,000 individuals. All jobs for which runtime was recorded were executed on an Intel Xeon Gold 6148 processor. Additionally,

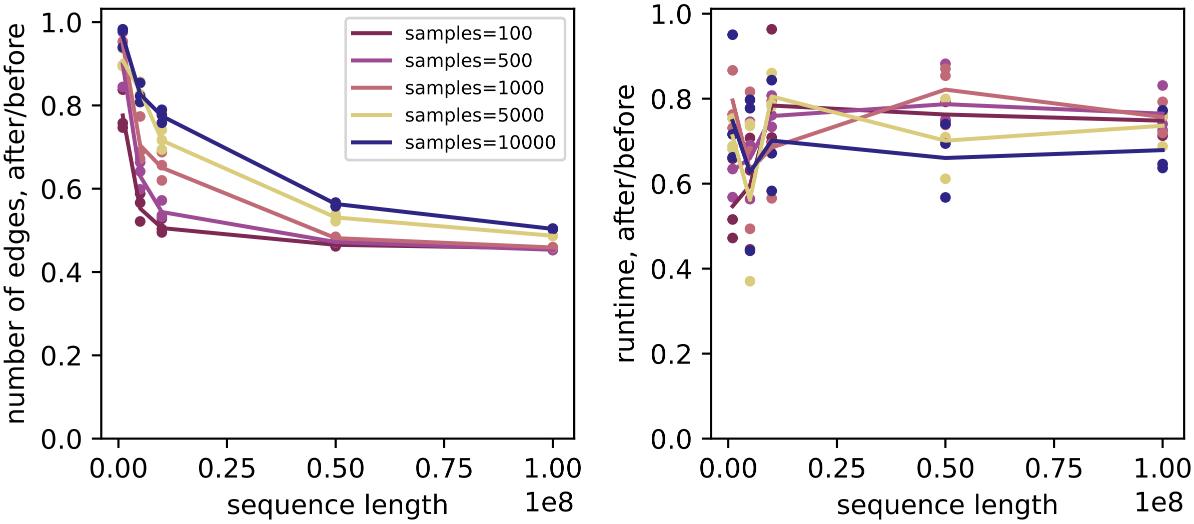

Figure 4 Ratio of **(A)** number of edges, and **(B)** runtime for computing Tajima's D; before and after extending haplotypes. For instance, extending haplotypes reduces number of edges by about 50% and statistic computation runtime by about 20% for long sequences. Horizontal axis shows sequence length; colors show numbers of samples; with lines showing averages across replicates. The original tree sequence was simulated with the "expanding dog" expanding population and subset to various sizes; see Methods for details. Absolute values are shown in Supplementary Figure S1.

we compare accuracy between sequences modified from a "true" ARG using our matched span methods. These ARGs were simulated with an effective population size of 10,000 and recombination rate of 10^{-8} with between 10 and 1,000 samples and a genome length between 10^6 to 5×10^7.

Results

Tree sequence compression and computation

In the simple example in Figure 1, extending haplotypes replaces three edges ($0 \rightarrow 3$ and $3 \rightarrow 4$ on the left tree, and $0 \rightarrow 4$ on the right tree) by two edges ($0 \rightarrow 3$ and $3 \rightarrow 4$ on both trees). If all edge endpoints were unique, then we'd expect *every* edge to be extendable on one of its ends (except those pendant to the root and some of those adjacent to chromosome ends), leading to a reduction in number of edges by almost exactly one third. Experiments with an earlier version of the algorithm showed that if we only extend haplotypes on such "paths of length 1", then the hypothesized reduction of 1/3 is achieved for long sequences. It is possible for Algorithm 1 to add edges, as in Figure 2, but we still expect the number of edges to decrease by more than 1/3. Indeed, Figure 4 shows that Algorithm 1 nearly cuts the number of edges in half, as long as the sequence is long enough.

This reduction in edges can also lead to a reduction in computation time for algorithms using the succinct tree sequence data structure. Indeed, Figure 4 shows that computation time is reduced by 10–20% for a typical statistic (here, Tajima's D), computed in an efficient incremental manner along the genome as implemented in `tskit`. As described in Ralph *et al.* (2020), for these incremental algorithms the addition or removal of an edge requires updates to the state of the parent node and all nodes ancestral to it. Extending haplotypes yields a tree sequence with fewer edge removals and insertions, and thus requires fewer traversals to the roots.

Supplementary Figure S2 shows these results are not specific to the demographic scenario. Supplementary Figures S1 and S3 also show that our implementation of Algorithm 1 is quite efficient, running at chromosome scale in seconds to minutes for hundreds or thousands of samples, or minutes to hours for tens of thousands of samples.

Accuracy with true trees

Our next task is to confirm that the haplotypes extended by Algorithm 1 are indeed correct – i.e., that in addition to compression, we are also gaining information. To do this, we simulate ARGs containing full haplotypes using `msprime`, apply the simplification algorithm (Kelleher *et al.* 2018; Wong *et al.* 2024) to reduce these so that there are no unary nodes (i.e., any node present in a local tree is a coalescent node or a sample), and then apply Algorithm 1 to the result (see Methods for more detail). The method can potentially extend the spans of each node (additional span over which the node will be unary); and we can quantify how much of these extended spans were in the original ARG (and thus correctly extended).

As seen in Figure 5, the vast majority of span added by extending haplotypes is correct. In this example (which is typical), 99% of all added span is correct; 95% of nodes have no incorrectly added span; and those incorrectly added spans are nearly always a small fraction of the original span. The added information is significant: the algorithm typically increases spans (i.e., lengths of ancestral haplotypes) by around 50%.

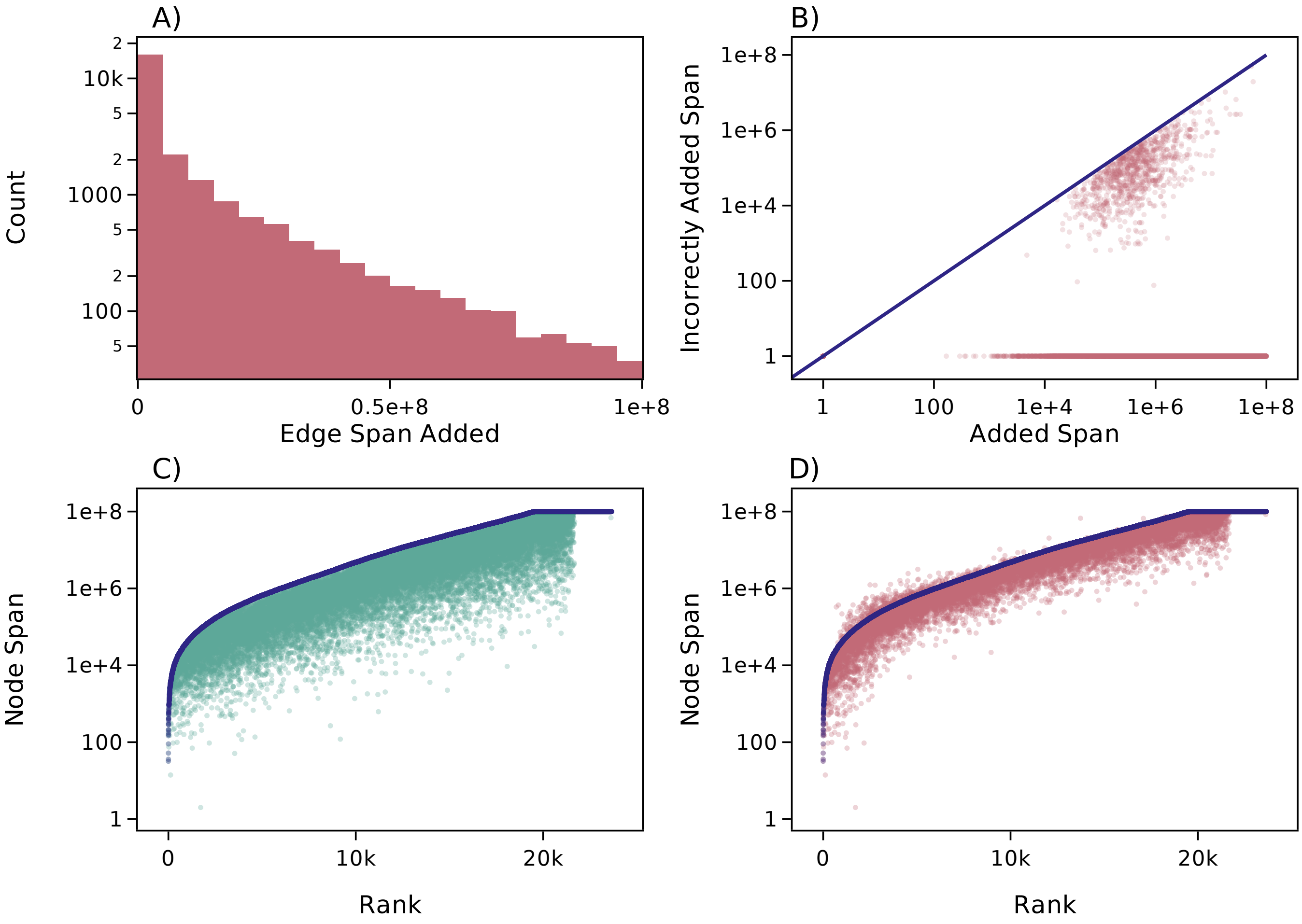

Figure 5 The effect of extending haplotypes on per-node spans in an ARG simulated with 10^4 diploid samples in a population with $N_e = 10^4$, and recombination rate of 10^{-8} on a sequence of length 10^8. **(A)** Distribution of total amount of span added across nodes by Algorithm 1; note the log scale on the y axis. **(B)** Amount of incorrectly added span, plotted against total span, by node. 95% of nodes have no incorrect span; of the remainder, nearly all have less than 5% incorrectly added; see Supplementary Figure S4. Note the log scale on both the x and y axes. Plots **(C)** and **(D)** show total spans per node, ordered by total span in the original ARG (which includes unary nodes). Dark blue dots in both show the spans in this original ARG. In **(C)**, lighter green dots show span after removing unary spans using *simplify*, while in **(D)**, lighter red dots show span after extending haplotypes, i.e., applying Algorithm 1 to the simplified ARG.

These statistics are also reflected at the genomic scale, using measures of matched span. Line “SE” in Figure 6 shows total amounts of span removed by simplification and re-inferred by Algorithm 1 (correctly and incorrectly). (Lines labeled with “I” involve re-inference of the ARG; discussed next.) The top row shows the proportion of the given ARG that does not match the original (“ARF”), showing that the total amount of mis-matching span produced by extending haplotypes is very small ($\approx$ 1%). (Simplification does not produce non-matched span, so the “S” line is at zero.) The bottom row shows the proportion of the original ARG that is represented in the given ARG (“TRP”). This shows that a large proportion of an ARG can be removed by simplification, indicating that coalescent nodes are unary over a substantial portion of their spans. (Since the `simplify` operations removes these unary portions of haplotypes, the “simplified” line

("S") on the bottom two plots of Figure 6 shows the proportion of nodes' spans on which they are not unary.) However, Algorithm 1 can correctly replace most of these portions of haplotypes, especially with larger sample sizes and sequence lengths (the "simplified–extended" line; "SE"). For instance, the rightmost points show that with 1,000 samples and a 5×10^7bp genome, haplotypes are unary on about half their spans (on average), and extending haplotypes can infer more than half of this missing unary span from coalescent information only.

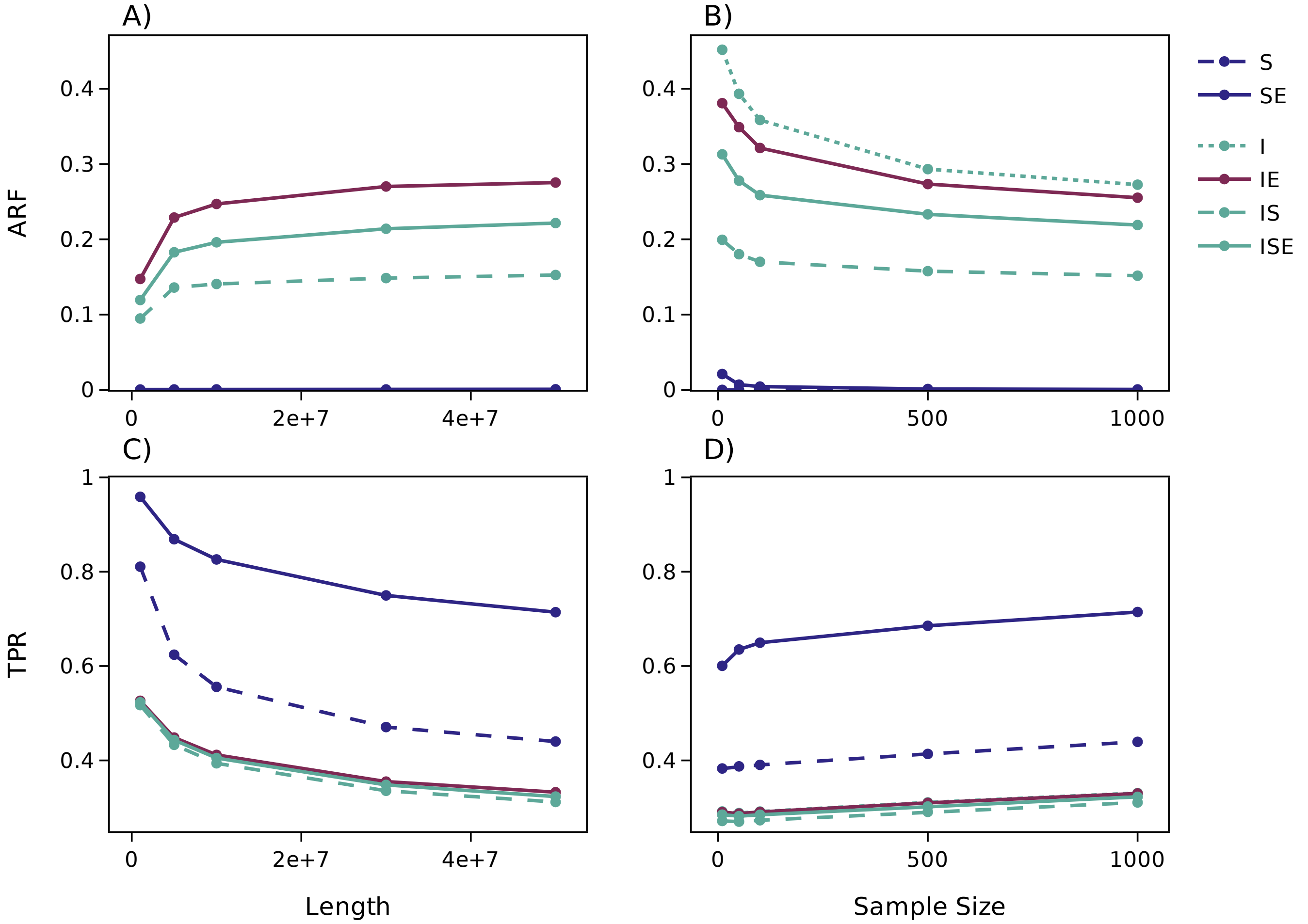

Figure 6 Accuracy and sensitivity of extended haplotypes across a range of sample sizes, on a sequence of length 5×10^7 (right); and a range of sequence lengths, with 1,000 diploid samples (left). For each, a simulated ARG containing unary haplotype spans was (i) *simplified* ('S'), removing the unary spans, and (ii) *inferred* ('I'), using `tsinfer` on genotypes and dated using `tsdate`; then each of these had its haplotypes *extended* ('SE', 'IE'). The inferred, then simplified, ARG ('IS') and its subsequent extension ('ISE') are also shown. **Top row:** ARF, the dissimilarity to the true ARG, as proportion of haplotypes that are not represented in the true ARG (Equation 4). **Bottom row:** TPR, agreement to the true ARG as proportion of the true ARG that is represented.

Inferred ARGs

So far, we have demonstrated that there is potentially ample information in the coalescent-only trees to extend haplotypes. Does this work with *inferred* ARGs? As illustrated by WONG *et al.* (2024), there is a significant diversity in the structures inferred by current methods, and here we focus on `tsinfer` KELLEHER *et al.* (2019) which infers ARGs containing unary nodes as a byproduct of its inference algorithm. A comparison across other ARG inference methods is left for future work. We simulated ARGs containing unary-spanning haplotypes (as above), then re-inferred ARGs from the associated genotypes and performed various operations on the results. Since our matched spans method breaks ties with time, we date the re-inferred ARG using `tsdate`. First, Figure 6 shows that `tsinfer` has a substantial portion of already–extended haplotypes: comparing the "inferred" ("I"; dotted green) line to the "inferred-simplified" ("IS"; dashed green) line we see that inclusion of these unary spans increases the amount of correctly inferred material by around 4% (bottom panels), but that roughly 20% of the unary spans in the inferred ARG are incorrect (top panels). Furthermore, comparing to the inferred-extend(“IE”; red) line, we see that extending the tree sequence output with tsinfer adds relatively little span (less than about 1%). Applying Algorithm 1 to the inferred-and-then-simplified ARG (“ISE”; solid green line) produces an ARG with both less correct and incorrect span. Additionally, the “IE” and “ISE” ARGs contain approximately the same number of edges (difference of $\approx 1\%$). It is also helpful to note that accuracy (i.e., proportion of the true ARG that is inferred; bottom panels) greatly increases with larger sample sizes, possibly due to resolution of polytomies. (Recall that due to computational constraints, “correct” and “incorrect” spans are determined here by *exact* match of subtending samples.) We additionally provide values of “I”, “IS”, “IE”, and “ISE” in Table S2.

Data availability

The method to extend haplotypes described here is available through the `tskit` python and C APIs (https://tskit.dev/tskit) as `extend_haplotypes`; methods to compare ARGs are implemented in the `tscompare` python package (https://tskit.dev/tscompare). Scripts used to produce the results in this paper are available at https://github.com/hfr1tz3/haplotypes-and-ancestral-recombination-graphs.

Discussion

We began this study with the observation that the simple transformation of Figure 2 would reduce the number of edges in the succinct tree sequence representation ARGs. This is essentially a recombination-based parsimony argument, and we have shown that this line of reasoning leads to ARGs that are substantially more compact and faster to operate on, and that contain more complete information about true ancestral relationships. These extended ancestral haplotypes manifest as unary nodes in the local trees. Although a number of ARG inference methods may be taking advantage of this information, it is our impression that this source of information is not widely appreciated. In fact, due to the field’s focus on local trees rather than haplotypes, we had to develop a haplotype-aware measure of (dis)agreement between ARGs in order to study the accuracy of the proposed algorithm.

There are good reasons to think that lengthening the spans of ancestral haplotypes could lead to substantial gains in accuracy of ARG inference. For instance, information about inferring the age of a particular mutation derives almost entirely from constraints at nearby, linked sites. Extending ancestral haplotypes from one site into neighboring regions conceptually allows information from those local trees to inform age inference at that site as well.

We have also explored the degree to which `tsinfer` already makes use of this information, and whether this algorithm can be used to improve inference. The results do not provide a clear ordering: for instance, although `tsinfer`-produced ARGs have a substantial portion of correctly inferred unary haplotypes, removing these with simplification decreases both ARF (i.e., proportion of “wrong” haplotypes) and TPR (the proportion of the truth that is correctly inferred). Extending haplotypes restores a large amount of this correctly inferred span, but also introduces incorrect spans. Further work is needed to determine how the balance of “true and false positive rates” affects downstream uses, and whether results would differ if the requirement that sets of subtended nodes match exactly was relaxed. The efficient computational tools we have implemented (in `tskit` and `tscompare`) should facilitate this exploration.

Ignorance and omission in an ARG As motivation, we presented above a “historical” view of ARGs – i.e., that each aspect of an inferred ARG is intended to represent a portion of some particular historical genome (for instance, the MRCA of some set of sampled genomes). Furthermore, Figure 1 A&B implicitly takes the position that relationships *not* depicted in an ARG are implied to not exist. As discussed in Wong *et al.* (2024), an alternative interpretation of the ARG depicted in Figure 1 A&B would be that we have no information as to how node 2 inherited from node 4 on the right-hand interval, rather than saying that the line of transmission specifically did not pass through node 3. The “simplification” algorithm (Kelleher *et al.* 2018) and the Hudson algorithm for coalescent simulation (Hudson 1983; Kelleher *et al.* 2016) each specifically discard information about any such “non-coalescent” portions of ancestral haplotypes; so for ARGs produced by these algorithms, the correct interpretation is that the omission of unary spans reflects a lack of knowledge. In this paper, we have shown that, for the most part, this missing information can be imputed.

Parsimony Much of the early work on ARG inference aimed to extract as much information as possible out of the small datasets of the time, and so, roughly speaking, integrated over possible ARGs with the goal of inferring higher-level parameters: mostly, scaled mutation rate and recombination rate (for instance, Hudson and Kaplan (1985); Griffiths and Marjoram (1996); Kuhner *et al.* (2000); Stephens and Donnelly (2000); Fearnhead and Donnelly (2001)). However, the space of possible ARGs for a given dataset is extremely large, and other work aimed to identify the minimum number of recombinations needed to explain a given dataset under the infinite alleles model of mutation (e.g., Hein 1990; Myers and Griffiths 2003; Song and Hein 2005), which turns out to be NP-complete (Wang *et al.* 2001). So, the field turned to more heuristic methods – for instance, Minichiello and Durbin (2006) used an algorithm to produce “plausible” ARGs (i.e., those that explained the data with few mutations and recombinations), and searched for associations with traits in the resulting ensemble of ARGs. (See Wong *et al.* (2024) for more historical discussion.) Our approach for extending haplotypes follows the same logic, that an ARG with fewer recombination events is more parsimonious, and thus more likely. For this reason, it will occasionally be wrong even if the trees are correct, although in practice this source of error is likely much smaller than error in tree inference itself.

IBD in ARGs The term "identity by descent" (IBD) is used to mean many different (but related) things, and length distributions of shared IBD segments can be used for inference of recent demographic history (for instance, Al-Asadi *et al.* 2019; Ringbauer *et al.* 2017; Browning and Browning 2015; Yang *et al.* 2016; Silcocks *et al.* 2023). A commonly-used definition in the context of a given ARG says that the two genomes share an IBD segment if each has inherited the segment from their common ancestor along a single path (e.g., Ralph and Coop 2013). Largely for computational reasons, this is the definition that is used in `tskit`'s IBD-finding methods (by G. Tsambos in `tskit` (Kelleher *et al.* 2024); see also Tsambos *et al.* (2023)). However, many simulation and inference methods produce ARGs as shown in Figure 1 A&B, in which inheritance of a single segment is represented by more than one edge. This means that the `ibd_segments` method of `tskit` will return shorter segments than it ought to. However, as we have shown above, our method of extending haplotypes will modify the tree sequence so that the inherited segment is represented by a single edge (as in Figure 1 C&D). So, if our method (`extend_haplotypes`) is applied before finding IBD segments (with `ibd_segments`), then the resulting segments should much more accurately represent the IBD segments (in the "path" sense used here) implied by the tree sequence. Whether the resulting segments better match those predicted by theory depends also on the quality of tree inference. Note also that Huang *et al.* (2024) and Guo *et al.* (2024) both provide methods for tree sequences to compute a different definition of IBD (segments on which the MRCA does not change), which is unaffected by this issue. Further work is needed to understand how accurately IBD segments are inferred by various ARG inference methods.

Acknowledgments

The authors would like to thank Dr. Yan Wong for his helpful comments and suggestions. JK acknowledges EPSRC (research grant EP/X024881/1) and the Robertson Foundation. JK and PR acknowledge support from the NIH NHGRI (research grant HG012473).

Literature Cited

Adrion, J. R., C. B. Cole, N. Dukler, J. G. Galloway, A. L. Gladstein, G. Gower, C. C. Kyriazis, A. P. Ragsdale, G. Tsambos, F. Baumdicker, J. Carlson, R. A. Cartwright, A. Durvasula, I. Gronau, B. Y. Kim, P. McKenzie, P. W. Messer, E. Noskova, D. O. D. Vecchyo, F. Racimo, T. J. Struck, S. Gravel, R. N. Gutenkunst, K. E. Lohmueller, P. L. Ralph, D. R. Schrider, A. Siepel, J. Kelleher, and A. D. Kern, 2020 A community-maintained standard library of population genetic models. eLife **9**.

Al-Asadi, H., D. Petkova, M. Stephens, and J. Novembre, 2019 Estimating recent migration and population-size surfaces. PLOS Genetics **15**: 1–21.

Baumdicker, F., G. Bisschop, D. Goldstein, G. Gower, A. P. Ragsdale, G. Tsambos, S. Zhu, B. Eldon, E. C. Ellerman, J. G. Galloway, A. L. Gladstein, G. Gorjanc, B. Guo, B. Jeffery, W. W. Kretzschumar, K. Lohse, M. Matschiner, D. Nelson, N. S. Pope, C. D. Quinto-Cortés, M. F. Rodrigues, K. Saunack, T. Sellinger, K. Thornton, H. van Kemenade, A. W. Wohns, Y. Wong, S. Gravel, A. D. Kern, J. Koskela, P. L. Ralph, and J. Kelleher, 2021 Efficient ancestry and mutation simulation with msprime 1.0. Genetics **220**: iyab229.

Böcker, S., S. Canzar, and G. W. Klau, 2013 The generalized Robinson-Foulds metric. In A. Darling and J. Stoye, editors, *Algorithms in Bioinformatics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 156–169.

Brandt, D., X. Wei, Y. Deng, A. H. Vaughn, and R. Nielsen, 2022 Evaluation of methods for estimating coalescence times using ancestral recombination graphs. Genetics **221**: iyac044.

Brandt, D. Y., C. D. Huber, C. W. Chiang, and D. Ortega-Del Vecchyo, 2024 The promise of inferring the past using the ancestral recombination graph. Genome Biology and Evolution **16**: evae005.

Browning, S. R., and B. L. Browning, 2015 Accurate non-parametric estimation of recent effective population size from segments of identity by descent. The American Journal of Human Genetics **97**: 404 – 418.

Campbell, C. L., C. Bhérer, B. E. Morrow, A. R. Boyko, and A. Auton, 2016 A pedigree-based map of recombination in the domestic dog genome. G3 Genes|Genomes|Genetics **6**: 3517–3524.

Chapman, N. H., and E. A. Thompson, 2003 A model for the length of tracts of identity by descent in finite random mating populations. Theor Popul Biol **64**: 141–150.

Deng, Y., R. Nielsen, and Y. S. Song, 2024 Robust and accurate Bayesian inference of genome-wide genealogies for large samples. bioRxiv .

Deng, Y., Y. S. Song, and R. Nielsen, 2021 The distribution of waiting distances in ancestral recombination graphs. Theoretical population biology **141**: 34–43.

Fearnhead, P., and P. Donnelly, 2001 Estimating recombination rates from population genetic data. Genetics **159**: 1299–1318.

Fisher, R. A., 1954 A fuller theory of 'junctions' in inbreeding. Heredity **8**: 187–197.

Griffiths, R., and P. Marjoram, 1996 Ancestral inference from samples of DNA sequences with recombination. Journal of Computational Biology **3**: 479–502. PMID: 9018600.

Gunnarsson, Á. F., J. Zhu, B. C. Zhang, Z. Tsangalidou, A. Allmont, and P. F. Palamara, 2024 A scalable approach for genome-wide inference of ancestral recombination graphs. bioRxiv .

Guo, B., V. Borda, R. Laboulaye, M. D. Spring, M. Wojnarski, B. A. Vesely, J. C. Silva, N. C. Waters, T. D. O'Connor, and S. Takala-Harrison, 2024 Strong positive selection biases identity-by-descent-based inferences of recent demography and population structure in *Plasmodium falciparum*. Nature Communications **15**: 2499.

Hein, J., 1990 Reconstructing evolution of sequences subject to recombination using parsimony. Mathematical Biosciences **98**: 185–200.

Huang, Z., J. Kelleher, Y.-b. Chan, and D. J. Balding, 2024 Estimating evolutionary and demographic parameters via ARG-derived IBD. bioRxiv .

Hudson, R. R., 1983 Properties of a neutral allele model with intragenic recombination. Theoretical Population Biology **23**: 183–201.

Hudson, R. R., and N. L. Kaplan, 1985 Statistical properties of the number of recombination events in the history of a sample of DNA sequences. Genetics **111**: 147–164.

Ignatieva, A., M. Favero, J. Koskela, J. Sant, and S. R. Myers, 2024 The length of haplotype blocks and signals of structural variation in reconstructed genealogies. bioRxiv .

Kelleher, J., A. M. Etheridge, and G. McVean, 2016 Efficient coalescent simulation and genealogical analysis for large sample sizes. PLoS computational biology **12**: e1004842.

Kelleher, J., B. Jeffery, Y. Wong, P. Ralph, G. Tsambos, S. H. Zhan, K. R. Thornton, D. Goldstein, A. W. Wohns, D. Mbuli-Robertson, L. Kirk, G. Gower, H. van Kemenade, M. F. Rodrigues, B. Zhang, G. Bisschop, C. Weiss, D. Palmer, C. Ellerman, J. Guez, N. Pope, S. Karthikeyan, I. Rebollo, S. Belsare, A. Kern, and M. Aspbury, 2024 tskit-dev/tskit: Python 0.5.8.

Kelleher, J., K. R. Thornton, J. Ashander, and P. L. Ralph, 2018 Efficient pedigree recording for fast population genetics simulation. PLOS Computational Biology **14**: 1–21.

Kelleher, J., Y. Wong, A. W. Wohns, C. Fadil, P. K. Albers, and G. McVean, 2019 Inferring whole-genome histories in large population datasets. Nature Genetics **51**: 1330–1338.

Kendall, M., and C. Colijn, 2016 Mapping phylogenetic trees to reveal distinct patterns of evolution. Molecular Biology and Evolution **33**: 2735–2743.

Kuhner, M. K., and J. Felsenstein, 1994 A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Molecular Biology and Evolution **11**: 459–468.

Kuhner, M. K., and J. Yamato, 2015 Assessing differences between ancestral recombination graphs. Journal of Molecular Evolution **80**: 258–264.

Kuhner, M. K., J. Yamato, and J. Felsenstein, 2000 Maximum likelihood estimation of recombination rates from population data. Genetics **156**: 1393–1401.

Lewanski, A. L., M. C. Grundler, and G. S. Bradburd, 2024 The era of the ARG: An introduction to ancestral recombination graphs and their significance in empirical evolutionary genomics. PLOS Genetics **20**: 1–24.

Llabrés, M., F. Rosselló, and G. Valiente, 2021 The generalized Robinson-Foulds distance for phylogenetic trees. Journal of Computational Biology **28**: 1181–1195. PMID: 34714118.

Minichiello, M. J., and R. Durbin, 2006 Mapping trait loci by use of inferred ancestral recombination graphs. The American Journal of Human Genetics **79**: 910 – 922.

Myers, S. R., and R. C. Griffiths, 2003 Bounds on the minimum number of recombination events in a sample history. Genetics **163**: 375–394.

Nielsen, R., A. H. Vaughn, and Y. Deng, 2024 Inference and applications of ancestral recombination graphs. Nature Reviews Genetics : 1–12.

Ralph, P., and G. Coop, 2013 The geography of recent genetic ancestry across Europe. PLoS Biol **11**: e1001555.

Ralph, P., K. Thornton, and J. Kelleher, 2020 Efficiently summarizing relationships in large samples: A general duality between statistics of genealogies and genomes. Genetics : genetics.303253.2020.

Rasmussen, M. D., M. J. Hubisz, I. Gronau, and A. Siepel, 2014 Genome-wide inference of ancestral recombination graphs. PLOS Genetics **10**: e1004342.

Ringbauer, H., G. Coop, and N. H. Barton, 2017 Inferring recent demography from isolation by distance of long shared sequence blocks. Genetics **205**: 1335–1351.

Robinson, D., and L. Foulds, 1981 Comparison of phylogenetic trees. Mathematical Biosciences **53**: 131–147.

Robinson, D. F., and L. R. Foulds, 1979 Comparison of weighted labelled trees. In A. F. Horadam and W. D. Wallis, editors, *Combinatorial Mathematics VI*. Springer Berlin Heidelberg, Berlin, Heidelberg, 119–126.

Rudin, W., 1976 *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill.

Silcocks, M., A. Farlow, A. Hermes, G. Tsambos, H. R. Patel, S. Huebner, G. Baynam, M. R. Jenkins, D. Vukcevic, S. Easteal, S. Leslie, The National Centre for Indigenous Genomics, A. Farlow, A. Hermes, H. R. Patel, S. Huebner, G. Baynam, M. R. Jenkins, S. Easteal, and S. Leslie, 2023 Indigenous Australian genomes show deep structure and rich novel variation. Nature **624**: 593–601.

Song, Y. S., and J. Hein, 2005 Constructing minimal ancestral recombination graphs. Journal of Computational Biology **12**: 147–169.

Speidel, L., M. Forest, S. Shi, and S. R. Myers, 2019 A method for genome-wide genealogy estimation for thousands of samples. Nature Genetics **51**: 1321–1329.

Stephens, M., and P. Donnelly, 2000 Inference in molecular population genetics. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **62**: 605–635.

Tsambos, G., J. Kelleher, P. Ralph, S. Leslie, and D. Vukcevic, 2023 link-ancestors: fast simulation of local ancestry with tree sequence software. Bioinformatics Advances **3**: vbad163.

Wang, L., K. Zhang, and L. Zhang, 2001 Perfect phylogenetic networks with recombination. Journal of computational biology : a journal of computational molecular cell biology **8**: 69–78.

Wohns, A. W., Y. Wong, B. Jeffery, A. Akbari, S. Mallick, R. Pinhasi, N. Patterson, D. Reich, J. Kelleher, and G. McVean, 2022 A unified genealogy of modern and ancient genomes. Science **375**: eabi8264.

Wong, Y., A. Ignatieva, J. Koskela, G. Gorjanc, A. W. Wohns, and J. Kelleher, 2024 A general and efficient representation of ancestral recombination graphs. Genetics **228**: iyae100.

Yang, S., S. Carmi, and I. Pe'er, 2016 Rapidly registering identity-by-descent across ancestral recombination graphs. Journal of Computational Biology **23**: 495–507.

Zhang, B. C., A. Biddanda, Á. F. Gunnarsson, F. Cooper, and P. F. Palamara, 2023 Biobank-scale inference of ancestral recombination graphs enables genealogical analysis of complex traits. Nature Genetics **55**: 768–776.

Initial	Iteration 1	Iteration 2	Iteration 3	Iteration 4
113463	76301	76059	76057	76057
113266	76084	75835	75833	75833
114489	76703	76436	76434	76434
114086	76550	76294	76291	76291

Table S1 Number of edges for each iteration of *extend haplotypes* applied to simulated tree sequences with sample size 10^4 and length 5×10^7. Each of the simulations terminated by the fourth iteration, however 99% of the edges are removed within the first iteration.

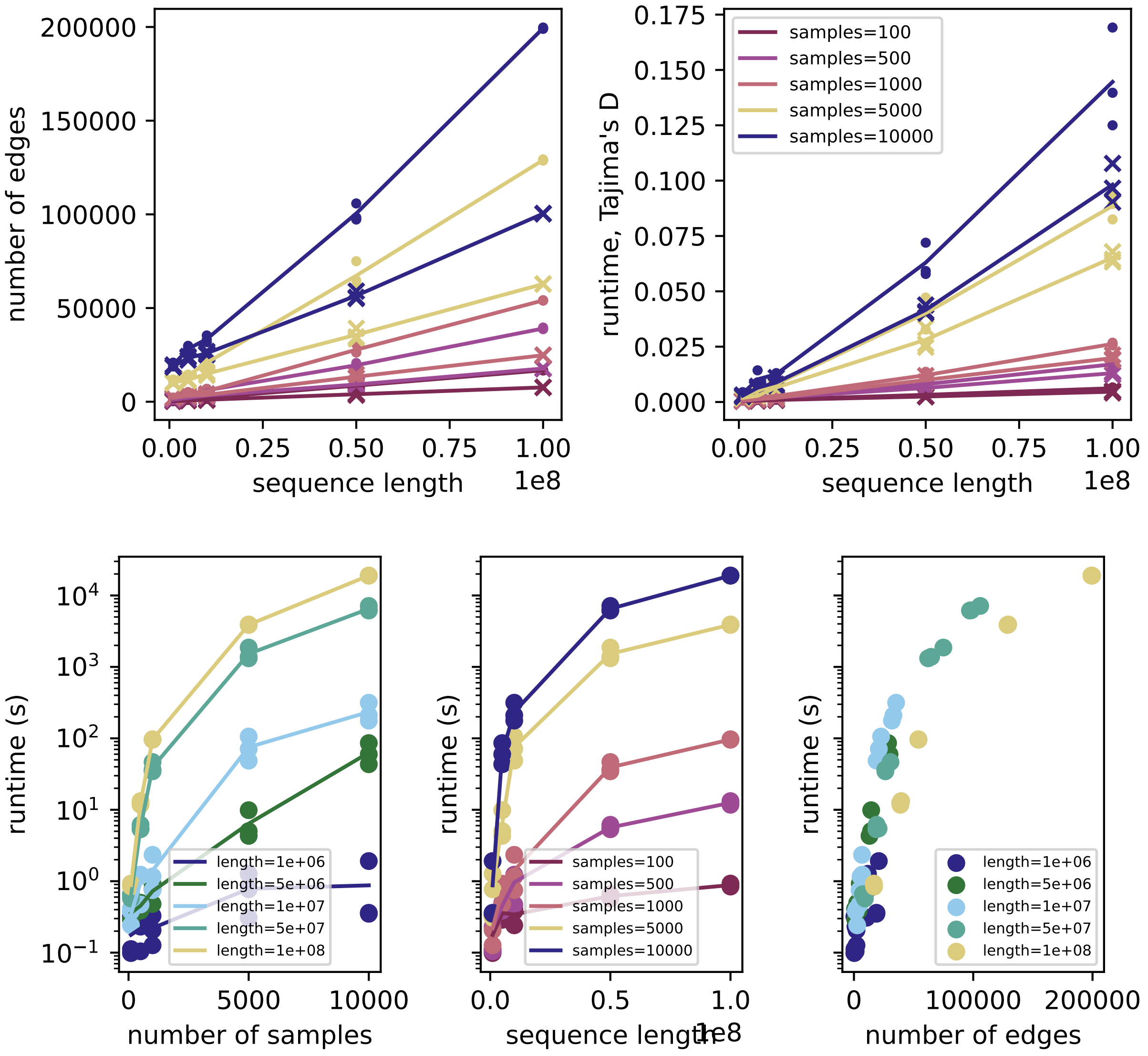

Figure S1 (Above:) Absolute values for ratios shown in Figure 4: **(left)** numbers of edges; and **(right)** runtime. **(Below:)** runtime for the `extend_haplotypes` implementation of Algorithm 1 provided in the `tskit` library. For other details, see Figure 4.

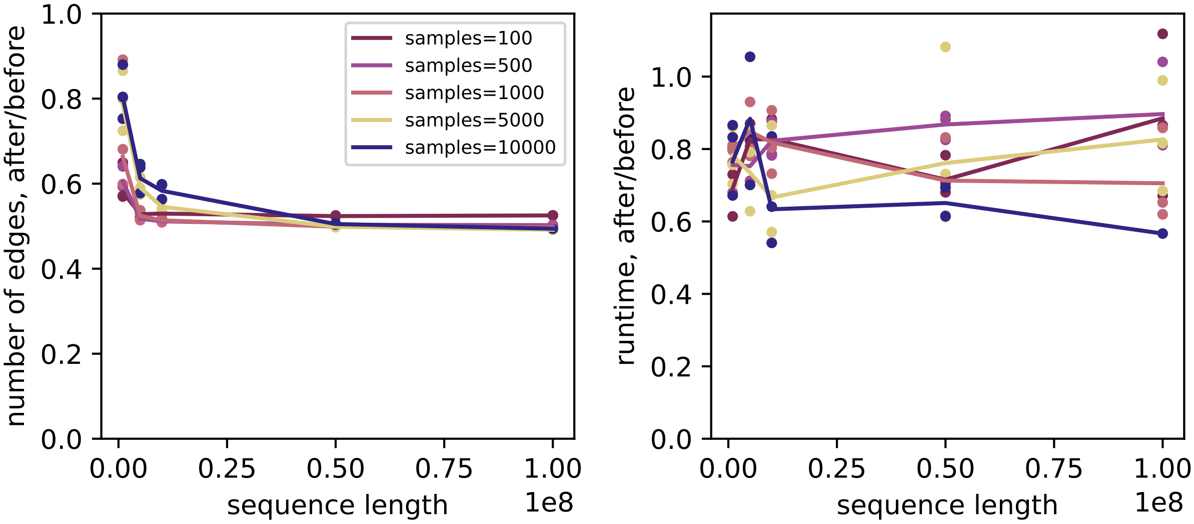

Figure S2 As in Figure 4 except that the original tree sequence was simulated with an constant population of size 10^4 (using the "constant dog" scenario); see Methods for details. Absolute values are shown in Supplementary Figure S3.

	ARF					TPR				
Sample Size	10	50	100	500	1000	10	50	100	500	1000
I	0.3808	.3490	0.3218	0.2734	0.2552	0.2904	0.2873	0.2908	0.3099	0.3299
IE	0.4518	0.3932	0.3584	0.2932	0.2726	**0.2917**	**0.2885**	**0.2916**	**0.3110**	**0.3309**
IS	**0.1993**	**0.1802**	**0.1701**	**0.1577**	**0.1516**	0.2717	0.2703	0.2732	0.2909	0.3109
ISE	0.3129	0.2779	0.2587	0.2332	0.2189	0.2850	0.2819	0.2848	0.3023	0.3228
Length	1e6	5e6	1e7	3e7	5e7	1e6	5e6	1e7	3e7	5e7
I	0.1383	0.2167	0.2324	0.2525	0.2595	0.5259	0.4478	0.4108	0.3540	0.3317
IE	0.1474	0.2289	0.2469	0.2702	0.2754	**0.5263**	**0.4485**	**0.4117**	**0.3551**	**0.3326**
IS	**0.0948**	**0.1359**	**0.1407**	**0.1485**	**0.1526**	0.5172	0.4331	0.3939	0.3356	0.3118
ISE	0.1193	0.1827	0.1960	0.2140	0.2216	0.5232	0.4432	0.4051	0.3481	0.3234

Table S2 Values for 'I', 'IE', 'IS', 'ISE' in Figure 6.

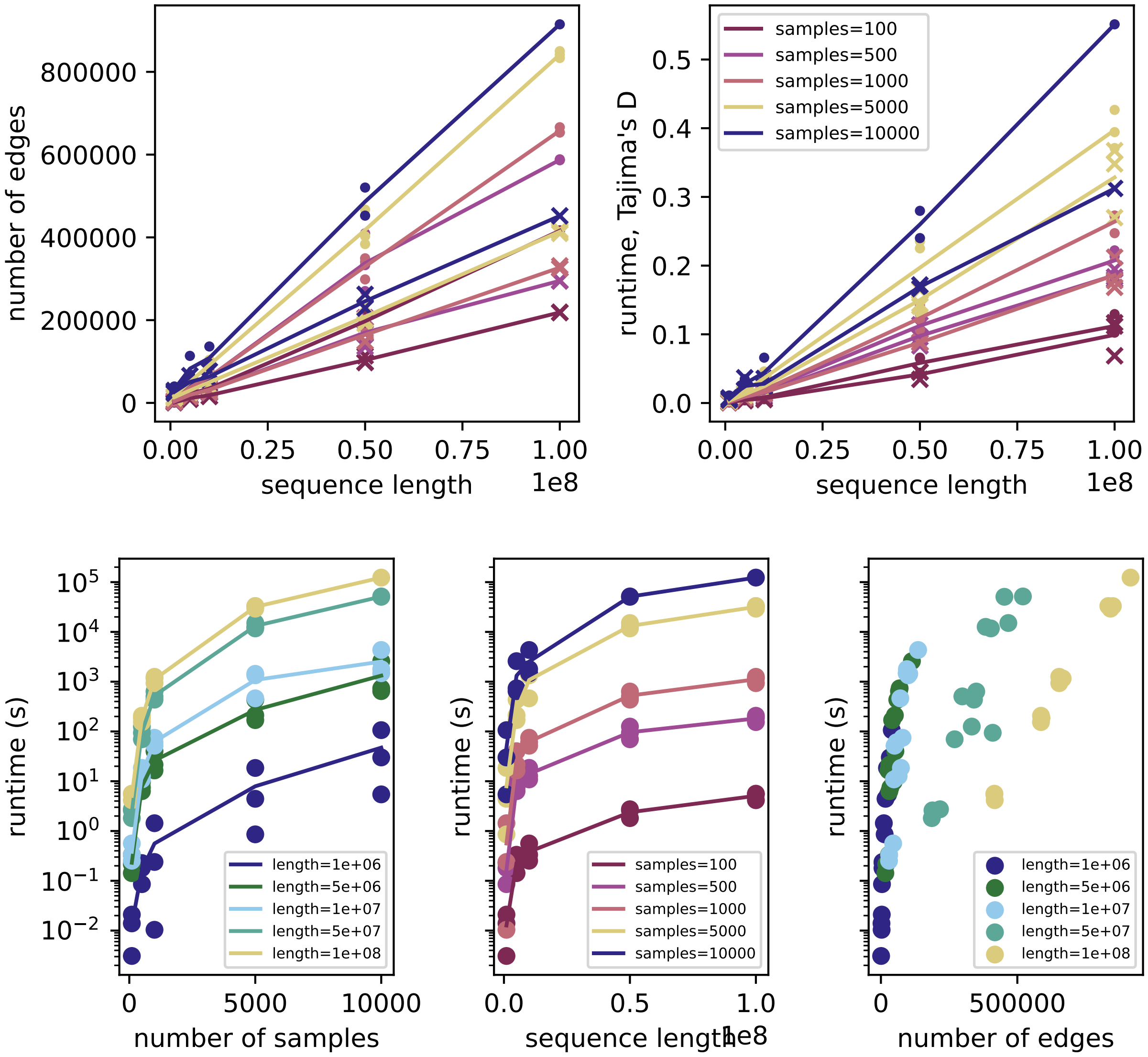

Figure S3 As in Figure S1 except that the original tree sequence was simulated with an constant population of size 10^4 (using the "constant dog" scenario).

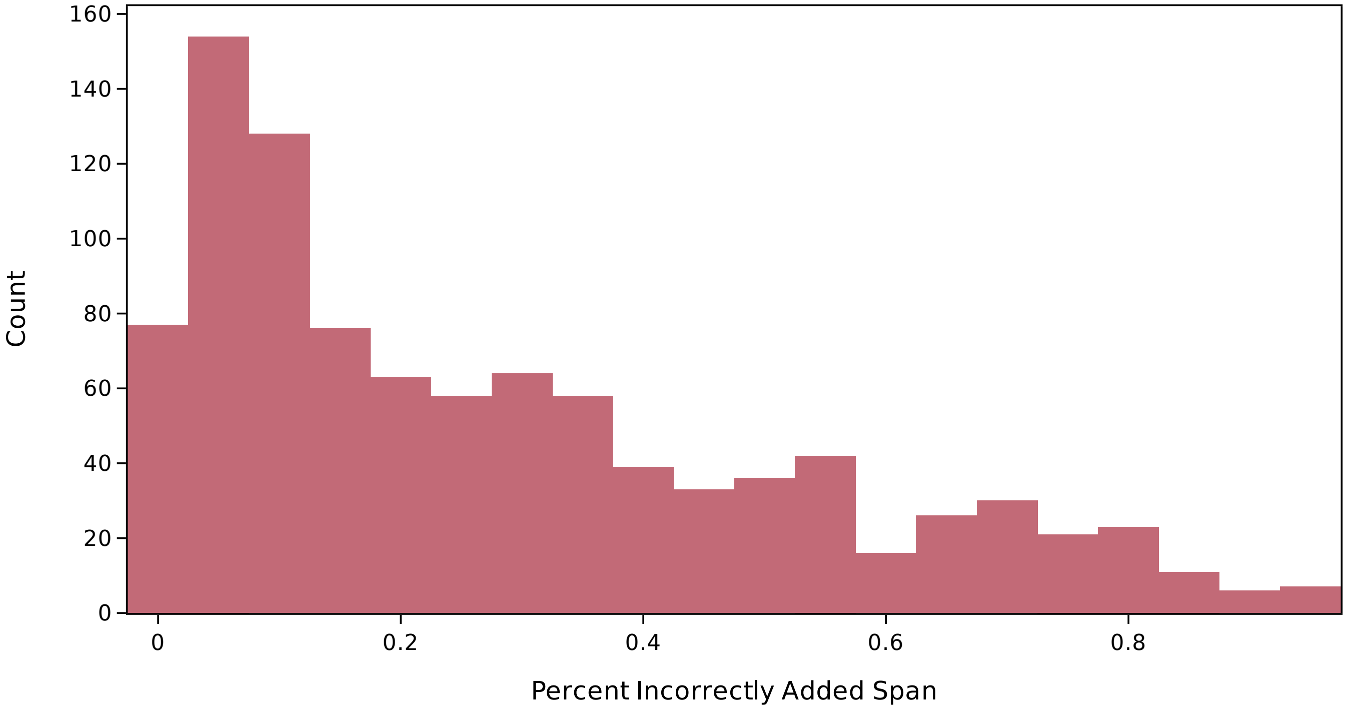

Figure S4 Distribution of incorrectly added span percentages for the simulated tree sequence in Figure 5.