Ancestry Inference Using Reference Labeled Clusters of Haplotypes

Keith Noto, Yong Wang, Shiya Song, Joshua G. Schraiber, Alisa Sedghifar, Jake K. Byrnes, David A. Turissini, Eurie L. Hong, Catherine A. Ball

AncestryDNA, San Francisco, CA, 94107, USA

Abstract

We present ARCHes, a fast and accurate haplotype-based approach for inferring an individual's ancestry composition. Our approach works by modeling haplotype diversity from a large, admixed cohort of hundreds of thousands, then annotating those models with population information from reference panels of known ancestry. The running time of ARCHes does not depend on the size of a reference panel because training and testing are separate processes, and the inferred population-annotated haplotype models can be written to disk and used to label large test sets in parallel (in our experiments, it averages less than one minute to assign ancestry from 32 populations to 1,001 sections of a genotype using 10 CPU). We test ARCHes on public data from the 1,000 Genomes Project and HGDP as well as simulated examples of known admixture. Our results demonstrate that ARCHes outperforms RFMix at correctly assigning both global and local ancestry at regional levels regardless of the amount of population admixture.

Introduction

Admixture has played an important role in shaping patterns of genetic variation among humans and other species. It is of interest at both population and individual levels and has motivated a large body of research into population demography[1, 2] and population stratification[3] in association studies. It has also fueled public interest in direct to consumer (DTC) services that provide estimates of ancestry proportions. In such applications, a consumer typically submits a DNA sample through a saliva collection kit and receives an individual-level report of their ancestral make-up based on genotype data.

Over the past decade, many tools have been developed to infer individual-level ancestry. One set of methods only infers global ancestry proportions, some of which model the probability of the observed genotypes using ancestry proportions and population allele frequency,[4] while others use cluster analysis and principal component analysis (PCA).[5] Another set of methods infer ancestral origin for genomic segments, which are then averaged over the entire genome.

These methods use either SNPs (Single Nucleotide Polymorphisms) or a sequence of SNPs (*i.e.* haplotypes) as the observed variables, and estimate ancestry in each segment of the genome (called local ancestry). Compared to SNPs, haplotypes contain richer information, and can be especially powerful in differentiating geographically close populations.[6] Among existing haplotype-based methods, both Chromopainter[6] and HAPMIX[7] use the Li and Stephen's haplotype copying model,[8] whereas RFMix[9] uses a random forest approach, training classifiers on haplotype features in a reference panel and using a linear-chain conditional random field to model the conditional distribution of local ancestry given observed haplotypes.

As the size of public and private genotype datasets grows (*e.g.,* Ancestry has processed over 15 million human genomes), there is an increased need for methods that can efficiently and accurately perform ancestry inference on a large number of samples. Here we describe ARCHes (**A**ncestry inference using **R**eference labeled **C**lusters of **H**aplotyp**es**), a method that leverages reference panel labeled haplotype models to estimate diploid ancestry locally throughout the genome. ARCHes first uses a large set of unlabeled haplotypes to learn BEAGLE haplotype-cluster models,[10] which are efficient at phasing and measuring haplotype frequency. These BEAGLE models are then annotated with the probability that genotype sequences from a given reference population run through a particular state. For a given test individual, ARCHes calculates the probability that the observed genotype sequence comes from a given pair of populations, followed by a genome-wide hidden Markov model to assign diploid ancestry. These trained models need only be computed once, and can be stored thereafter, allowing ARCHes to efficiently estimate the ancestry of any number of subsequent test individuals from their genotype data.

Previous studies have shown that RFMix[9] outperforms ADMIXTURE[4] in both global and local ancestry estimation.[11] RFMix generally performs well at assigning ancestry at continental level but can struggle at regional level assignment, where populations may not be very differentiated. ARCHes is capable of differentiating nearby populations and performing ancestry inference at a much finer scale. We train both ARCHes and RFMix on research-consented individuals representing 32 different regions and test selected individuals from 1000 genomes[12] and HGDP,[13] representing 15 different regions. We compare the performance of ancestry assignments for individuals with single ancestry as well as simulated individuals with admixed ancestry in terms of both global ancestry proportions and diploid local ancestry assignments to those of RFMix.[9] Our results demonstrate that ARCHes outperforms RFMix in both global ancestry and diploid local ancestry assignments at regional levels.

Material and Methods

Overall ARCHes method

Our approach begins with dividing the genome into a large number of small windows (*e.g.,* 3-4 centimorgans each), such that, in a recently admixed individual, each of the maternal and paternal haplotypes in a given window are likely to each come from a single population. For each window, we construct a BEAGLE haplotype-cluster model[10] from a large, unlabeled training set of haplotypes. A BEAGLE haplotype-cluster model is a directed acyclic graph with haplotype represented as a path traversing the graph. Each node of the graph represents a cluster of haplotypes. A BEAGLE model is often interpreted as Markov model where the states are the nodes (Supplemental Figure 1), and thus as an “arbitrary order Markov model” of SNPs along a haplotype. Using a reference panel of genotypes from individuals whose ancestry is known in each window, we then annotate each state in the haplotype models with the probability that genotype sequences from a given population belong to the haplotype cluster represented by the state (Figure 1).

Given a new potentially admixed genotype sequence x, we assume that the ancestors of x are all ultimately from the K origin groups, and that x is admixed recently enough that relatively long haplotypes (on the scale of the genomic windows mentioned above) from each group are intact. We run a genome-wide hidden Markov model (HMM) whose hidden states are the true assignment (population label pairs) in each window. The emission probabilities are the probability distributions of diploid population assignments for each window arising from the annotated BEAGLE models and the transition probabilities (the probability that the population assignment will change at any point along the genome) are learned through an Expectation-Maximization (E-M) algorithm. We assign diploid ancestry to each window and estimate the global assignment based on the Viterbi path through this HMM. We also sample paths through the HMM to estimate the uncertainty of assignment amounts.

We describe our detailed method in the following sections, and provide pseudo-code in the Appendix.

Annotating haplotype cluster models

We follow Browning and Browning[10] in building haplotype cluster models. Briefly, we divide the genome into W partially overlapping windows with approximately the same number of SNPs. Within each window, we build a haplotype cluster model from a large, unlabeled set of training phased haplotypes. For simplicity, we restrict to biallelic variants, and code them as 0 and 1. Building this haplotype cluster model from a large, unlabeled set of individuals provides a “background” of haplotype diversity against which we can measure the informativeness of different haplotypes.

With a haplotype cluster model built for each window, we can then annotate populations using the haplotype cluster model. Recall that each path through a BEAGLE model corresponds to a realization of a haplotype, and each node at a given SNP represents a cluster of haplotypes that are similar near that SNP. For the genotypes of a reference individual in window w, x_w, we compute the probability that the individual's two haplotypes pass through two specific nodes in the graph, u and v, at SNP d,

$$P_d(u, v|\mathbf{x}_w) = \frac{P_d(\mathbf{x}_w, u, v)}{P(\mathbf{x}_w)}$$

where we compute $P_d(u, v|\mathbf{x}_w)$ and $P(\mathbf{x}_w)$ using a modification of the forward-backward algorithm for hidden Markov models, treating the node as a hidden state (see Appendix for pseudo-code). In the following, we will refer to the HMM used to analyze the BEAGLE models as the *haplotype HMM*, and its properties as *haplotype emission probabilities,* and *haplotype probabilities*. This contrasts with the *ancestry HMM* we use to smooth ancestry estimates across the genome, which is described in the subsequent section.

We then marginalize over one of the haplotypes of each diploid to create a haplotype posterior probability that the genotypes $\mathbf{x}_w$ in window w passes through node u at SNP d,

$$P_d(u|\mathbf{x}_w) = \sum_v P_d(u, v|\mathbf{x}_w)$$

Finally, we annotate a node u by its average haplotype probability in a set of individuals belonging to a reference population p, $R_p = \{\mathbf{x}_{i,p,w}, i \in 1, 2, \ldots, n_p\}$ where n_p is the total number of reference samples in population p. Then, we compute

$$P_d(u|p) = \frac{1}{n_p} \sum_{i=1}^{n_p} P_d(u|\mathbf{x}_{i,p,w}) \tag{1}$$

This equation gives us the probability that an individual drawn from population p will pass through node u at SNP d of the haplotype cluster model for window w.

Ancestry emission probabilities for test individuals in windows

With Equation (1) in hand, we can compute the probability that a test individual's genotypes in a given window w descend from a specific pair of populations. Letting $\mathbf{t}$ be the unphased genotype of our test individual, we first compute the probability of $\mathbf{t}$ given that the two haplotypes in window w belong to clusters u and v of the haplotype cluster model at SNP d,

$$P_d(\mathbf{t}_w|u, v) = \frac{P_d(\mathbf{t}_w, u, v)}{P_d(u, v)},$$

where $P_d(\mathbf{t}_w, u, v)$ is computed using the haplotype forward-backward algorithm and $P_d(u, v)$ is obtained by multiplying the transition matrices of the haplotype cluster model up to SNP d (equivalent to running the haplotype forward algorithm up to SNP d with all haplotype emission probabilities set equal to 1).

We then want to know the probability that the individual's two haplotypes come from populations p and q using the information around SNP d. We compute this quantity by first computing the probability that a haplotype passes through nodes u and v and SNP d of window w given underlying populations p and q by averaging over the equally likely combinations of whether node u corresponds to population p and node v corresponds to population q or vice versa,

$$P_d(u, v|p, q) = \frac{1}{2}(P_d(u|p)P_d(v|q) + P_d(u|q)P_d(v|p)).$$

Note that this result is equivalent to assuming that the two haplotype clusters that make up a diploid sample are independent, and that the two populations that make up those haplotypes are also independent.

Now, we use the law of total probability to average over all haplotype clusters at SNP d, and compute the probability that the individual's haplotype clusters at that point arise from populations p and q,

$$P_d(\mathbf{t}_w|p, q) = \sum_{u,v} P_d(\mathbf{t}_w|u, v)P_d(u, v|p, q).$$

This probability weighs similarity to haplotypes in population p and q more strongly for SNPs closest to SNP d in window w; because we have no *a priori* knowledge of which part of a window is most informative about population membership, we finally compute our ancestry emission probability for a window by averaging over the population probability for every SNP in the window,

$$P(\mathbf{t}_w|p, q) = \frac{1}{D}\sum_{d} P_d(\mathbf{t}_w|p, q) \qquad (2)$$

where D is the total number of SNPs in window w. This process can then be repeated for every window in the genome to obtain the probability of the test individual's genotype in each window, given that the two haplotypes arose from any pair of populations p and q.

Smoothing ancestry estimates using a genome-wide ancestry hidden Markov model

In principle, the ancestry emission probabilities computed in the previous section could be used to compute maximum likelihood estimates of diploid local ancestry in each window, one at a time. However, doing so would result in highly noisy ancestry estimates. Instead, we share information across the genome using an additional layer of smoothing via a genome-wide hidden Markov model. Moreover, because ancestry segments from recent admixture are expected to be longer than a single window, this model helps reduce false ancestry transitions.

If we wish to assign ancestry to K populations, the hidden states of our hidden Markov model are the $\binom{K}{2} + K$ possible unphased ancestry pairs, (p, q), with ancestry emission probabilities window w given by equation (2). Because we model *unphased* diploid ancestry, we define a population pair as unordered, *i.e.* (p, q) is the same ancestry assignment as (q, p). Our ancestry hidden Markov model assumes that between windows ancestry can change for *one* of the two haplotypes with probability τ. The assumption that ancestry switches only for one of the two haplotypes within an individual is both biologically realistic (assuming individuals are admixed relatively recently) and greatly reduces the complexity of the hidden Markov model. Thus, a change occurs from (p, q) to (p', q') to any pair such that exactly one of p' or q' is different from p or q. Each new ancestry pair is drawn with probability proportional to the stationary probability of that ancestry pair, $\pi_{p,q}$. In full, the transition probabilities are

$$P(p', q'|p, q) = \begin{cases} 1 - \tau & \text{if } p' = p, q' = q, \\ \tau \frac{\pi_{p',q'}}{Z_{p,q}} & \text{if } p' \neq p, q' = q \text{ or } p = p, q' \neq q \\ 0 & \text{else} \end{cases} \tag{3}$$

where the normalizing constant $Z_{p,q}$ is given by summing over all accessible unphased haplotype pairs.

Between chromosomes, both ancestry pairs are allowed to change, and the ancestry at the start of each chromosome is drawn independently from that individual's global distribution of ancestry pairs, $\pi_{p,q}$. For a more formal description of how changes between chromosomes are handled, see the Appendix.

We initialize the $\pi_{p,q}$ to a uniform distribution and τ to some low value, and use a modified Baum-Welch algorithm to update $\pi_{p,q}$ and τ (see Appendix). Empirically, we observed a tendency to overfit by estimating a large τ parameter, resulting in inference of a large number of different ancestries; thus we run a fixed number of update steps, rather than stopping at convergence.

Estimating ancestry proportions in individuals

In principle, the value $\pi_p = \sum_{q'} \pi_{p,q}$ could be used as an estimate of the admixture proportion from population p in an individual. However, we instead opt to use a path-based approach that also allows us to obtain credible intervals of the ancestry proportions conditioned on the inferred parameters. Specifically, we provide a point estimate of global ancestry proportions by computing the maximum probability path through the HMM using the Viterbi algorithm, and computing the proportion of windows (weighted by their length) that are assigned to population p. We then provide a credible interval by then sampling paths from the posterior distribution on paths, and for each one can compute the ancestry proportion in the same way as from the Viterbi path. Because these credible intervals condition on the parameters, particular the $\pi_{p,q}$, they tend to be conservative with respect to ancestry proportions, reflecting mostly genotype sampling randomness. Thus, we advocate caution in interpreting them too literally.

Below we describe experiments we did for benchmarking ARCHes and RFMix[9].

Reference Panel and Testing Data

We built our reference panel using genotypes from customer candidates who explicitly provided prior consent to participate in research and have all family lineages tracing back to the same geographic region. All the candidates were genotyped on Ancestry's SNP array and were analyzed through a quality control pipeline to remove samples with low genotype call rates, samples genetically related to each other, and samples who appear as outliers from their purported population of origin based on Principal Component Analysis. The reference panel contains 11,051 samples, representing ancestry from 32 global regions (Supplemental Table 1). We then use 1,705 individuals from 1,000 Genomes[12] and HGDP Project[13] from 15 populations as testing data. We used SNP array data of individuals from 1,000 Genomes[12] and HGDP Project[13] and limited them to around 300,000 SNPs that overlapped with Ancestry's SNP array. Lists of populations and associated sample counts included in reference panel and testing data are specified in Supplemental Table 1 and 2, respectively. We align populations that come from different data sources, in some cases combining populations together. For example, we combined the ancestries that are assigned to 'England, Wales, and Northwestern Europe' and 'Ireland & Scotland' to represent ancestry for 'Britain'. We combined the ancestry that are assigned to 'Benin & Togo' and 'Nigeria' to represent ancestry for 'Yoruba'.

Simulation

We simulated genomes of admixed individuals with ancestors from a pair of populations and we performed the simulation for 16 pairs of neighboring populations. We first constructed a pedigree with 32 founders, with a single founder from one population, and the rest from the other population. We then simulated the recombination process and obtained the haplotypes for each descendant for 4 generations. We then select descendants from the pedigrees that are roughly 50%-50% admixed, 25%-75% admixed, 12.5%-87.5% admixed, and 6.25%-93.75%

admixed. We simulated 20 individuals for each of the 16 different pairings and 4 different levels of admixture.

We also simulated 100 individuals with an admixture history similar to modern Latinos that admixed 12 generations ago with 45% Native American, 50% European and 5% African ancestry. We constructed 100 12-generation pedigrees and randomly selected founders from the reference panel, with the ratio of 45% Native American (from the Maya and Peru regions), 50% European (from the France, Britain, Italy, Spain and Finland regions), and 5% African ancestry (from the Yoruba region). We then simulated the recombination process as above and obtained the genotypes of the descendant in each pedigree, which are roughly 45% Native American, 50% European and 5% African.

Since RFMix needs the phased haplotypes for both query and reference individuals, we used Eagle[15] v2 with the HRC[17] reference panel to get the phased haplotypes of the simulated individuals as well as for the individuals in the reference panel. However, ARCHes requires only the unphased, diploid genomic sequences for both query and reference individuals.

RFMix parameters

We first used default parameters in RFMIX v2.03-r0 (https://github.com/slowkoni/rfmix). We then performed a parameter sweep using different number of generations since admixture(the -G parameter), with value of 2, 4, 6 and 8 coupled with different window sizes (set both CRF window size and random forest window size) with values of 0.2cM, 0.5cM, 100 SNPs (roughly 1cM) and 300 SNPs (roughly 3cM) on chromosome 1 of simulated pair admixed individuals. We then selected the parameters with the best performance, namely 4 generations since admixture and a window size 0.2cM, and ran RFMix on the whole genome of simulated pair admixed individuals. For simulated latino individuals, we used 12 generations since admixture and a window size 0.2 cM. For single origin individuals, we used 2 generations since admixture and a window size 0.2 cM. None of the RFMix runs used the E-M procedure or phase error correction.

ARCHes parameters

We divide the genome into 3,882 windows of 80 SNPs each, overlapping by 5 SNPs (with some adjustments made near chromosome boundaries). We build a haplotype model for each of these windows from the phased haplotypes of 50,000 individuals that are not in the reference panel, but we tie small groups of 3-4 windows together by disallowing population assignment transitions within those groups, which allows us to set the granularity with which we assign local population assignments (there are 1,001 such window groups) and has the benefit of increased computational efficiency. ARCHes's haplotype model annotation process is robust to missing data, which is handled by marginalizing over all possible genotypes. In fact, the annotations may benefit from intentionally downsampling reference panel genotypes so that haplotypes are considered that are similar to but not exactly the same as those in the reference panel, and the amount of downsampling and the number of downsampled genotypes used for annotation are tunable parameters of the annotation process. In our experiments, we sample each reference panel genotype sequence 100 times, each time setting 20% of genotypes to missing and annotating the 3,882 haplotype models with them.

We set the initial $\tau_{\mathbf{x}}$ parameter to be 0.01 and learned this parameter using 10 iterations of the E-M approach described above. ARCHes assigns diploid local ancestry to 1,001 windows of the genome and the global ancestry estimates are summarized from these 1,001 windows.

Results

Separate Training and Test Phases to Facilitate High-Throughput Ancestry Estimation

The ARCHes software represents a change in design that explicitly separates two phases, first model creation and annotation and second ancestry estimation, in order to make ancestry estimation both efficient and distributable. The first phase, learning the haplotype models from a large unlabeled training set and then annotating them with the reference panel populations, need only be carried out once. In order to estimate ancestry on subsequent instances, ARCHes software need only reload models and can be run on new examples at any time, distributed as necessary, and the running time depends only on the number of the number of individuals to be processed and labeled, not the size of the reference panels. In contrast, the training and testing processes of RFMix are not separate and require significantly more time per individual. We compare ARCHes's runtime and memory usage with RFMix in Supplemental Table 3.

Accuracy for single origin individuals

We built our reference panel using genotypes from research consented individuals, representing 32 regions. We then applied ARCHes on individuals from 1,000 genomes and HGDP representing 15 regions. Lists of populations and associated sample sizes for both training and testing data are in Supplemental Table 1 and 2. We first looked at the accuracy for single-origin individuals, namely the average estimated ancestry proportions for individuals from a given region. ARCHes predicted on average 66.1% of the ancestry to be from the correct region (Figure 2). The rest of the ancestry mainly came from nearby regions (Supplemental Figure 3). ARCHes performed well at separating different countries within Africa, and within Europe, and within Asia, with only a few exceptions. In comparison, RFMix predicted on average 43.5% of the ancestry to be from the correct region, and the rest of the ancestry mainly came from neighboring regions, showing that RFMix is accurate for continental level assignments but performs less well at finer scales.

Accuracy for simulated admixed individuals

Next, we simulated genotypes for individuals with ancestry from 16 different pairings of 11 regions and ran ARCHes using the same reference panel from research consented individuals, representing 32 regions, as we used for analyzing single origin individuals. We measured the precision and recall for each of the 11 regions (Supplemental Figure 4). Precision was calculated as the amount of correctly identified ancestry divided by the estimated value for that region and recall was calculated as the amount of correctly identified ancestry divided by the true value for that region. Precision can be thought of as how much of the reported ancestry is true, and recall can be thought of as how much of the true ancestry is called by the process. We find that ARCHes generally outperforms RFMix in terms of both precision and recall.

We calculated the concordance in terms of global ancestry assignments, namely the sum of overlap between the true and estimated proportions for each region. We also calculated the accuracy of diploid local ancestry assignments, namely the proportion of genomic windows with correct diploid assignments regardless of the phase. Overall, ARCHes achieves more than 50% global ancestry concordance and diploid local ancestry concordance, especially for simulated individuals who are from two nearby European or Asian countries (Figure 3). We don't find a large difference between global ancestry concordance and diploid local ancestry concordance, indicating that ARCHes achieves its global ancestry accuracy by estimating local ancestry accurately. It is also encouraging that ARCHes is capable of differentiating populations not only on a continental level but also on sub-continental and even country levels. RFMix in general performs worse than ARCHes, with a roughly 20% -30% deduction of concordance in terms of both global and local ancestry concordance.

Accuracy for simulated Latino individuals

We finally simulated 100 individuals using forward simulation with a pedigree mimicking Latino population history in which founders admixed 12 generations ago with 45% Native American, 50% European and 5% African ancestry. Their American ancestry came from Maya and Peru, their European ancestry came from France, Britain, Italy, Spain and Finland, and their African ancestry came from Yoruba (refer to Supplemental Table 1 and 2 for population labels). This dataset provides information on ARCHes's power to differentiate continental level admixture that happened as many as 12 generations ago. Moreover, we can see if it can even differentiate the subregions that individuals' continental ancestry comes from. We found that ARCHes accurately recovered both global ancestry assignments and diploid local ancestry assignments, with average concordances of 72.3% and 47.8%, respectively (Supplemental Figure 5). RFMix achieved 65.7% global ancestry concordance but failed to infer the local assignments correctly, with average diploid local ancestry concordance of 18.5%. This is probably due to difficulties that RFMix has in differentiatiating subregions within Europe and between Maya and Peru.

Discussion

Ancestry inference in large, heterogeneous sample sets is becoming increasingly important for academics, clinicians, and consumers. We showed that ARCHes is able to be trained on a wide set of global populations and perform accurately at within and among continental scales, without any specific fine tuning. Moreover, because our approach separates the time-consuming training step from the fast testing step, it can be applied to large scale databases, such as Ancestry's 15 million customers.

Our approach works because haplotypes contain rich information for distinguishing subpopulations. Instead of coding haplotype sequences as consecutive alleles, we take advantage of the haplotype-cluster models that have been shown to be effective at phasing[10]. Compared to RFMix[9]'s approach of training random forests on haplotypes from reference panels, we annotated haplotype clusters with the probability of each population label in our reference panel. To improve robustness, ARCHes can account for incomplete haplotype representation in the reference panel via tuning of a parameter that controls the proportion of missing genotypes during model annotation. It is important to note that our reference panel does not need to consist exclusively of whole-genome single-origin training examples. Because we annotate haplotype models in individual windows across the genome, we are able to utilize population-labeled partial-genome diploid or haploid genotype examples as well. That means that the accuracy of ARCHes can be improved even if a reference genotype is admixed, or if only part of it has known ancestry.

Utilizing rich haplotype models in each window, we assign a likelihood over all population labels to the haplotypes in our test sample, which are used as emission probabilities in the genome-wide HMM. HMMs are used in a number of existing approaches for estimating ancestral proportions.[6,7] We applied standard HMM techniques and learn parameters through iterations of a Baum-Welch[16] approach. In cases where sufficient prior knowledge is in hand, parameter learning can also be turned off and ARCHes can use predefined parameters. For instance, when analyzing descendents of a specific, known admixture event, it may be desirable to fix the prior distribution on global ancestry proportions. Nonetheless, in our benchmark experiments, we use one set of parameters for testing single origin individuals and simulated admixed individuals who were admixed from a range of generations ago. Without fine-tuning the parameters to each situation, we can achieve high accuracy. However, to achieve the highest accuracy, we suggest performing a parameter sweep to optimize ARCHes's performance for a particular dataset. In particular, our results show that ARCHes often overfits the data, and estimates too many different ancestral backgrounds in an individual. This suggests that for applications where precision is important, a user may want to constrain the switch rate parameter τ to be small.

The size and composition of the reference panel may have an impact on the accuracy of the ancestry estimation. As one might expect, a larger reference panel will improve performance, as a greater proportion of haplotype diversity is represented. We found that even with small reference panel sizes (for example, Maya, France and Slavic have less than 50 samples each), ARCHes can achieve very high accuracy for single origin individuals (Supplemental Figure 6). On the other hand, an imbalance in reference panel size/diversity between populations can result in mis-assignment to the larger reference panel. We can compensate for this effect by tuning a parameter that controls the fraction of missing genotypes set to missing during the annotation process. However, the impact of doing so may be limited due to the intrinsic properties of haplotype diversity and the sharing of haplotypes between labeled populations.

For other local ancestry approaches, such as RFMix, phasing error will result in decreased accuracy of ancestry estimation. However, because ARCHes uses unphased genotype data, it is unaffected by phasing errors, thus removing an entire source of error from the analysis. Moreover, ARCHes can account for missing data by integrating for all possible paths on the haplotype-cluster model, though it may be preferable to use imputed genotypes.

ARCHes provides a fast and accurate method for inferring unphased local ancestry and combining that into estimates of diploid global ancestry. There are nonetheless several opportunities for future research. First of all, the confidence intervals provided by ARCHes are underestimated; it is possible that they can be rescued by using a recalibration procedure on simulated data. Second, despite the fact that using unphased local ancestry in ARCHes helps it to overcome phasing errors, it may be desirable to provide phased local ancestry in some circumstances. Because of the modular nature of the ancestry hidden Markov model, it may be possible to extend this framework to provide phased local ancestry estimates.

Description of Appendices

Appendix includes 3 algorithm boxes and another 3 algorithm descriptions.

Description of Supplemental Data

Supplemental Data includes six figures and three tables.

Declaration of Interests

The authors declare competing financial interests: authors affiliated with AncestryDNA may have equity in Ancestry. The work described in this manuscript is covered by one or more patents including US patent entitled Local Genetic Ethnicity Determination System US10558930B2.

Acknowledgements

We appreciate Carlos Bustamante and Mark Koni Wright for providing RFMix software and guidance on using it.

References

1. Loh, P.-R., Lipson, M., Patterson, N., Moorjani, P., Pickrell, J.K., Reich, D., and Berger, B. (2013). Inferring admixture histories of human populations using linkage disequilibrium. Genetics *193*, 1233–1254.

2. Gravel, S. (2012). Population Genetics Models of Local Ancestry. Genetics *191*, 607–619.

3. Marchini, J., Cardon, L.R., Phillips, M.S., and Donnelly, P. (2004). The effects of human population structure on large genetic association studies. Nature Genetics *36*, 512–517.

4. Alexander, D.H., Novembre, J., and Lange, K. (2009). Fast model-based estimation of ancestry in unrelated individuals. Genome Res. *19*, 1655–1664.

5. Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. Nature Genetics *38*, 904–909.

6. Lawson, D.J., Hellenthal, G., Myers, S., and Falush, D. (2012). Inference of population structure using dense haplotype data. PLoS Genet. *8*, e1002453.

7. Price, A.L., Tandon, A., Patterson, N., Barnes, K.C., Rafaels, N., Ruczinski, I., Beaty, T.H., Mathias, R., Reich, D., and Myers, S. (2009). Sensitive detection of chromosomal segments of distinct ancestry in admixed populations. PLoS Genet. *5*, e1000519.

8. Li, N., and Stephens, M. (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. Genetics *165*, 2213–2233.

9. Maples, B.K., Gravel, S., Kenny, E.E., and Bustamante, C.D. (2013). RFMix: a discriminative modeling approach for rapid and robust local-ancestry inference. Am. J. Hum. Genet. *93*, 278–288.

10. Browning, S.R., and Browning, B.L. (2007). Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. Am. J. Hum. Genet. *81*, 1084–1097.

11. Uren, C., Hoal, E.G., and Möller, M. Putting RFMix and ADMIXTURE to the test in a complex admixed population.

12. 1000 Genomes Project Consortium, Abecasis, G.R., Altshuler, D., Auton, A., Brooks, L.D., Durbin, R.M., Gibbs, R.A., Hurles, M.E., and McVean, G.A. (2010). A map of human genome variation from population-scale sequencing. Nature *467*, 1061–1073.

13. Li, J.Z., Absher, D.M., Tang, H., Southwick, A.M., Casto, A.M., Ramachandran, S., Cann, H.M., Barsh, G.S., Feldman, M., Cavalli-Sforza, L.L., et al. (2008). Worldwide human relationships inferred from genome-wide patterns of variation. Science *319*, 1100–1104.

14. Williams, A.L., Patterson, N., Glessner, J., Hakonarson, H., and Reich, D. (2012). Phasing of

many thousands of genotyped samples. Am. J. Hum. Genet. *91*, 238–251.

15. Loh, P.-R., Danecek, P., Palamara, P.F., Fuchsberger, C., A Reshef, Y., K Finucane, H., Schoenherr, S., Forer, L., McCarthy, S., Abecasis, G.R., et al. (2016). Reference-based phasing using the Haplotype Reference Consortium panel. Nat. Genet. *48*, 1443–1448.

16. Rabiner, L.R. (1990). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Readings in Speech Recognition 267–296.

17. McCarthy, S., Das, S., Kretzschmar, W., Delaneau, O., Wood, A.R., Teumer, A., Kang, H.M., Fuchsberger, C., Danecek, P., Sharp, K., et al. (2016). A reference panel of 64,976 haplotypes for genotype imputation. Nat. Genet. *48*, 1279–1283.

Figures

Figure 1. Illustration of annotating haplotype-cluster model. Each box illustrates the expected proportion of haplotypes in all the genotypes of different populations that include a certain model state at a certain level.

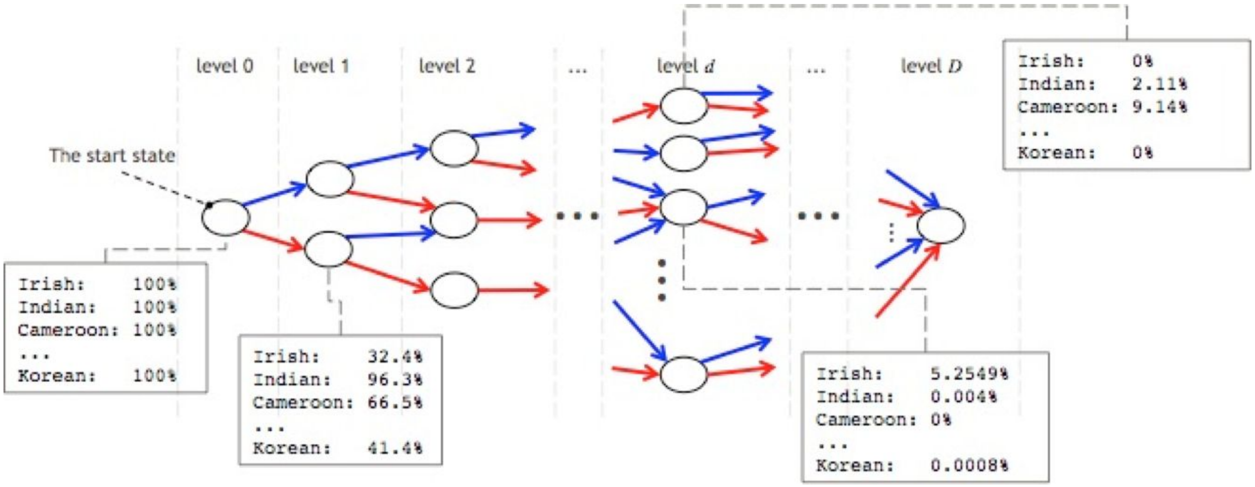


Figure 2. Boxplot of the estimated ancestry proportions for single-origin individuals from each testing population comparing ARCHes and RFMix.

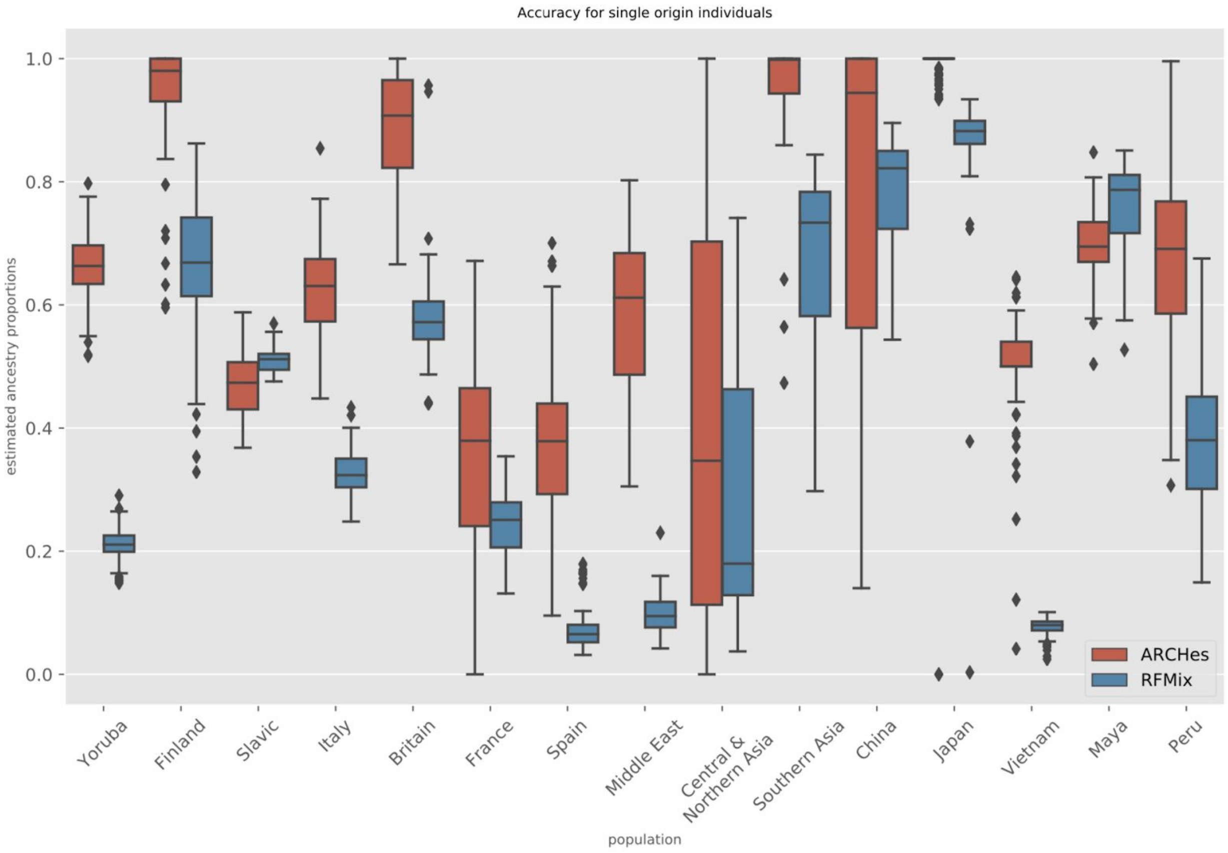
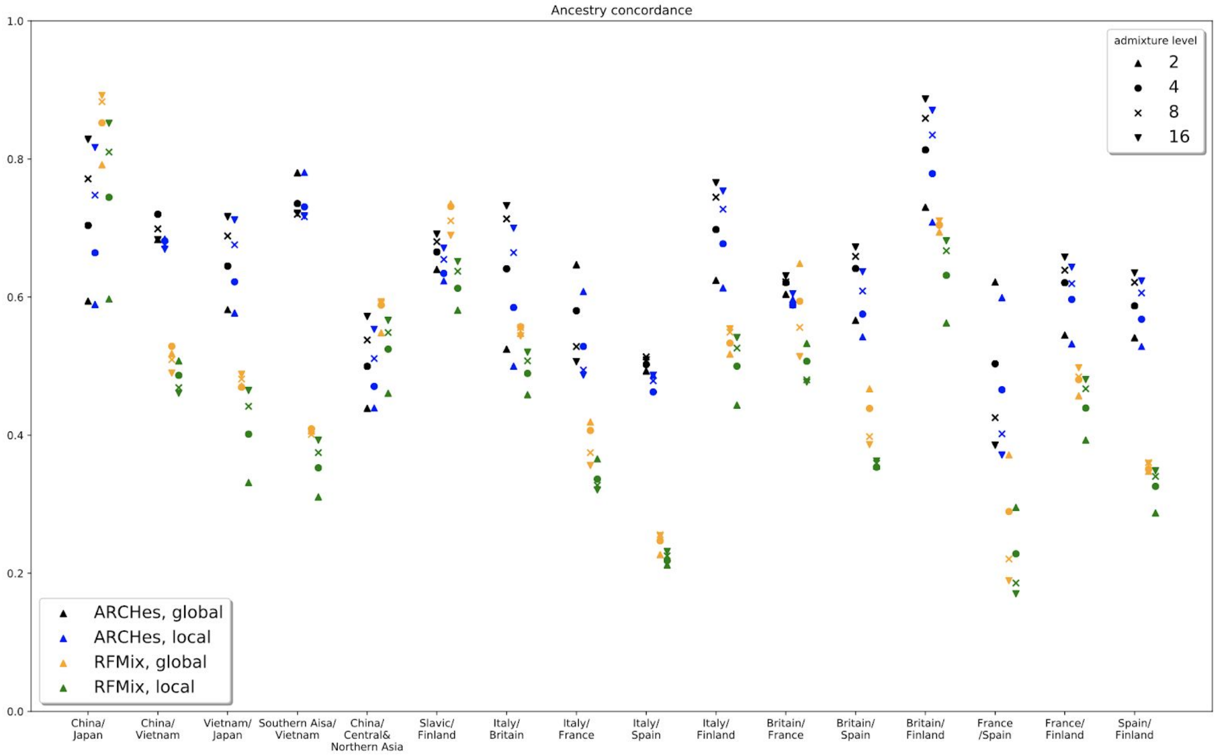
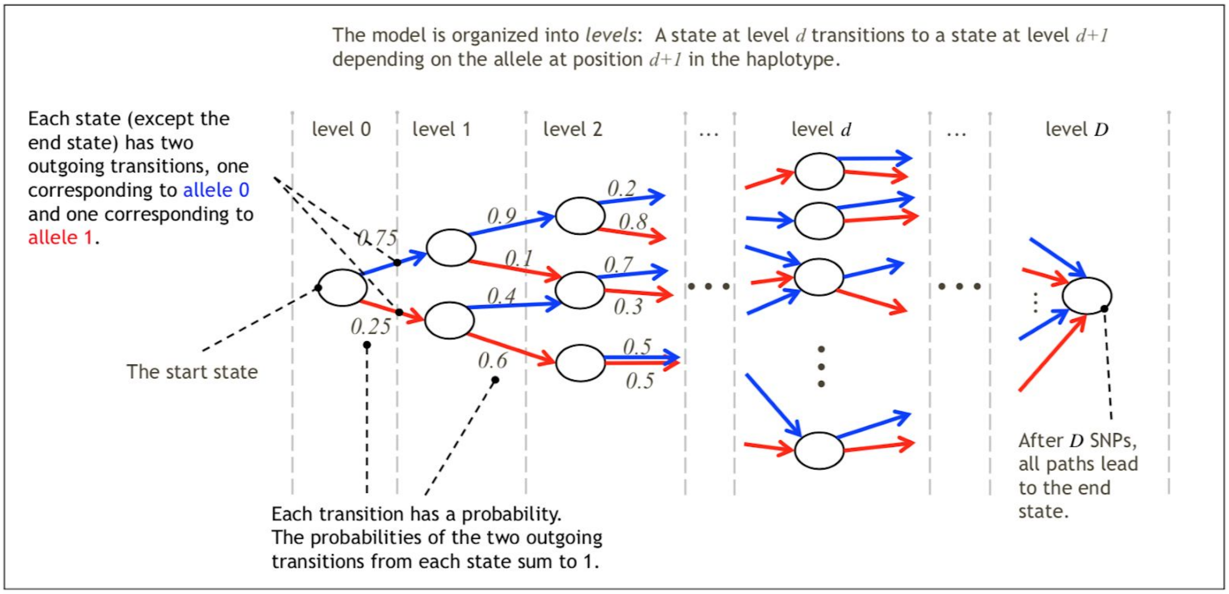


Figure 3. Concordance of global ancestry assignments and diploid local ancestry assignments for simulated admixed individuals from 16 different pairings of 11 populations. Admixture level is defined as x-way admixed with x founders, 1 of which belong to one population, the rest belong to another population. 2-way admixed results in 50%-50%, 4 way admixed results in roughly 25%-75%, 8 way admixed results in roughly 12.5%-87.5%, 16 way admixed results in roughly 6.25%-93.75%.

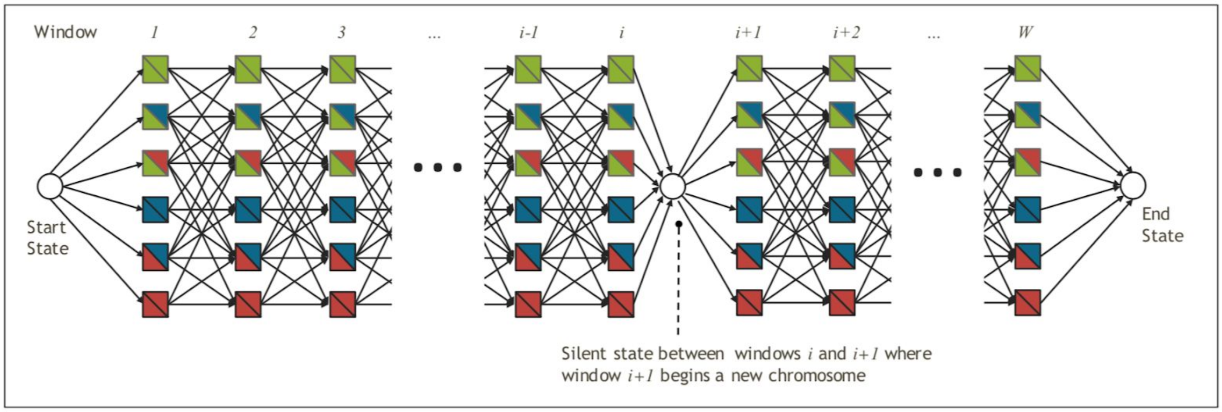


Supplemental Data

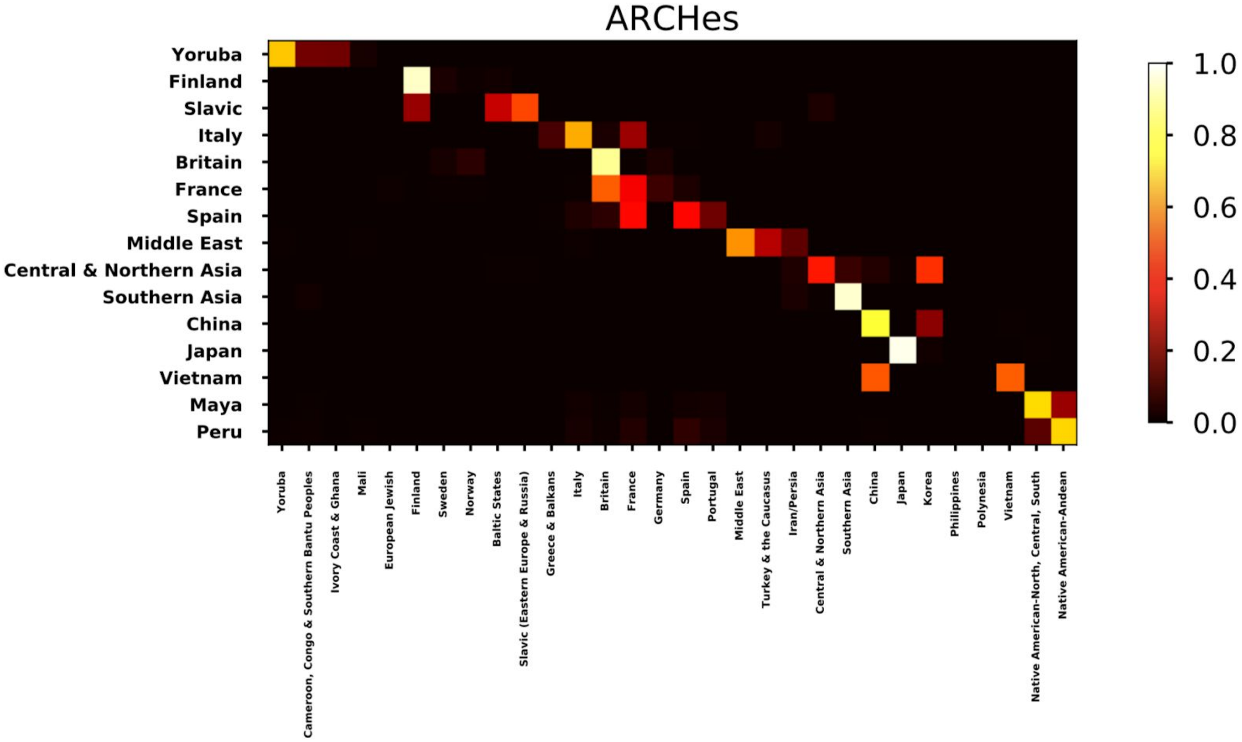
Supplemental Figure 1. Illustration of haplotype model for one window of the genome, consisting of *D* SNPs.

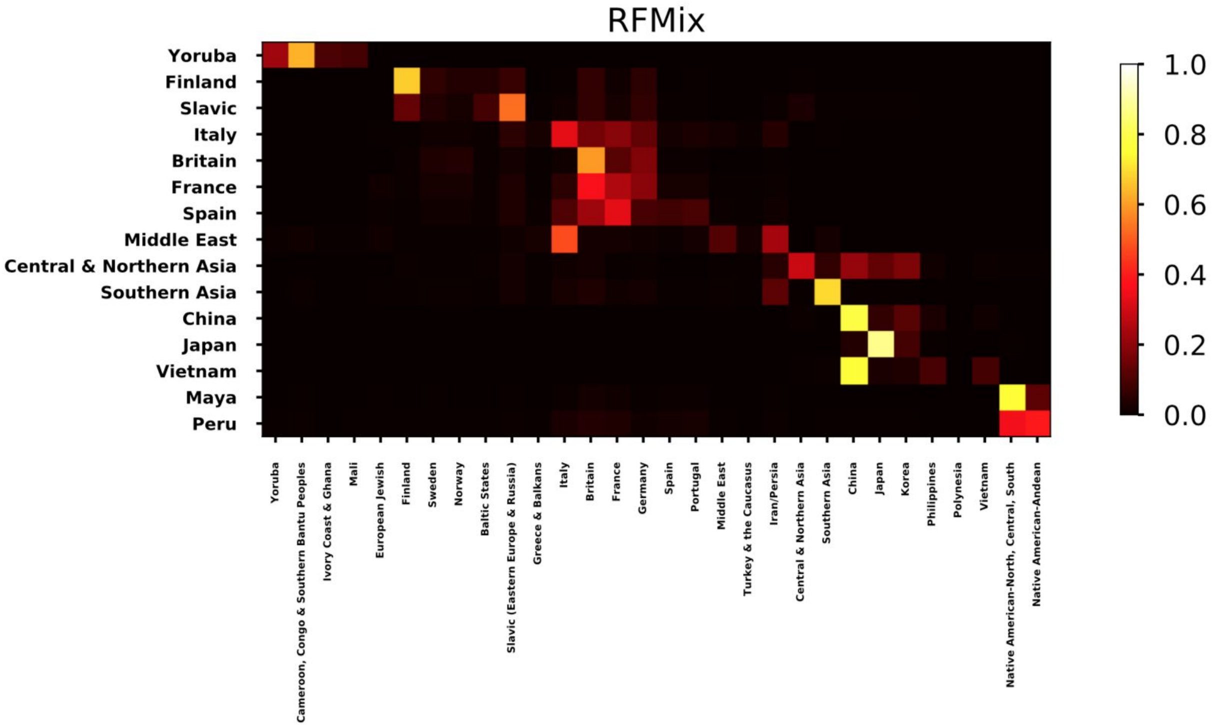

The model is organized into *levels*: A state at level *d* transitions to a state at level *d+1* depending on the allele at position *d+1* in the haplotype.
Each state (except the end state) has two outgoing transitions, one corresponding to allele 0 and one corresponding to allele 1.
level 0
level 1
level 2
...
level *d*
...
level *D*
0.75
0.25
0.9
0.1
0.4
0.6
0.2
0.8
0.7
0.3
0.5
0.5
The start state
Each transition has a probability. The probabilities of the two outgoing transitions from each state sum to 1.
After *D* SNPs, all paths lead to the end state.

Supplemental Figure 2. Illustration of genome wide HMM where each window has a series of emitting states, which corresponds to a population assignment (p, q) with $1 \leq p \leq q \leq K$

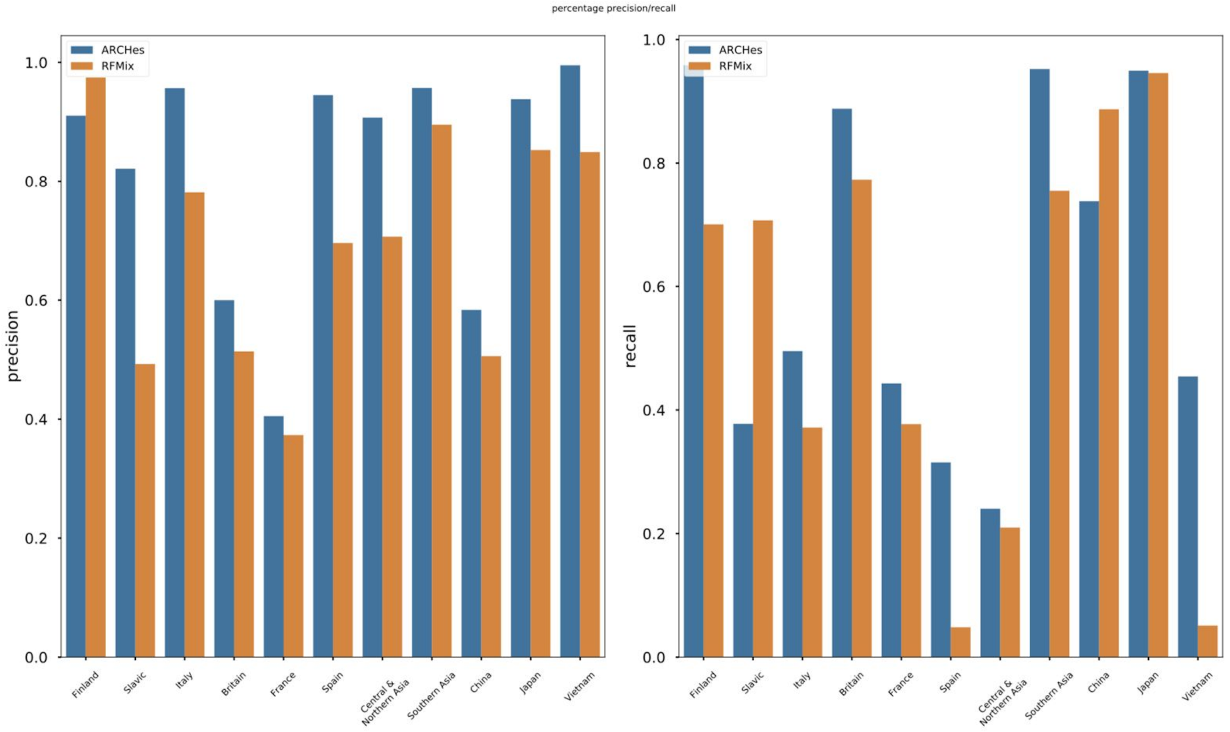

Window
1
2
3
...
i-1
i
i+1
i+2
...
W
Start State
End State
Silent state between windows *i* and *i+1* where window *i+1* begins a new chromosome

Supplemental Figure 3. Average estimated ancestry proportions for single-origin individuals from each testing population. In this matrix figure, each row represents single-origin individuals from the testing population. Each column represents each of the possible 30 populations that the single-origin individuals might be assigned to.

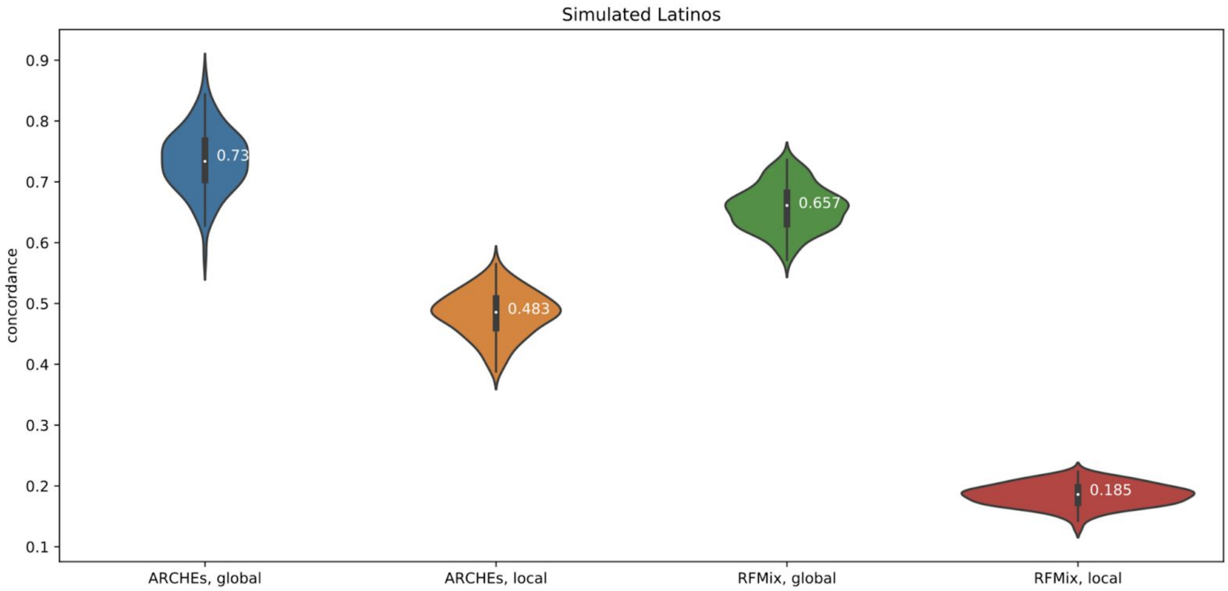




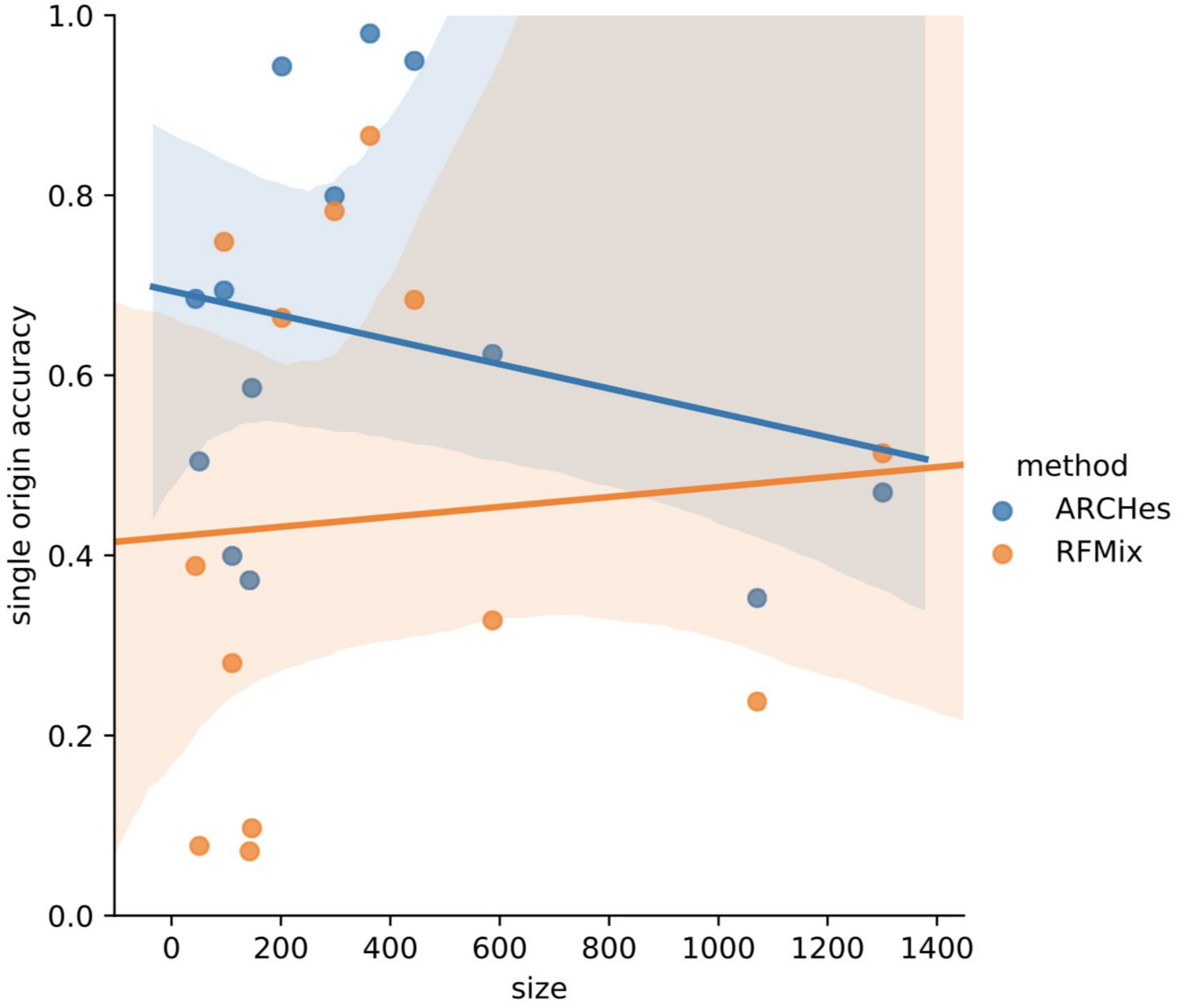
Supplemental Figure 4. Precision/Recall for each population calculated from estimated ancestry proportions of simulated admixed individuals with ancestry from a pair of neighboring population.



Supplemental Figure 5. Concordance of global ancestry assignments and diploid local ancestry assignments on 100 simulated latino individuals.



Supplemental Figure 6. Relationship between the number of individuals in the reference panel and the accuracy for single origin individuals for each population.



Supplemental Table 1. Sample size and geographic location for 32 populations in the reference panel. Some population is matched with testing population specified in Supplemental Table 2.

Population Label	Sample size	Matched testing population
Native American-North, Central, South	96	Maya
Native American-Andean	44	Peru
England, Wales, and Northwestern Europe	1226	Britain
Central & Northern Asia	111	Central & Northern Asia

Southern Asia	444	Southern Asia
Baltic States	127	
Benin & Togo	102	Yoruba
Cameroon, Congo & Southern Bantu Peoples	576	
Ireland & Scotland	319	Britain
China	298	China
European Jewish	129	
France	1071	France
Germany	1314	
Greece & Balkans	149	
Italy	587	Italy
Ivory Coast & Ghana	119	
Japan	363	Japan
Korea	201	
Mali	169	
Middle East	147	Middle East
Nigeria	109	Yoruba
Norway	242	
Iran/Persia	413	
Philippines	385	
Polynesia	57	
Portugal	257	
Slavic (Eastern Europe & Russia)	1301	Slavic
Spain	143	Spain
Sweden	240	

Turkey & the Caucasus	59	
Finland	202	Finland
Vietnam	51	Vietnam

Supplemental Table 2. Sample size and geographic label for testing population from HGDP and 1000 Genomes.

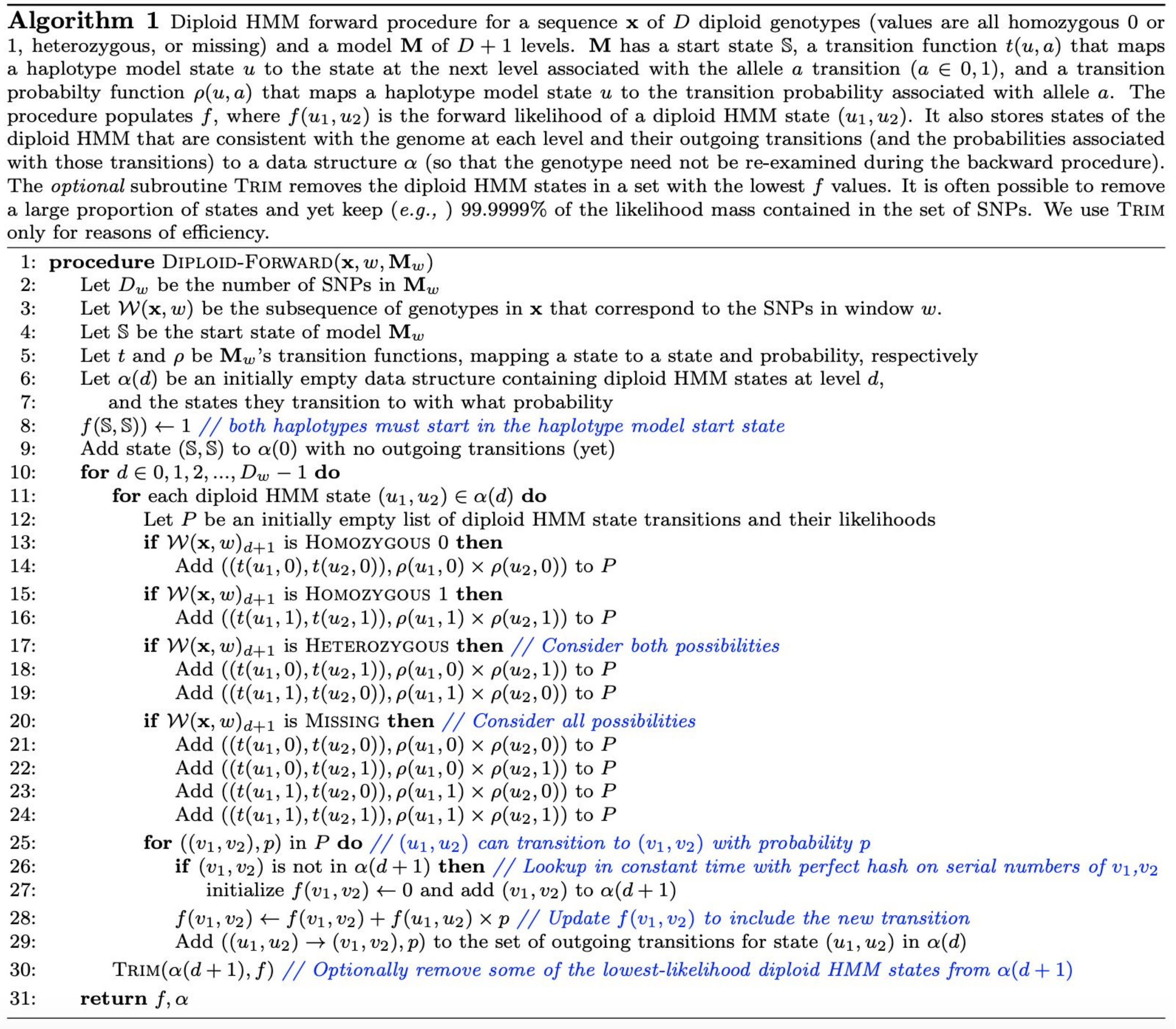
Population label	Detailed label	Sample size	Source
Maya	Maya	25	HGDP
Peru	PEL(Peruvians from Lima, Peru)	105	1000 Genomes
Central & Northern Asia	Daur, Hazara, Hezhen, Mongola, Oroqen, Tu, Uygur, Xibo, Yakut	116	HGDP
Southern Asia	Pathan, Sindhi	48	HGDP
Yoruba	YRI (Yoruba in Ibadan, Nigeria), Yoruba	213	1000 Genomes, HGDP
China	CHS (Southern Han Chinese), Han, She, Tujia	325	1000 Genomes, HGDP
France	French	29	HGDP
Britain	GBR (British in England and Scotland)	104	1000 Genomes
Italy	TSI (Toscani in Italia)	112	1000 Genomes
Japan	JPT (Japanese in Tokyo, Japan), Japanase	134	1000 Genomes, HGDP
Middle East	Druze, Palestinian	98	HGDP
Slavic	Russian	25	HGDP

Spain	IBS (Iberian Population in Spain)	150	1000 Genomes
Finland	FIN (Finnish in Finland)	100	1000 Genomes
Vietnam	KHV (Kinh in Ho Chi Minh City, Vietnam)	121	1000 Genomes

Table 3. Run time and Memory Usage (Maximum resident set size, MaxRSS) comparison between ARCHes and RFMix. Since ARCHes trains models in a separate process, we only count the running time and MaxRSS for inferring ancestry for test individuals. However, because RFMix combines the training and testing process together, we count the running time and MaxRSS for both training and testing process for RFMix.

Experiment	# of test individuals	Method	User time (s)	MaxRSS
Single origin individual	1705	ARCHes	98237 (10 CPU)	7.9G
		RFMix	390443 (10 CPU)	6.18G
Simulated pair admixed individual	3200	ARCHes	188709 (10 CPU)	14.8G
		RFMix	378838 (10 CPU)	7.27G
Simulated Latino individual	100	ARCHes	6814 (1 CPU)	0.53G
		RFMix	389388 (10 CPU)	8.07G

Appendix



Algorithm 1 Diploid HMM forward procedure for a sequence $\mathbf{x}$ of D diploid genotypes (values are all homozygous 0 or 1, heterozygous, or missing) and a model $\mathbf{M}$ of $D+1$ levels. $\mathbf{M}$ has a start state $\mathbb{S}$, a transition function $t(u, a)$ that maps a haplotype model state u to the state at the next level associated with the allele a transition ($a \in 0, 1$), and a transition probabilty function $\rho(u, a)$ that maps a haplotype model state u to the transition probability associated with allele a. The procedure populates f, where $f(u_1, u_2)$ is the forward likelihood of a diploid HMM state (u_1, u_2). It also stores states of the diploid HMM that are consistent with the genome at each level and their outgoing transitions (and the probabilities associated with those transitions) to a data structure α (so that the genotype need not be re-examined during the backward procedure). The *optional* subroutine TRIM removes the diploid HMM states in a set with the lowest f values. It is often possible to remove a large proportion of states and yet keep (*e.g.,*) 99.9999% of the likelihood mass contained in the set of SNPs. We use TRIM only for reasons of efficiency.

```
procedure DIPLOID-FORWARD($\mathbf{x}, w, \mathbf{M}_w$)
    Let $D_w$ be the number of SNPs in $\mathbf{M}_w$
    Let $\mathcal{W}(\mathbf{x}, w)$ be the subsequence of genotypes in $\mathbf{x}$ that correspond to the SNPs in window $w$.
    Let $\mathbb{S}$ be the start state of model $\mathbf{M}_w$
    Let $t$ and $\rho$ be $\mathbf{M}_w$'s transition functions, mapping a state to a state and probability, respectively
    Let $\alpha(d)$ be an initially empty data structure containing diploid HMM states at level $d$,
      and the states they transition to with what probability
    $f(\mathbb{S}, \mathbb{S})) \leftarrow 1$ // both haplotypes must start in the haplotype model start state
    Add state $(\mathbb{S}, \mathbb{S})$ to $\alpha(0)$ with no outgoing transitions (yet)
    for $d \in 0, 1, 2, ..., D_w - 1$ do
        for each diploid HMM state $(u_1, u_2) \in \alpha(d)$ do
            Let $P$ be an initially empty list of diploid HMM state transitions and their likelihoods
            if $\mathcal{W}(\mathbf{x}, w)_{d+1}$ is HOMOZYGOUS 0 then
                Add $((t(u_1, 0), t(u_2, 0)), \rho(u_1, 0) \times \rho(u_2, 0))$ to $P$
            if $\mathcal{W}(\mathbf{x}, w)_{d+1}$ is HOMOZYGOUS 1 then
                Add $((t(u_1, 1), t(u_2, 1)), \rho(u_1, 1) \times \rho(u_2, 1))$ to $P$
            if $\mathcal{W}(\mathbf{x}, w)_{d+1}$ is HETEROZYGOUS then // Consider both possibilities
                Add $((t(u_1, 0), t(u_2, 1)), \rho(u_1, 0) \times \rho(u_2, 1))$ to $P$
                Add $((t(u_1, 1), t(u_2, 0)), \rho(u_1, 1) \times \rho(u_2, 0))$ to $P$
            if $\mathcal{W}(\mathbf{x}, w)_{d+1}$ is MISSING then // Consider all possibilities
                Add $((t(u_1, 0), t(u_2, 0)), \rho(u_1, 0) \times \rho(u_2, 0))$ to $P$
                Add $((t(u_1, 0), t(u_2, 1)), \rho(u_1, 0) \times \rho(u_2, 1))$ to $P$
                Add $((t(u_1, 1), t(u_2, 0)), \rho(u_1, 1) \times \rho(u_2, 0))$ to $P$
                Add $((t(u_1, 1), t(u_2, 1)), \rho(u_1, 1) \times \rho(u_2, 1))$ to $P$
            for $((v_1, v_2), p)$ in $P$ do // $(u_1, u_2)$ can transition to $(v_1, v_2)$ with probability $p$
                if $(v_1, v_2)$ is not in $\alpha(d+1)$ then // Lookup in constant time with perfect hash on serial numbers of $v_1$,$v_2$
                    initialize $f(v_1, v_2) \leftarrow 0$ and add $(v_1, v_2)$ to $\alpha(d+1)$
                $f(v_1, v_2) \leftarrow f(v_1, v_2) + f(u_1, u_2) \times p$ // Update $f(v_1, v_2)$ to include the new transition
                Add $((u_1, u_2) \rightarrow (v_1, v_2), p)$ to the set of outgoing transitions for state $(u_1, u_2)$ in $\alpha(d)$
        TRIM($\alpha(d+1), f$) // Optionally remove some of the lowest-likelihood diploid HMM states from $\alpha(d+1)$
    return $f, \alpha$
```

Algorithm 2 Diploid HMM backward procedure (see DIPLOID-FORWARD). The procedure populates b, where $b(u_1, u_2)$ is the backward likelihood of a diploid HMM state (u_1, u_2). D is the number of SNPs in the window associated with the haplotype model, α is the set of diploid HMM states at each level and their probabilistic outgoing transitions as computed by DIPLOID-FORWARD.

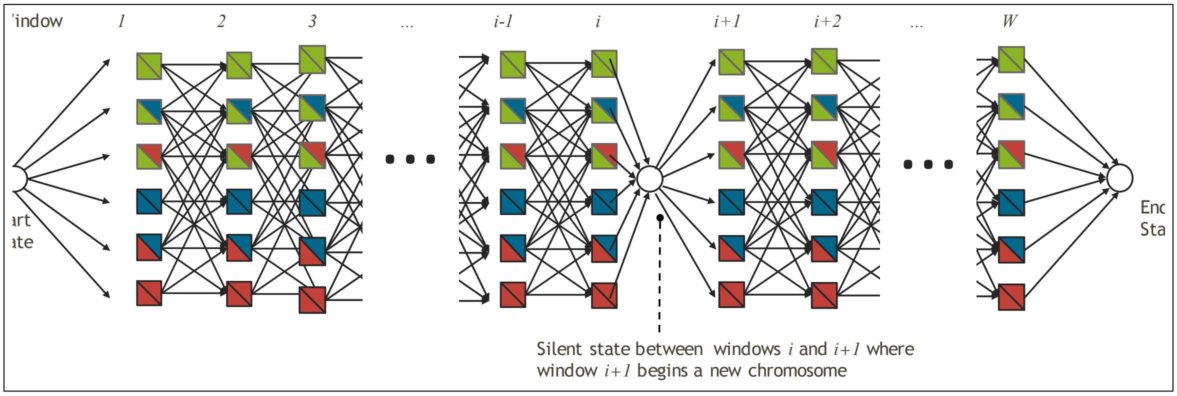
```
procedure DIPLOID-BACKWARD($D, \alpha$)
    Initialize $b(u_1, u_2) \leftarrow 0$ for all diploid HMM states $(u_1, u_2)$
    for $d \in D-1, D-2, ..., 2, 1, 0$ do
        for each diploid HMM state $(u_1, u_2) \in \alpha(d)$ do // $(u_1, u_2)$ is a source state
            for each diploid HMM state $(v_1, v_2)$ such that $((u_1, u_2) \rightarrow (v_1, v_2), p) \in \alpha(d)$ do// $(v_1, v_2)$ is a destination state
                // $(u_1, u_2)$ transitions to $(v_1, v_2)$ with probability $p$
                $b(u_1, u_2) \leftarrow b(u_1, u_2) + b(v_1, v_2) \times p$
    return $b$
```

Algorithm 3 Diploid HMM forward-backward procedure (see DIPLOID-FORWARD and DIPLOID-BACKWARD). The procedure populates f and b, where $f(u_1, u_2)$ is the ("forward") likelihood that a path through the diploid HMM ends in state (u_1, u_2) after emitting d alleles (where d is the level of u_1 and u_2) of a haplotype in the input genotype sequence $\mathbf{x}$, and $b(u_1, u_2)$ is the likelihood of all paths from (u_1, u_2) to the end state. The probability $P_d(u_1, u_2|\mathbf{x})$ that the haplotypes of genotype sequence $\mathbf{x}$ belongs to clusters u_1 and u_2 is calculated as $\frac{f(u_1,u_2)b(u_1,u_2)}{b(\mathbb{S},\mathbb{S})}$, where $\mathbb{S}$ is the start state of model $\mathbf{M}$ and f and b are computed by this procedure.

```
procedure DIPLOID-FORWARD-BACKWARD($\mathbf{x}, w, \mathbf{M}_w$)
    $f, \alpha \leftarrow$DIPLOID-FORWARD($\mathbf{x}, w, \mathbf{M}_w$)
    Let $D_w$ be the number of SNPs in $\mathbf{M}_w$
    $b \leftarrow$DIPLOID-BACKWARD($D_w, \alpha$)
    return $f, b$
```

A Computing Forward-Backward on the Genome-Wide Ancestry HMM, and Updating π and τ Transition Probability Parameters

The genome-wide ancestry HMM computes the likelihoods that a test instance's genotype sequence, $\mathbf{t}$, in a genomic window w (denoted $\mathbf{t}_w$) is explained by populations p and q for a set of populations and genomic windows. It is parameterized by $\pi_{\mathbf{t}}$ and $\tau_{\mathbf{t}}$,[1] which are typically learned for a specific test instance. The ancestry HMM representing a K populations and a set of SNPs on multiple chromosomes has a single silent (non-emitting) state before the first, after the last, and in-between each chromosome, and a series of $\frac{(K+1)\times K}{2}$ emitting states for each window of each chromosome, each corresponding to a population assignment (p, q) with $1 \leq p \leq q \leq K$. Figure 1 illustrates such a genome-wide HMM with $K = 3$ populations. Let the emitting state corresponding to window w and population assignment (p, q) be denoted $S_{w,p,q}$. Its emission probability $P(\mathbf{t}_w|p, q)$ is precomputed and fixed based on the genotype $\mathbf{t}_w$ in window w. Let S_c represent the silent state that preceeds the emitting states correpsonding to windows on chromosome c. Thus the start state of the HMM is S_1 (and if the HMM represents C chromosomes, the end state would be S_{C+1}). Let $\mathcal{C}(c)$ map a chromosome number to the window that begins the chromosome. Then, our HMM transitions from silent states to emitting states $S_c \rightarrow S_{\mathcal{C}(c),p,q}$, from emitting states in the last window of a chromosome to a silent state $S_{\mathcal{C}(c+1)-1,p,q} \rightarrow S_{c+1}$, and from emitting states to emitting states for windows w that are not the first or last in a chromosome $S_{w,p,q} \rightarrow S_{w+1,p',q'}$. A transition from $S_{w,p,q} \rightarrow S_{w+1,p',q'}$ represents a change in population assignment between windows w and $w + 1$ if $p \neq p'$ or $q \neq q'$.



Silent state between windows i and $i+1$ where window $i+1$ begins a new chromosome

Figure 1: A genome-wide ancestry HMM, consisting of 3 ancestral populations (green, blue, and red) and W windows.

The transition probabilities from a silent state to an emitting state $S_c \rightarrow S_{\mathcal{C}(c),p,q}$ is $\pi_{\mathbf{t},(p,q)}$, where $\pi_{\mathbf{t}}$ is a learned parameter vector over all possible assignments (p, q) $(1 \leq p \leq q \leq K)$ indicating a global assigment preference. The transition probability from a state in the last window of a chromosome to a silent state $S_{\mathcal{C}(c)-1,p,q} \rightarrow S_c$ is always 1, and transitions between emitting states on the same chromosome, from state (p, q) in window w to state (p', q') (with $p' \leq q'$) in window $w + 1$, are as follows:

$$P(S_{w,p,q} \rightarrow S_{w+1,p',q'}|\pi_{\mathbf{t}}, \tau_{\mathbf{t}}) = \begin{cases} 1 - \tau_{\mathbf{t}} & \text{if } p = p' \text{ and } q = q' \\ \tau_{\mathbf{t}} \times \frac{\pi_{\mathbf{t},(p',q')}}{\sum_{(p'',q'')|p'' \leq q'', p=p'' \oplus q=q''} \pi_{\mathbf{t},(p'',q'')}} & \text{if } p = p' \oplus q = q' \\ 0 \text{ (transition ignored)} & \text{otherwise} \end{cases} \quad (1)$$

[1] The subscript $\mathbf{t}$ may be dropped from these and other terms when there is only one test genotype instance in question.

where $\tau_{\mathbf{t}}$ is a parameter representing the probability of changing population assignment that enforces the bias against changing population assignments from window to window ($\oplus$ is the exclusive or operator). We initialize $\pi_{\mathbf{t}}$ to a uniform distribution, and $\tau_{\mathbf{t}}$ to a (typically low) initial value and learn $\pi_{\mathbf{t}}$ and $\tau_{\mathbf{t}}$ using expectation-maximization over a number of iterations (similar to the standard Baum-Welch algorithm [Rabiner, 1989], except that $\pi_{\mathbf{t}}$ and $\tau_{\mathbf{t}}$ are tied to all state transition probabilities).

Let $F_{\mathbf{t}}(s)$ be the forward probability, the sum probability of all paths through the Ancestry HMM (as opposed to the *haplotype* HMM used to calculate per-window emission probabilities) that start in the start state and end in state s (including the emission of state s) and $B_{\mathbf{t}}(s)$ be the backward probability of all paths through the HMM that start in state s (excluding emission) and end in the end state. F and B are computed recursively as follows.

$$F_{\mathbf{t}}(S_1) = 1. \tag{2}$$

For the emitting states in the first window of a chromosome,

$$F_{\mathbf{t}}(S_{\mathcal{C}(c),p',q'}) = F_{\mathbf{t}}(S_c) \times \pi_{\mathbf{t},(p',q')} \times P(\mathbf{t}_{\mathcal{C}(c)}|p,q) \tag{3}$$

for all p' and q'. When a window w is not the first window of a chromosome,

$$F_{\mathbf{t}}(S_{w,p',q'}) = \sum_{p=1}^{K}\sum_{q=p}^{K} F_{\mathbf{t}}(S_{w-1,p,q}) \times P(S_{w-1,p,q} \to S_{w,p',q'}|\pi_{\mathbf{t}},\tau_{\mathbf{t}}) \times P(\mathbf{t}_w|p,q) \tag{4}$$

where $P(S_{w-1,p,q} \to S_{w,p',q'}|\pi_{\mathbf{t}},\tau_{\mathbf{t}})$ is given by (1). The forward probability of the silent state preceding chromosome c is

$$F_{\mathbf{t}}(S_c) = \sum_{p=1}^{K}\sum_{q=p}^{K} F_{\mathbf{t}}(S_{\mathcal{C}(c)-1,p,q}) \tag{5}$$

Similarly, if there are C chromosomes in the model,

$$B_{\mathbf{t}}(S_{C+1}) = 1. \tag{6}$$

For the last window on chromosome c,

$$B_{\mathbf{t}}(S_{\mathcal{C}(c+1)-1,p,q}) = B_{\mathbf{t}}(S_{c+1}) \tag{7}$$

for all p and q. When window w is not the last window on a chromosome,

$$B_{\mathbf{t}}(S_{w,p,q}) = \sum_{p'=1}^{K}\sum_{q'=p'}^{K} P(S_{w,p,q} \to S_{w+1,p',q'}|\pi_{\mathbf{t}},\tau_{\mathbf{t}}) \times P(\mathbf{t}_{w+1}|p',q') \times B_{\mathbf{t}}(S_{w+1,p',q'}). \tag{8}$$

Finally, for the silent state preceding chromosome c,

$$B_{\mathbf{t}}(S_c) = \sum_{p'=1}^{K}\sum_{q'=p'}^{K} \pi_{\mathbf{t},p',q'} \times P(\mathbf{t}_{\mathcal{C}(c)}|p',q') \times B_{\mathbf{t}}(S_{\mathcal{C}(c),p',q'}). \tag{9}$$

After computing $\mathbf{F_t}$ and $\mathbf{B_t}$, we compute the expectation for each $\pi_{\mathbf{t},(p,q)}$ as

$$\mathbf{E}(\pi_{\mathbf{t},(p,q)}) = \sum_{c=1}^{C}\sum_{w=\mathcal{C}(c)}^{\mathcal{C}(c+1)-1} F_{\mathbf{t}}(w,p,q) \times B_{\mathbf{t}}(w,p,q) \tag{10}$$

and reset each $\pi_{\mathbf{t},(p,q)}$ to the value that maximizes the likelihood of $\mathbf{E}(\pi_{\mathbf{t},(p,q)})$:

$$\pi_{\mathbf{t},(p,q)} \leftarrow \frac{\mathbf{E}(\pi_{\mathbf{t},(p,q)})}{\sum_{p'=1}^{K}\sum_{q'=p'}^{K}\mathbf{E}(\pi_{\mathbf{t},(p',q')})}. \tag{11}$$

We learn $\tau_{\mathbf{t}}$ in a similar fashion, by updating it based on the expected number of transitions that do not change assignment, compared to all transitions. If there are C chromosomes,

$$\tau_{\mathbf{t}} \leftarrow 1 - \frac{\sum_{c=1}^{C}\sum_{w=\mathcal{C}(c)}^{\mathcal{C}(c+1)-1}\sum_{p=1}^{K}\sum_{q=p}^{K} F_{\mathbf{t}}(S_{w,p,q}) \times P(S_{w,p,q} \rightarrow S_{w+1,p,q}|\pi_{\mathbf{t}}, \tau_{\mathbf{t}}) \times B_{\mathbf{t}}(S_{w+1,p,q})}{\sum_{c=1}^{C}\sum_{w=\mathcal{C}(c)}^{\mathcal{C}(c+1)-1}\sum_{p=1}^{K}\sum_{q=p}^{K}\sum_{p'=1}^{K}\sum_{q'=p'}^{K} F_{\mathbf{t}}(S_{w,p,q}) \times P(S_{w,p,q} \rightarrow S_{w+1,p',q'}|\pi_{\mathbf{t}}, \tau_{\mathbf{t}}) \times B_{\mathbf{t}}(S_{w+1,p',q'})} \tag{12}$$

B Computing the Viterbi Path

The Viterbi path is the single most likely path (relative to a genotype sequence $\mathbf{t}$) through the genome-wide HMM $\mathbf{V_t} = \langle V_{\mathbf{t},1}, V_{\mathbf{t},2}, ..., V_{\mathbf{t},W}\rangle$, where each $V_{\mathbf{t},w}$ is an assignment (p, q) in a window w, $1 \leq w \leq W$.

To compute $\mathbf{V}$, we must first define $\mathbf{M_t}$, where $M_{\mathbf{t}}(s)$ is the probability of the most likely path through the HMM that start in the start state and end in state s (including the emission of state s), analagous to the forward probability $F_{\mathbf{t}}(s)$ in Appendix A, but referring to the probability of the single most likely path instead of the sum probability of all paths.

$$M_{\mathbf{t}}(S_1) = 1. \tag{13}$$

For the emitting states in the first window of a chromosome,

$$M_{\mathbf{t}}(S_{\mathcal{C}(c),p',q'}) = M_{\mathbf{t}}(S_c) \times \pi_{\mathbf{t},(p',q')} \times P(\mathbf{t}_{\mathcal{C}(c)}|p', q') \tag{14}$$

for all p' and q'. When a window w is not the first window of a chromosome,

$$M_{\mathbf{t}}(S_{w,p',q'}) = \underset{1\leq p\leq q\leq K}{\operatorname{argmax}} M_{\mathbf{t}}(S_{w-1,p,q}) \times P(S_{w-1,p,q} \rightarrow S_{w,p',q'}|\pi_{\mathbf{t}}, \tau_{\mathbf{t}}) \times P(\mathbf{t}_w|p', q') \tag{15}$$

And for a silent state that is not the start state,

$$M_{\mathbf{t}}(S_c) = \underset{1\leq p\leq q\leq K}{\operatorname{argmax}} M_{\mathbf{t}}(S_{\mathcal{C}(c)-1,p,q}) \tag{16}$$

The Viterbi path $\mathbf{V}$ is then defined for windows that are the last window in a chromosome, c, as

$$V_{\mathbf{t},\mathcal{C}(c+1)-1} = \underset{1\leq p\leq q\leq K}{\operatorname{argmax}} M_{\mathbf{t}}(S_{\mathcal{C}(c+1)-1}, p, q), \tag{17}$$

and for all other windows as

$$V_{\mathbf{t},w} = \underset{1\leq p'\leq q'\leq K}{\operatorname{argmax}} P(S_{w,p,q} \rightarrow S_{w+1,p',q'}|\pi_{\mathbf{t}}, \tau_{\mathbf{t}}) \times P(\mathbf{t}_{w+1}|p', q') \times M_{\mathbf{t}}(S_{w+1,p',q'}). \tag{18}$$

C Computing Path Samples

Let *choose* be a operator that chooses an argument with a probability relative to an expression so that $\underset{x\in D}{\text{choose}}\, f(x)$ returns x with probability $\frac{f(x)}{\sum_{x'\in D} f(x')}$. Then a stochastic path $\mathbf{Q}$ for a genomic sequence $\mathbf{t}$ is defined over all windows $1 \leq w \leq W$ as follows. For windows that are last in a chromosome, c,

$$Q_{\mathbf{t},\mathcal{C}(c+1)-1} = \underset{p,q}{\text{choose}}\, F_{\mathbf{t}}(S_{\mathcal{C}(c+1)-1,p,q}). \tag{19}$$

For other windows w,

$$Q_{\mathbf{t},w} = \underset{p,q}{\text{choose}}\, F_{\mathbf{t}}(S_{w,p,q}) \times P(S_{w,p,q} \to S_{w+1,Q_{\mathbf{t},w+1}} | \pi_{\mathbf{t}}, \tau_{\mathbf{t}}) \times P(\mathbf{t}_{w+1} | Q_{\mathbf{t},w+1}). \tag{20}$$