Inference of Population Admixture Network from Local Gene Genealogies: a Coalescent-based Maximum Likelihood Approach

Running Title: Inference of Admixture Networks with Local Genealogies

Yufeng Wu
Department of Computer Science and Engineering
University of Connecticut
Storrs, CT 06269, U.S.A.

Abstract

Population admixture is an important subject in population genetics. Inferring population demographic history with admixture under the so-called admixture network model from population genetic data is an established problem in genetics. Existing admixture network inference approaches work with single genetic variation sites. While these methods are usually very fast, they don't fully utilize the information (e.g., linkage disequilibrium or LD) contained in population genetic data. In this paper, we develop a new admixture network inference method called GTmix. Different from existing methods, GTmix works with local gene genealogies that can be inferred from population haplotypes. Local gene genealogies represent the evolutionary history of sampled alleles and contain the LD information. GTmix performs coalescent-based maximum likelihood inference of admixture networks with the inferred genealogies based on the well-known multispecies coalescent (MSC) model. GTmix utilizes various techniques to speed up likelihood computation on the MSC model and optimal network search. Our simulations show that GTmix can infer more accurate admixture networks with much smaller data than existing methods, even when these existing methods are run with much larger data. GTmix is reasonably efficient and can analyze genetic datasets of current interests.

Keywords. Population admixture, population genetics, coalescent theory, multispecies coalescent, maximum likelihood inference.

1 Introduction

Population demographic history is a complex interplay of various processes, such as population divergence and isolation, population size changes, migration and admixture. The simplest population demographic history model is the population tree model (similar to phylogenetic tree for species evolution on the high level), which only models population divergence. In practice, however, the population tree model is often too simplistic for most population genetic study. One key missing aspect by the population tree model is gene flow. In this paper, we focus on population admixture, which is one of the most important types of gene flow. Population admixture is often so widespread that admixture has to be addressed in most demographic history studies. Human populations, for example, are known to be strongly affected by admixture throughout the human history [21, 18]. When admixture is considered, population history becomes a network (called admixture network). In addition to modeling population divergence, admixture network has admixture nodes that model admixture events. See Figure 1(a) for an illustration of admixture network.

Since demographic history is not directly observable, it is highly desirable to infer population history with admixture (i.e., admixture network) from extant population genetic data [20, 13]. Inferring admixture networks from population genetic data is challenging computationally. First, the space of admixture networks can be very large even for moderate number (say ten) of populations and small number (say two) of admixture events[1]. Moreover, the effect of admixture on population genetic data is subtle and is not easily detectable. Meiotic recombination further complicates the situation by breaking the linkage between genetic variation sites.

A natural population genetic model for admixture network inference is the coalescent model [11]. Coalescent process determines stochastically how sampled alleles coalesce in a population. Since there are multiple populations in an admixture network, the underlying coalescent is the multispecies coalescent (MSC). Multispecies coalescent is the fundamental genetic model for the study of multiple populations (species) [23]. Multispecies coalescent can be extended to allow gene flow (see, e.g. [32]). Under the multispecies coalescent model, one can (at least in principle) perform likelihood-based inference of admixture networks. However, inference based on multispecies coalescent is computationally intensive [23, 4]. Due to these difficulties, existing methods for inferring admixture networks usually perform inference with allele frequencies at individual genetic variation sites on somewhat simplified genetic models. For example, the TreeMix approach [20] infers admixture networks from allele frequency data based on a Gaussian approximation of genetic drift. MixMapper [13], another method for network inference, has similar high-level approach as TreeMix. Both TreeMix and MixMapper are fast to handle large genetic data. There are several downsides of these existing methods. First, only working with allele frequencies at individual sites may potentially lose information, especially the linkage disequilibrium (LD) among nearby sites. Moreover, these approaches are based on approximations of the underlying genealogical process, which may not very accurately model the MSC process. Our experience indicates that TreeMix and MixMapper, while useful, do not provide very accurate inference results in some of our simulations.

In this paper, we present a new method for inferring admixture networks from population genetic data. Our new method is named GTmix (which stands for Gene Tree based admixture network inference). The following summarizes the main features of GTmix.

1. GTmix works with haplotypes, rather than individual variation sites. This allows GTmix to exploit the LD information from genetic data, which is not considered by TreeMix and MixMapper. GTmix doesn't directly perform inference from haplotypes. Instead, GTmix takes input in the form a set of gene genealogies $\mathcal{T} = (T_1, T_2, \ldots, T_m)$ that are *inferred* from the given haplotypes. Here, T_i is the inferred gene genealogy at a site s_i, which represents

[1] There are over 34 million rooted binary trees with ten taxa. Adding two admixture nodes will lead to at least 3.4×10^{11} distinct networks.

the evolutionary history of haplotypes at s_i. Due to recombination, genealogies at different loci may be different. Gene genealogy is arguably more informative than individual genetic variation sites. A traditional view in population genetics is that while gene genealogy is informative, there is little power in inferring gene genealogies from genetic data largely due to recombination (see, e.g., [27]). Recently, however, local genealogy inference is being actively studied and applied in population genetics. There are several existing tools for inferring gene genealogies from population haplotypes [22, 15, 24, 9]. Among these tools, RENT+ [15] is efficient and easy to use. So we use RENT+ for local genealogy inference in this paper.

2. GTmix is a maximum likelihood approach: it aims at finding an admixture network $\mathcal{N}_{opt}$ such that $Pr(\mathcal{T}|\mathcal{N}_{opt})$ is maximized. Here, $Pr(\mathcal{T}|\mathcal{N}) = \prod_{i=1}^{m} Pr(T_i|\mathcal{N})$ (based on the assumption the independence of each T_i). $Pr(T_i|\mathcal{N})$ is the probability of observing genealogy T_i for the admixture network $\mathcal{N}$ under the multispecies coalescent with admixture (MSCA) model. While in general computing $Pr(\mathcal{T}|\mathcal{N})$ under the MSCA model is computationally challenging, GTmix takes advantage of the recent algorithmic progress in probability computation on the MSC model [29, 31, 19], which allows such probability to be computable in practice for medium size data. By working on the MSCA model, GTmix performs admixture network inference on an arguably more rigorous model than existing methods (such as TreeMix).

3. GTmix can be configured to infer admixture networks with arbitrary number of admixture events, although the running time will increase with the number of admixture events. It can infer recent admixture (i.e., that forms an extant population) and also ancient admixture (i.e., that forms an ancestral population). GTmix is reasonably efficient computationally. In Phylogenetics, there exists methods (most notably the program Phylonet [26]) for inferring phylogenetic networks (somewhat related to admixture networks) based on the MSC model. Our experience indicates that GTmix can perform likelihood based inference on data with significantly larger number of taxa and alleles than that allowed by Phylonet.

Through simulation, we show that while GTmix is not as fast as existing methods such as TreeMix, GTmix is more accurate than existing methods in many simulations. In particular, GTmix can infer more accurate admixture networks under many settings using much smaller amount of data than existing methods, even when those existing methods use much larger data than GTmix. Our results suggest that inferred genealogies can indeed be informative for population demographic history inference. Also coalescent-based probabilistic inference with local genealogies can be potentially more accurate than existing methods using single variation sites.

2 Background

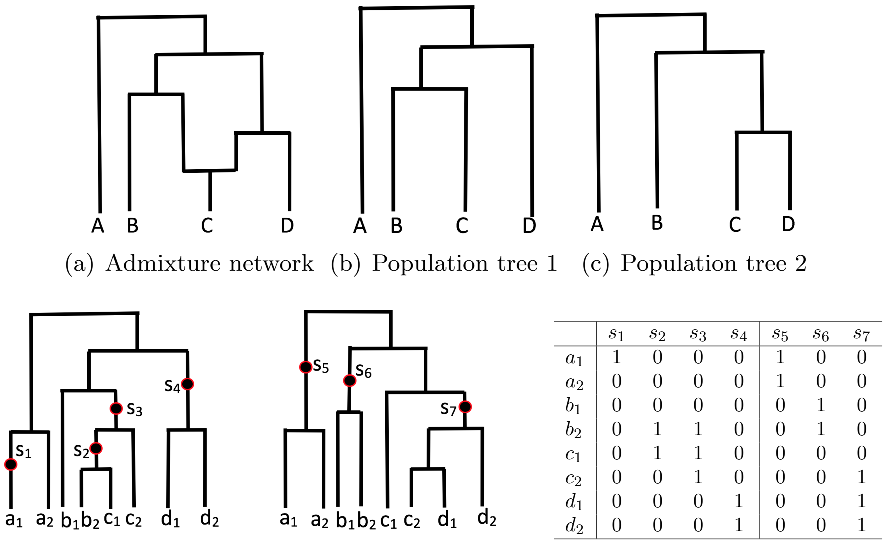
2.1 Admixture network

Admixture network is conceptually similar to phylogenetic network as studied in the Phylogenetics literature [8, 16]. Admixture is a natural extension to the population tree model (see, e.g., [30]). Admixture network (or simply network) is a directed acyclic graph $\mathcal{N}$, and is leaf-labeled by a set of populations. Moreover,

1. Admixture nodes are those nodes in $\mathcal{N}$ with two incoming edges. The two ancestral nodes of an admixture node are called the source populations.

2. For a network $\mathcal{N}$, when only *one* of the incoming edges of each admixture node is kept and the other is deleted, we always obtain a tree T. This tree is called population tree.

Note that in admixture network, a node refers to a population. An admixed node has two incoming edges, which originate from two source populations for the admixed population. So for an



(a) Admixture network (b) Population tree 1 (c) Population tree 2

	s_1	s_2	s_3	s_4	s_5	s_6	s_7
a_1	1	0	0	0	1	0	0
a_2	0	0	0	0	1	0	0
b_1	0	0	0	0	0	1	0
b_2	0	1	1	0	0	1	0
c_1	0	1	1	0	0	0	0
c_2	0	0	1	0	0	0	1
d_1	0	0	0	1	0	0	1
d_2	0	0	0	1	0	0	1

(d) Gene tree 1 (e) Gene tree 2 (f) Haplotypes

Figure 1: *Illustration of admixture network. Network is shown in part 1(a). Four populations. C is admixed. Two population trees in this network are shown in parts 1(b) and 1(c). Parts 1(d) and 1(e): two gene trees at different loci. Two sampled alleles per population: alleles a_1 and a_2 are from the population A, alleles b_1 and b_2 are from the population B and so on. Darkened dots: mutations. Mutations: follow infinite sites model. Part 1(f): haplotypes for the two gene trees.*

admixture node v in a network, there are two ancestral nodes: the left parent (denoted as $Left(v)$) and the right parent (denoted as $Right(v)$). Throughout this paper, we assume each admixture is two-way (i.e., formed by two ancestral populations). More general admixture (involving three or more ancestral populations) can in principle be converted to a chain of two-way admixture events.

Population tree Recall that a population tree T is contained in a network $\mathcal{N}$ when we remove one of the two incoming edges at each admixture node. Suppose there are n_a admixture nodes in $\mathcal{N}$. There are 2^{n_a} population trees contained in $\mathcal{N}$. In Figure 1, there are two population trees contained in the network, as shown in Figures 1(d) and 1(e). Note that population trees don't capture all the possible demographic histories in $\mathcal{N}$. To see this, look at the network in Fig. 1(a) and the two population trees in Figures 1(b) and 1(c). Here, C is either sibling of B or D in a population tree. However, this doesn't capture the possible scenarios that, for example, one lineage from C coalesces with a lineage from B, while the other from C coalesces with a lineage from D. Despite this shortcoming, GTmix relies on population trees in computing the coalescent likelihood because it is computationally more efficient to work with population trees, instead of the network.

Branch length Branches in an admixture network have lengths in the standard coalescent units. This accommodates the case where different branches have different population sizes. In this paper, we assume population sizes remain constant in a branch of $\mathcal{N}$.

Admixture proportions At each admixture node v, there is an admixture proportion $f_0(v)$ where $0 \leq f_0(v) \leq 1$. $f_0(v)$ refers to the proportion of alleles of population v originating from the left parent $Left(v)$. We define $f_1(v) = 1.0 - f_0(v)$ as the proportion of alleles of v from the right parent $Right(v)$. Admixture proportions determine the probability of individual population trees in $\mathcal{N}$.

Genealogy and multispecies coalescent. Gene genealogy and multispecies coalescent are central to the GTmix method. Due to the lack of the space, background information on genealogy and multispecies coalescent is given in the Supplemental Materials.

3 Method

3.1 The high-level approach

In this paper, we develop a new admixture network inference tool, called GTmix. We assume haplotypes $\mathcal{H}$ from n_p populations are given. We further assume the number of admixture events n_a is known. Our goal is inferring the maximum likelihood admixture network $\mathcal{N}$ with n_a admixture nodes for these populations from $\mathcal{H}$. One natural approach is finding $\mathcal{N}$ that maximizes the likelihood of $\mathcal{H}$. However, computing the likelihood $P(\mathcal{H}|\mathcal{N})$ is challenging. GTmix uses the following techniques to develop a practical coalescent-likelihood based inference approach.

1. GTmix works with local genealogies $\mathcal{T}$ which are inferred from $\mathcal{H}$, not $\mathcal{H}$ directly. Before using GTmix, we first use a local genealogy inference tool (e.g., RENT+ [15]) to infer local gene genealogies $\mathcal{T}$ from $\mathcal{H}$. The main benefit of working with inferred genealogies is that computing the likelihood of $\mathcal{T}$ is easier than directly computing the likelihood of $\mathcal{H}$ for $\mathcal{N}$; also local genealogies $\mathcal{T}$, while often noisy, capture the important LD information, where single SNP sites don't. The number of genealogies can be very large in practice. GTmix uses a filtering scheme to choose more reliable gene genealogies for inference.

2. GTmix takes the inferred $\mathcal{T}$ to infer $\mathcal{N}_{opt}$ that maximizes $Pr(\mathcal{T}|\mathcal{N}_{opt})$. $Pr(\mathcal{T}|\mathcal{N})$ is computed approximately by summing up the probability of $\mathcal{T}$ for each population tree in $\mathcal{N}$. The latter probability is computed using the fast gene tree probability algorithm in [19]. This simplified probability can be computed relatively fast when the network is not too large. GTmix implements effective methods to perform admixture network search.

3.2 Inferring local gene genealogies

GTmix takes a list of local gene genealogies $\mathcal{T}$ as input. $\mathcal{T}$ can be inferred by applying a local genealogy inference tool on the given haplotypes $\mathcal{H}$. GTmix is tested with genealogies inferred by the program RENT+ [15], although other tools (e.g., [24, 9]) can be used as well. Gene genealogies are inferred from haplotypes of each locus separately. RENT+ infers a (rooted binary) genealogy for each SNP site within the locus. The total number of genealogies can be very large because there can be different genealogy at each SNP. This makes GTmix infeasible to perform inference on all genealogies since computing the probability of a genealogy is not trivial. To develop a practical method, GTmix samples a subset of genealogies (trees) using a two-stage approach: (i) it first chooses a subset of (likely more reliable) trees from each locus, and (ii) it then chooses a smaller subset of trees from the list of trees from all loci. Due to the lack of space, the details are given in the Supplemental Materials.

3.3 Gene tree probability computation for admixture network

GTmix takes a list of inferred genealogies $\mathcal{T} = \{T_1, \ldots, T_K\}$ as input. The most important aspect for inferring admixture network $\mathcal{N}$ is computing $P(\mathcal{T}|\mathcal{N})$ on the multispecies coalescent with admixture (MSCA) model. Assuming the independence of genealogies, $P(\mathcal{T}|\mathcal{N}) = \prod_{i=1}^{K} P(T_i|\mathcal{N})$. Computing $P(T_i|\mathcal{N})$ for a network $\mathcal{N}$ is not trivial (see, e.g., [32]). The main difficulty for computing $P(T_i|\mathcal{N})$ is that the number of feasible coalescent histories can be very large under the MSC model. To develop a practical method that can work with relatively large data, GTmix takes an approximation here: instead of computing the probability of T_i for $\mathcal{N}$, GTmix computes the probabilities of T_i for each population tree in $\mathcal{N}$ and then use the sum of these probability as an approximation of $P(T_i|\mathcal{N})$. While this is only an approximation, our experience indicates this allows reasonably accurate and efficient inference of admixture network.

To be specific, we consider all possible population trees in $\mathcal{N}$. Each population tree is obtained by picking one of the two incoming admixture edges at admixture node v with probabilities m_v

and $1.0 - m_v$ respectively. Let the embedded population trees within $\mathcal{N}$ be $\mathcal{T}_{\mathcal{N}} = \{T_{\mathcal{N},1}, \ldots, T_{\mathcal{N},k}\}$. Let n_a be the number of admixture notes in $\mathcal{N}$. A population tree $T_{\mathcal{N},j}$ is associated with a binary vector D_j of length n_a. $D_j[a] = 0$ (respectively $D_j[a] = 1$) if the left (respectively right) incoming edge is chosen to obtain $T_{\mathcal{N},j}$. Here, each $T_{\mathcal{N},j}$ has a probability of being the population tree: $P(T_{\mathcal{N},j}) = \prod_{i=1}^{n_a} m_{v_i}^{1-D_j(i)}(1.0 - m_{v_i})^{D_j(i)}$. Here, v_i is the i-th admixture node. Recall that branches in $\mathcal{N}$ have lengths. So trees in $\mathcal{T}_N$ have branch lengths. In contrast, we assume genealogical trees in $\mathcal{T}$ are topologies only. Thus,

$$P(T_i|\mathcal{N}) \approx \prod_{j=1}^{2^{n_a}} P(T_{\mathcal{N},j}) P(T_i|T_{\mathcal{N},j}) \tag{1}$$

$P(T_i|T_{\mathcal{N},j})$ is exactly the gene tree probability of a gene tree topology for a fixed population tree (with branch lengths). GTmix uses the approximate gene tree probability algorithm in [19]. This is because this algorithm is much more scalable than exact gene tree probability algorithms in [4, 29, 31]. Note that Equation 1 is only an approximation. It assumes all lineages at an admixture node are inherited from a single parent. In reality, when there are multiple lineages from an admixed population, different lineages may trace to different parental populations of this population. To see this, we again look at the network in Fig. 1(a) and the gene tree in Fig. 1(d). Suppose at the time of admixture, lineages c_1 and c_2 remain un-coalesced. Then, Equation 1 ignores the situation where, for example, c_1 is from the left and c_2 is from the right. We adopt this approximation because: (i) computing gene tree probability for a population tree is much faster than computing the gene tree probability for a network, and (ii) empirical tests suggest that this approximation appears to provide reasonably accurate inference.

3.4 Finding the maximum likelihood admixture network

GTmix assumes the number of admixture events n_a in the network is known. Note that existing network inference methods such as TreeMix and MixMapper also make the same assumption. GTmix finds the maximum likelihood admixture network using a simple iterative procedure:

1 Construct an initial network $\mathcal{N}_{init}$ with n_a admixture nodes from the inferred gene genealogies $\mathcal{T}$. Let $\mathcal{N}_{opt} \leftarrow \mathcal{N}_{init}$.

2. Find the set of admixture networks $\mathcal{S}$ that are similar topologically to $\mathcal{N}_{opt}$ and have n_a admixture nodes.

3. Let $\mathcal{N} \in \mathcal{S}$ that maximizes the likelihood $P(\mathcal{T}|\mathcal{N})$. If $P(\mathcal{T}|\mathcal{N}) > P(\mathcal{T}|\mathcal{N}_{opt})$, set $\mathcal{N}_{opt} \leftarrow \mathcal{N}$, and go to step 2. Otherwise, stop.

3.4.1 Constructing the initial network

GTmix constructs an initial network $\mathcal{N}_{init}$ from a set of genealogies $\mathcal{T}$ as follows. It first constructs a network $\mathcal{N}_0$ without admixture nodes (i.e., a population tree) by neighbor joining (NJ). Then, GTmix adds n_a admixture nodes to $\mathcal{N}_0$, which gives the initial network $\mathcal{N}_{init}$. To run the NJ algorithm, GTmix estimates the pairwise distance between each pair of populations p_i and p_j based on topological information in $\mathcal{T}$. This is shown in Algorithm 2 (in the Supplemental Materials). The pairwise population distance between p_i and p_j is estimated based on the average distance between two sample alleles from p_i and p_j (in terms of the number of edges separating these two alleles on an inferred genealogy).

GTmix now adds n_a admixture nodes to $\mathcal{N}_0$ by choosing proper leaf nodes in $\mathcal{N}_0$ and turning them into admixture nodes. That is, the initial admixture populations are the extant populations in $\mathcal{N}_0$. Note that ancestral admixture events (i.e., admixture at ancestral populations) are accommodated during the network search stage. GTmix relies on the so-called minimum deep coalescent (MDC) [17, 14] to identify likely admixed extant populations. Briefly, MDC is a statistic that measures the topological deviation of the given gene trees from a consensus tree based on the MSC model. Refer to [17, 14] for more details on MDC. While MDC is known to have issues in phylogenetics (see, e.g., [5]), we choose to use MDC due to its simplicity and efficiency.

The key idea of inferring likely admixed extant populations by GTmix is that admixed extant populations tend to make gene trees *deviate* significantly from the standard MSC process. This is because the standard MSC process doesn't consider admixture. Thus, intuitively, an admixed population tends to lead to gene trees that are quite different from gene trees arising from the standard MSC process. Suppose we discard all alleles from the admixed population from the gene trees, the (reduced) gene trees tend to have a better fit to the standard MSC process. GTmix uses the MDC statistic to measure how well gene trees fit the standard MSC process. More specifically, suppose a population p is a candidate for admixed extant population. We first compute the MDC score (denoted as MDC) of the original $\mathcal{T}$. Then we discard all gene lineages from the population p in all trees in $\mathcal{T}$. We call the reduced genealogies $\mathcal{T}'$. We then compute the MDC score (denoted as MDC_p) on $\mathcal{T}'$. Since each tree in $\mathcal{T}'$ is a subtree of the corresponding tree in $\mathcal{T}$, $MDC_p \leq MDC$. We expect $MDC - MDC_p$ for admixed p to be significantly larger than $MDC - MDC_{p_0}$ for an un-admixed population p_0. This leads to Algorithm 1 for finding likely admixed extant populations.

Algorithm 1 Finding likely admixed extant populations from genealogies $\mathcal{T}$

```
Let $\mathcal{P}$ be the set of all populations.
for $k = 1 \ldots n_a$ do
    Compute the MDC score (denoted as $MDC$) for $\mathcal{T}$.
    for each $p \in \mathcal{P}$ do
        Construct reduced genealogies $\mathcal{T}_p$ by discarding lineages from $p$ (and cleaning up the trees so that trees remain to be binary).
        Compute the MDC score (denoted as $MDC_p$) on $\mathcal{T}_p$
    end for
    Let $p_a \leftarrow argmax_{p \in P}(MDC - MDC_p)$. $\mathcal{P} = \mathcal{P} - \{p_a\}$. $\mathcal{T} \leftarrow \mathcal{T}_{p_a}$. Add $p_a$ to $\mathcal{P}_a$ (the list of likely admixed extant populations).
end for
```

Constructing initial network $\mathcal{N}$ by adding admixture nodes For each likely admixed extant population $p_a \in \mathcal{P}_a$, GTmix modifies $\mathcal{N}_0$ by adding one admixed branch into p_a to make p_a an admixture node. GTmix considers all possible admixture source populations that can be ancestral to p_a according to the current network. Each choice leads to a network $\mathcal{N}_{p_a}$. Then it chooses one source population that gives the maximum probability $P(\mathcal{T}|\mathcal{N}_{p_a})$. After this step, we obtain the initial network $\mathcal{N}_{init}$.

3.4.2 Search over the space of networks

GTmix searches for the set of topologically similar networks $\mathcal{S}$ from the current network N_{opt}. Networks in $\mathcal{N}$ are obtained by applying one nearest neighbor interchange (NNI) to the current network. More specifically, consider each branch $b = (v_p, v_c)$ in $\mathcal{N}_{opt}$, where node v_p is the parent of node v_c. Let v_s be the sibling of v_c, and v_{sp} be the sibling of v_p. Then after applying one NNI on b, we obtain a new network with v_{sp} being the sibling of v_c and v_s being the sibling of v_p. That is, the NNI swaps v_s with v_{sp} which leads to a new network $\mathcal{N}'$. For each network $\mathcal{N}' \in \mathcal{S}$, GTmix uses hill climbing to find the network that maximizes the likelihood.

Simulation Due to the lack of space, simulation procedure is given in the Supplemental Materials.

4 Results

Due to the lack of space, some results are provided in the Supplemental Materials.

4.1 Results on simulated data

At present, GTmix cannot run on data with large number of populations and/or large number of alleles per population. Thus, we run GTmix with relatively small number of alleles per population (by default, four alleles per population). Existing methods such as TreeMix and MixMapper are designed to work with data with relatively large number of alleles. For comparison, we run TreeMix and MixMapper on two settings: (i) the "small" data: the same data as given to GTmix, and (ii) the "large" data: simulated data with much larger number of alleles per population (by default, 100 alleles per population, i.e., 25 times of the small data). Note that the number of combinations of different parameters is very large. In the following, when we evaluate the effect of a particular parameter, we keep all other parameters to be the default values.

4.1.1 Metrics for benchmarking the accuracy of admixture network inference

In this paper, we use the following two metrics for comparing the topology of $\mathcal{N}$ and that of $\mathcal{N}'$.

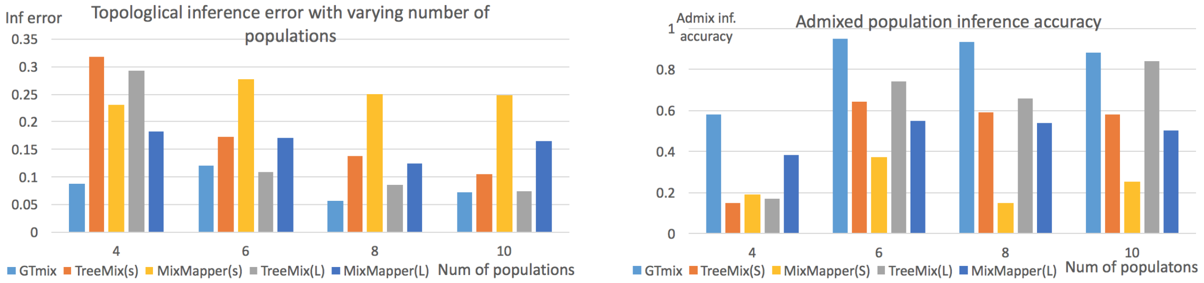
1. Best-match population tree inference error. Recall that a network $\mathcal{N}$ with n_a admixture nodes contains 2^{n_a} population trees. Assuming $\mathcal{N}$ and $\mathcal{N}'$ have the same number of admixture nodes, we compare the list of 2^{n_a} population trees $\mathcal{T}$ in $\mathcal{N}$ with 2^{n_a} population trees $\mathcal{T}'$ in $\mathcal{N}'$. Here we need to match each tree $T \in \mathcal{T}$ with a tree $T' \in \mathcal{T}'$. There are $2^{n_a}!$ ways of matching $\mathcal{T}$ with $\mathcal{T}'$. For example, when $n_a = 1$, there are two ways of matching; when $n_a = 2$, there are 24 ways of matching. The so-called Robinson-Foulds (RF) distance between a pair of matched trees T and T' is used. The RF distance is equal to the number of clades (subtrees) that are in T but not in T'. We normalize the RF distance to be between 0 and 1. For each matching, we take the average RF distance of all matched pairs of trees as the inference error. We take the smallest average RF distance over all matchings as the best-match inference error.

2. Percentage of correctly inferred admixed populations. It can be biologically important to identify which populations are admixed. This is measured by the average percentage of correctly inferred admixed populations.

In addition to topology inference, we also examine the accuracy of the inferred admixture proportions. We use the absolute difference between the inferred admixture proportion and the true admixture proportion as the inference error for admixture proportion. For each settings, we simulate ten replicate datasets. Reported results are the average over these replicates.

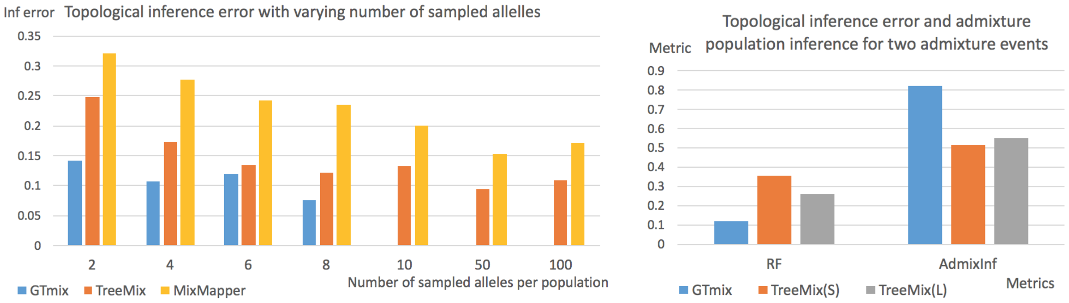
4.1.2 Network inference accuracy

Varying number of populations. We run GTmix, TreeMix and MixMapper on data with varying number of populations (from 4 to 10 populations). We sample four alleles for each population. We also run TreeMix and MixMapper with large data (100 alleles for each population). Fig. 2 shows the results. We can see that GTmix is significantly better than TreeMix and MixMapper on small data in terms of both the topology inference and admixture population inference. For topology inference, Also GTmix with small data is often more accurate in both the topology inference and the admixture population inference than TreeMix and MixMapper with large data in most cases. Note that TreeMix appears to outperform MixMapper when the number of populations increases. **Varying number of alleles per population** We evaluate the performance of GTmix when the number of alleles per population varies. Due to the computational difficulty, GTmix only runs on data with up to 8 alleles per population. For comparison, TreeMix and MixMapper are run on



(a) Topological inference error (b) Admixture population inference accuracy

Figure 2: *Admixture network inference with varying number of populations. GTmix: four alleles per population. TreeMix and MixMapper: small data (four alleles per population, denoted as "S") and large data (100 alleles per population, denoted as "L"). Part 2(a): average best-match RF distance between inferred networks and true networks. Part 2(b): average admixture population inference accuracy. X-axis: number of populations.*



(a) Varying number of alleles per population (b) Two admixture events

Figure 3: *Network inference with varying number of alleles and number of admixture events. Part 3(a): varying number of sampled alleles per population. GTmix: 2 to 8 alleles per population. That is, GTmix is not run for 10, 50 and 100 alleles. TreeMix and MixMapper: 2 to 100 alleles per population. X-axis: number of alleles per populations. Part 3(b): with two admixture events; inference error for topology and admixture population inference accuracy. Six populations. TreeMix is run with both small (S) and large (L) data.*

data with up to 100 alleles per population. Six populations are simulated. Figure 3(a) shows the results. As expected, when the number of alleles increases, all methods tend to be more accurate. On the same data, GTmix is consistently better than TreeMix and MixMapper. GTmix can be more accurate with smaller data than TreeMix and MixMapper even the latter are run with larger data. For example, the error of GTmix with 8 alleles per population is about 0.077, while the errors are about 0.11 and 0.17 respectively for TreeMix and MixMapper with 100 alleles per population.

Two admixture events By default, we simulate a single admixture event. To test the performance of GTmix on more complex demographic models, we simulate two admixture events for six populations (plus one outgroup). We compare GTmix (on small data) with TreeMix which are run on both small and large data. The results are shown in Figure 3(b). We can see that GTmix outperforms TreeMix in both topological inference and admixture population inference (on both small and large data) significantly even when the latter is run with large data.

Phasing. So far we use true simulated haplotypes. In practice, haplotypes are usually inferred from genotypes. We now evaluate the performance of GTmix on inferred haplotypes. Note that phasing accuracy can be affected by recombination rate. We simulate data with varying recombination rates (with default mutation parameter of 50). We randomly pair up two simulated haplotypes to form a genotype. Then we use the program beagle [1] to phase the gentoypes to obtain phased haplotypes. We compare the topological inference error of networks inferred with both the true haplotypes and the phased haplotypes.

The results are shown in Figure 4. When recombination rate is low, phasing error appears to have marginal effect on the inference accuracy. When recombination rate is high (say 250), the impact of recombination rate becomes larger: there is a significant reduction in inference accuracy when recombination rate is high.

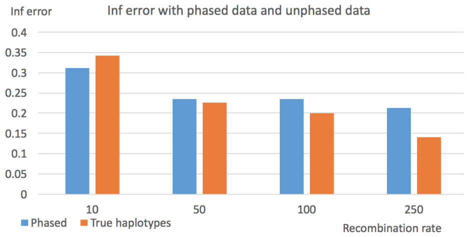
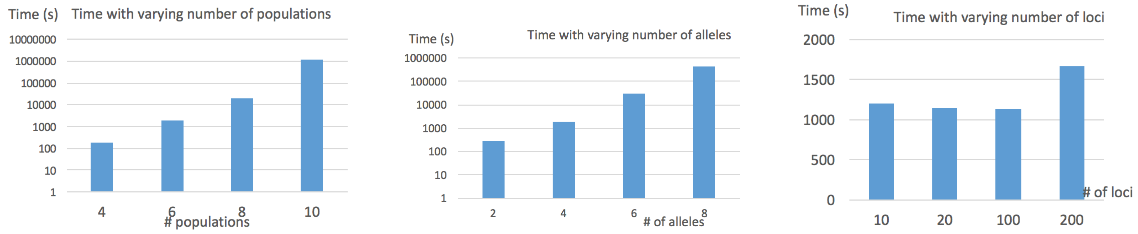


Figure 4: *Topological inference error on phased and true haplotypes. X-axis: recombination rate used in data simulation.*

4.1.3 Network inference efficiency

We show the running time of GTmix under various settings in Figure 5. The running time of GTmix apparently grows exponentially with regard to both the number of populations and the number of alleles per population. On the other hand, GTmix scales well with regard to the number of loci. Overall, although GTmix is still computationally intensive on relatively large data, GTmix can be practical for inference for many data of current interests.



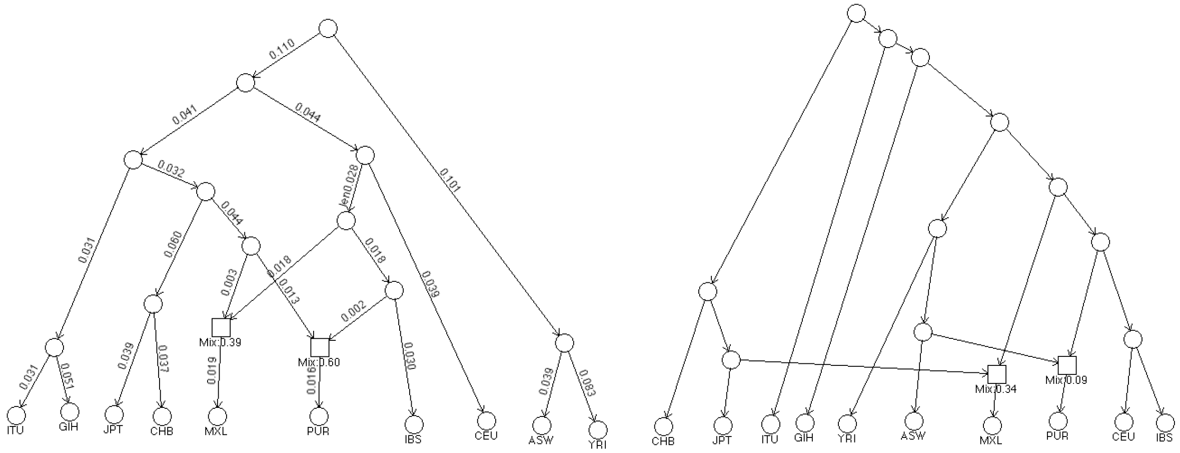
(a) Varying number of populations (b) Varying number of alleles (c) Varying number of loci

Figure 5: *Running time of GTmix with varying number of populations, number of alleles per population, and number of loci. Default: six populations and four alleles per population.*

4.2 Results on the 1000 Genomes Project data

The 1000 Genomes Project [3] has released haplotypes of 1,092 individuals from 26 populations in Phase I integrated variant set release. The 1000 Genomes Project defines five super populations: African, East Asian, European, South Asian and Admixed American. We pick two populations from each of these super populations. Namely, we pick the following ten populations: Americans of African Ancestry in SW USA (ASW), Yoruba in Ibadan, Nigeria (YRI), Han Chinese in Beijing, China (CHB), Japanese in Tokyo, Japan (JPT), Utah Residents with Northern and Western European Ancestry (CEU), Iberian Population in Spain (IBS), Gujarati Indian from Houston, Texas (GIH), Indian Telugu from the UK (ITU), Mexican Ancestry from Los Angeles USA (MXL) and Puerto Ricans from Puerto Rico (PUR). We create two test datasets, one small and one large. (i) The small data contains haplotypes from two randomly chosen diploid individuals (i.e., four alleles) from each population. We sample up to 50 loci from chromosomes 1 to 10 as follows: for each chromosome, randomly sample up to 50 regions of 100 Kb each, which are evenly located as 3 Mb apart. (ii) The large dataset contains SNPs from twenty diploid individuals (i.e., ten times as many as the small data) from the *whole* genome (i.e., all SNPs are used). To reduce noise, any SNP sites with non-binary alleles are discarded.

We run GTmix to infer admixture network with two admixture events on the small dataset only. It takes about 34 hours for GTmix to find the optimal network in a computer cluster (with 2.1 Ghz CPU). The network is shown in Fig. 6(a). For comparison, we run TreeMix with whole genome data (with 20 individuals per population). We use the whole genome data here because TreeMix can run with large data. We perform preprocessing to discard rare variants: only SNPs with minor allele frequencies of 5% or larger are kept [2]. This leaves total 7.63 million SNPs in the data. Note that this is about 50 times more data than what is used by GTmix. The resulting network is shown in Fig. 6(b). We note that the networks inferred by GTmix and TreeMix share some key topological properties. For example, both networks have MXL and PUR as admixed. On the other hand, the two networks also differ in some aspects. The network by GTmix, as expected, has IBS to be closely related to the source populations involved in both inferred admixture, while the other ancestral populations involved are closely related to CHB/JPT. While the true admixture network for these populations is not known, at least some aspects of the TreeMix's network don't agree with commonly accepted human demographic history. First, the rooting of the TreeMix network doesn't agree with the "out-of-Africa" scenario. Also, an ancestral Eastern Asian population (which is after the divergence of CHB and JPT) is involved in the admixture of MXL but not for PUR. So potentially more accurate demographic history for these aspects are given in the network by GTmix, even when TreeMix uses 50 times more genetic data than GTmix. To summarize, our results show that GTmix can provide accurate inference of admixture networks on real data.



(a) GTmix's result

(b) TreeMix's result

Figure 6: *Inferred admixture network for ten populations from the 1000 Genomes Project by GTmix and TreeMix. GTmix is run with small data with two diploid individuals per population, while TreeMix is run with whole genome data with twenty diploid individuals per population (more than 50 times more data than that used by GTmix). Square box: admixed nodes (with inferred admixture proportions; the displayed admixture proportion is for the left source population). Each branch in the GTmix result has the estimated branch length (in the standard coalescent unit).*

Different runs of GTmix and TreeMix We have conducted more simulations with the 1000 Genomes data for GTmix and TreeMix. The purpose is testing the reproducibility and and the effect of data amount. Due to the space limit, the results are given in the Supplemental Materials.
Discussions Due to the lack of space, discussion section is given in the Supplemental Materials.

[2] If rare alleles are kept, our simulation results show that the constructed network appears to be less accurate.

Software availability

The program GTmix can be downloaded from `https://github.com/yufengwudcs/GTmix`.

ACKNOWLEDGMENTS

This work is partly supported by U.S. National Science Foundation grants CCF-1718093 and IIS-1909425.

References

[1] S. R. Browning and B. L. Browning. Rapid and accurate haplotype phasing and missing data inference for whole genome association studies by use of localized haplotype clustering. *American Journal of Human Genetics*, 81:1084–1097, 2007.

[2] D. Bryant, R. Bouckaert, J. Felsenstein, N. A. Rosenberg, and A. RoyChoudhury. Inferring species trees directly from biallelic genetic markers: Bypassing gene trees in a full coalescent analysis. *Mol. Biol. Evol.*, 29:1917–1932, 2012.

[3] The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526:64–74, 2015.

[4] J. H.. Degnan and L. A. Salter. Gene tree distributions under the coalescent process. *Evolution*, 59(1):24–37, 2005.

[5] J. Felsenstein. *Inferring Phylogenies*. Sinauer, Sunderland, MA., 2004.

[6] R. C. Griffiths and P. Marjoram. Ancestral inference from samples of DNA sequences with recombination. *J. Comput. Biol.*, 3:479–502, 1996.

[7] R. Hudson. Generating samples under the Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.

[8] D. H. Huson, R. Rupp, and C. Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press, Cambridge, U. K., 2010.

[9] Jerome Kelleher, Yan Wong, Anthony W. Wohns, Chaimaa Fadil, Patrick K. Albers, and Gil McVean. Inferring whole-genome histories in large population datasets. *Nature Genetics*, 51(9):1330–1338, 2019.

[10] M. Kimura. The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations. *Genetics*, 61:893–903, 1969.

[11] J. F. C. Kingman. The coalescent. *Stochast. Process. Appl.*, 13:235–248, 1982.

[12] H. Li and R. Durbin. Inference of human population history from individual whole-genome sequences. *Nature*, 475:493496, 2011.

[13] Mark Lipson, Po-Ru Loh, Alex Levin, David Reich, Nick Patterson, and Bonnie Berger. Efficient Moment-Based Inference of Admixture Parameters and Sources of Gene Flow. *Molecular Biology and Evolution*, 30(8):1788–1802, 05 2013.

[14] W. Maddison. Gene trees in species trees. *Systematic Biology*, 46:523–536, 1997.

[15] S. Mirzaei and Y. Wu. RENT+: An improved method for inferring local genealogical trees from haplotypes with recombination. *Bioinformatics*, 33:1021–1030, 2017.

[16] D. A. Morrison. *Introduction to Phylogenetic Networks.* RJR Productions, Uppsala, Sweden, 2011.

[17] R. D. M. Page and M. A. Charleston. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. *Molecular Phylogenetics and Evolution*, 7:231–240, 1997.

[18] Nick Patterson, Priya Moorjani, Yontao Luo, Swapan Mallick, Nadin Rohland, Yiping Zhan, Teri Genschoreck, Teresa Webster, and David Reich. Ancient admixture in human history. *Genetics*, 192(3):1065–1093, 2012.

[19] J. Pei and Y. Wu. STELLS2: Fast and accurate coalescent-based maximum likelihood inference of species trees from gene tree topologies. *Bioinformatics*, 33:1789–1797, 2017.

[20] J.K. Pickrell and J.K. Pritchard. Inference of population splits and mixtures from genome-wide allele frequency data. *PLoS Genetics*, 8:e1002967, 2012.

[21] A. L. Price, N. Patterson, F. Yu, D. Cox, and et al. A genomewide admixture map for latino populations. *American Journal of Human Genetics*, 80, 2007.

[22] M.D. Rasmussen, M.J. Hubisz, I. Gronau, and A. Siepel. Genome-wide inference of ancestral recombination graphs. *PLoS Genet*, 10:e1004342, 2014.

[23] N. A. Rosenberg. The probability of topological concordance of gene trees and species trees. *Theor. Popul. Biol.*, 61:225–247, 2002.

[24] Leo Speidel, Marie Forest, Sinan Shi, and Simon Myers. A method for genome-wide genealogy estimation for thousands of samples. *bioRxiv*, 2019.

[25] G. A. Watterson. On the number of segregating sites in genetical models without recombination. *Theoretical Population Biology*, 7:256–276, 1975.

[26] D. Wen, Y. Yu, J. Zhu, and L. Nakhleh. Inferring phylogenetic networks using phylonet. *Syst Biol.*, 67:735–740, 2018.

[27] Daniel J. Wilson and Gilean McVean. Estimating diversifying selection and functional constraint in the presence of recombination. *Genetics*, 172:14111425, 2006.

[28] Y. Wu. New methods for inference of local tree topologies with recombinant snp sequences in populations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8:182–193, 2011.

[29] Y. Wu. Coalescent-based species tree inference from gene tree topologies under incomplete lineage sorting by maximum likelihood. *Evolution*, 66:763–775, 2012.

[30] Y. Wu. A coalescent-based method for population tree inference with haplotypes. *Bioinformatics*, 31:691–698, 2015.

[31] Y. Wu. An algorithm for computing the gene tree probability under the multispecies coalescent and its application in the inference of population tree. *Bioinformatics*, 32:i225–i233, 2016.

[32] Y. Yu, N. Ristic, and L. Nakhleh. The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection. *PLoS Genet*, 8(4):e1002660, 2012.