BIOINFORMATICS

Vol. 26 ISMB 2010, pages i199–i207
doi:10.1093/bioinformatics/btq187

Efficient genome ancestry inference in complex pedigrees with inbreeding

Eric Yi Liu[1,*], Qi Zhang[2], Leonard McMillan[1], Fernando Pardo-Manuel de Villena[3] and Wei Wang[1,*]

[1]Department of Computer Science, University of North Carolina at Chapel Hill, [2]Department of Biostatistics, University of Washington and [3]Department of Genetics, University of North Carolina at Chapel Hill, USA

ABSTRACT

Motivation: High-density SNP data of model animal resources provides opportunities for fine-resolution genetic variation studies. These genetic resources are generated through a variety of breeding schemes that involve multiple generations of matings derived from a set of founder animals. In this article, we investigate the problem of inferring the most probable ancestry of resulting genotypes, given a set of founder genotypes. Due to computational difficulty, existing methods either handle only small pedigree data or disregard the pedigree structure. However, large pedigrees of model animal resources often contain repetitive substructures that can be utilized in accelerating computation.

Results: We present an accurate and efficient method that can accept complex pedigrees with inbreeding in inferring genome ancestry. Inbreeding is a commonly used process in generating genetically diverse and reproducible animals. It is often carried out for many generations and can account for most of the computational complexity in real-world model animal pedigrees. Our method builds a hidden Markov model that derives the ancestry probabilities through inbreeding process without explicit modeling in every generation. The ancestry inference is accurate and fast, independent of the number of generations, for model animal resources such as the Collaborative Cross (CC). Experiments on both simulated and real CC data demonstrate that our method offers comparable accuracy to those methods that build an explicit model of the entire pedigree, but much better scalability with respect to the pedigree size.

Contact: weiwang@cs.unc.edu

1 INTRODUCTION

Model organisms, such as laboratory mice, are frequently *bred* or *crossed* in order to study genetic influences (Chia *et al*., 2005; Churchill *et al*., 2002; Valdar *et al*., 2006). Often, such animal resources are generated using prescribed breeding system to ensure diversity and reproducibility, which leads to complex pedigree structure consisting of many generations. Through recombination, the DNA sequences of founder organisms are intermixed in each generation. A DNA sequence of any descendant organism is a mosaic of its founders' DNA segments. As recombinations at each breeding stage cannot be observed directly, it is of great interest to infer the ancestry of resulting DNA sequences. In other words, which part of a resulting DNA sequence is inherited from which founder.

The vast majority of the sequence variations are attributed to single base pair mutations known as *single nucleotide polymorphism* (SNPs), thus making SNPs ideal for resolving the genome ancestry problem. The set of SNPs on the same chromosome constitutes a *haplotype*. While any of the four nucleotides (A,T,C,G) is possible, in practice nearly all SNPs appear in only two variations. This results from the fact that SNPs originate as mutations, which are rare events within a vast genome. It is therefore convenient to encode a SNP allele as a binary value and represent haplotypes as binary sequences. Modern high-throughput genotyping technologies are unable to distinguish between the two haplotypes of a diploid organism. Instead, a *genotype sequence* is measured where, at each SNP site, one of three possibilities is observed ($\{00, 01, 11\}$, since 10 cannot be distinguished from 01).

Using the genotype representation for DNA sequences, the genome ancestry problem estimates the origin of each genotype from a descendant's sequence given the genotype sequences of its distant founders. To achieve high resolution, dense SNP markers are used (tens of thousands on each chromosome). Knowledge of genotype's ancestry is particularly useful in many problems such as studying the structure and history of haplotype blocks (Gabriel *et al*., 2002; Schwartz *et al*., 2004; Zhang *et al*., 2002), and mapping quantitative trait loci (QTLs; Mott *et al*., 2000; Valdar *et al*., 2006). In these studies, a probabilistic interpretation is favored over discrete solutions, due to the prevalence of ambiguities and measurement errors.

The genome ancestry problem is closely related to haplotype inference with pedigree data. Inferring haplotypes in a pedigree often involves solving the inheritance flow of alleles at each generation. On the other hand, given the genome ancestry information, it is straightforward to reconstruct the descendant haplotypes. As pedigree analysis is Non-deterministic Polynomial-time (NP) hard (Piccolboni and Gusfield, 2003), existing algorithms are either approximate or suffer exponential running times. Among the maximum likelihood approaches, methods (Abecasis *et al*., 2002; Gudbjartsson *et al*., 2005; Kruglyak *et al*., 1996) based on the Lander–Green algorithm (Lander and Green, 1987) are often favored because their running time is linear to the number of markers. MERLIN (Abecasis *et al*., 2002), an implementation based on sparse binary trees, is one of the most successful pedigree analysis programs. Unfortunately, methods based on Lander–Green algorithms are limited to pedigrees of moderate size since the running time grows exponentially with pedigree size. MCMC sampling methods (Jensen and Kong, 1999; Sobel and Lange, 1996) have been proposed to address larger pedigrees. But their computing time can be substantial when applied to a large number of tightly linked markers. Other efforts include rule-based methods (Li and Jiang, 2005; Qian and Beckmann, 2002), which approximates a solution by minimizing recombinations in the pedigree (MRHC).

*To whom correspondence should be addressed.

PedPhase (Li and Jiang, 2005), which employs an effective integer linear programming (ILP) formulation, has been widely used in solving the MRHC.

Current haplotyping methods for pedigrees are incapable of solving the genome ancestry problem in animal resources for the following reasons: (i) Pedigrees of model animal resources often contain large number of generations to ensure diversity and reproducibility. (ii) None or few of the intermediate generations is genotyped due to the size of the resources. (iii) A large number of dense markers are genotyped to achieve fine resolution. As a concrete example, more than 1000 lines have been started in the Collaborative Cross (CC) project (Churchill *et al*., 2002). Each line is expected to undergo at least 23 generations before reaching 99% inbred. Hundreds of mice of various generations were genotyped, but on average only few are from the same line. The missing genotypes make the search space extraordinarily large.

Other computationally efficient approaches for solving the genome ancestry problem have largely ignored the breeding scheme. While breeding design does not determine the locations of recombination, it often places constraints on the possible ancestry choices at a single site and at neighboring sites. The genome ancestry problem was modeled as a combinatorial optimization problem in Zhang *et al*. (2008). By minimizing recombinations, discrete solutions are generated. Mott and coworker (Mott *et al*., 2000; Valdar *et al*., 2006) has proposed an approach using hidden Markov model (HMM) for ancestry inference in HAPPY, a QTL mapping tool suite for association studies. All founder pairs are considered as possible hidden states for emitting the observed genotype at each site. Besides founder genotypes, no pedigree data are used in these two approaches.

There have also been many efforts to analyze pedigree by identifying symmetries in HMM state space (Browning and Browning, 2002; Donnelly, 1983; Geiger *et al*., 2009; McPeek, 2002). The states are then grouped to accelerate the calculation. However, finding the maximal grouping is non-trivial. In real-world problems, only obvious symmetries such as founder phase and chain structure in pedigree can be best utilized.

Besides model organisms, the genetic ancestry problem has been studied for human individuals that have recently been admixed from a set of isolated populations, instead of a set of founders (Pasaniuc *et al*., 2009; Sankararaman *et al*., 2008; Sundquist *et al*., 2008; Tang *et al*., 2006). In this problem, pedigree structure is usually not present. Efficient methods have been developed to handle large-scale datasets (Sankararaman *et al*., 2008; Sundquist *et al*., 2008; Tang *et al*., 2006).

Leveraging the observation that large animal resource pedigrees often contain repetitive substructures, we propose a method that can efficiently handle complex pedigrees with inbreeding which is an important process in generating animal resources. Using a pair of dependent quaternary indicators to capture all recombinations in the inbreeding history, our method achieves accurate ancestry inference without explicit modeling in every generation. By encoding the inbreeding model into the inheritance vectors, we design a Lander–Green-like algorithm whose running time remains constant with respect to the number of inbreeding generations. Our method is implemented and evaluated on the CC breeding design (Churchill *et al*., 2002) with dense SNP data. Experiments show that, our approach generates accurate results efficiently on data that cannot be handled by existing pedigree haplotyping software. Compared with HAPPY, which does not consider pedigree structure, our approach significantly reduces ambiguities and errors in ancestry inference.

2 THE GENOME ANCESTRY PROBLEM

Given a pair of chromosomes, we consider L SNP markers ordered by their chromosomal locations. For each SNP site, we use 0 and 1 to encode the two possible values. The genotype at each site is the unordered combination of corresponding alleles from both chromosomes, which can assume one of three values: 00, 01, 11. A genotype sequence is a genome-ordered set of genotypes denoted as: $G=g_1...g_l...g_L, (g_l \in \{00,01,11\})$. A haplotype $H=h_1...h_l...h_L$ consists of alleles from one of the chromosomes where $h_l \in \{0,1\}$.

Consider a pedigree containing a set of founders $FS=\{F1,...,F_N\}$ and a descendant of interest. We denote the set of founder genotype sequences by $\{G_{F1},...,G_{F_N}\}$, all of which are given. Given the genotype sequence, G_D, of the descendant generated through the pedigree structure, its genome ancestry is to be determined. Every genotype g_l in G_D inherits its alleles from two founders, say F_A and F_B. We refer to the founder pair (F_A,F_B) as the genome ancestry at site l of genotype sequence G_D. We want to estimate, for every SNP site l, the probability $P(\text{Ancestry}(g_l)=(F_A,F_B))$ for every founder pair $(F_A,F_B) \in FS \times FS$. Note that founder pairs are unordered $((F_A,F_B)=(F_B,F_A))$, and it is possible that $F_A=F_B$.

3 MODELING INHERITANCE IN PEDIGREE

We start from the standard Lander–Green approach to model a pedigree: at each SNP site, an inheritance indicator is used to indicate the outcome of each meiosis. These inheritance indicators together form the inheritance vector. Since a child haplotype inherits its allele from either the paternal or maternal sequence, an inheritance indicator is a binary variable. For a pedigree with n non-founder animals, there are $2\times n$ inheritance indicators at each site. Hence, the inheritance vector at site l, v_l, can be defined as a binary sequence of length $2\times n$. An instance of v_l specifies a possible configuration of inheritance flow at site l of all animals in the pedigree. When SNP markers are dense enough, we can assume at most one recombination between two sites in generating one haplotype. If a recombination happens between site l and $l+1$, the corresponding inheritance indicator will have different states for the two sites. Hence, to measure the number of recombinations between l and $l+1$ in the whole pedigree, we can count the difference in bits between v_l and v_{l+1}. The probability of having d recombinations between l and $l+1$ is $\theta^d(1-\theta)^{2n-d}$, where θ is the recombination fraction.

The length of inheritance vector grows linearly with the number of animals in the pedigree and this causes exponential growth in the number of possible inheritance patterns. Considering the fact that full pedigree analysis is computationally intractable, we overcome the issue by modeling important substructure in breeding systems as a shortcut to efficient computation. Our first natural choice of substructure is inbreeding: (i) Inbreeding is often used in model animal resources to generate genetically diverse and/or reproducible descendants. (ii) Inbreeding is often carried out for many generations and each generation elongates the inheritance vectors by 4 bits. Hence, if a pedigree involves inbreeding, the inbreeding generations often account for most of the computational complexity. We seek an aggregated inheritance indicator to replace the collection of many

inheritance indicators in the inbreeding process. Such an aggregated indicator can be encoded in much shorter length and incorporated into the inheritance vector. If the state and transition probability of the aggregated indicator can be modeled efficiently, full pedigree analysis will become feasible on these animal resources. In the next section, we explain how inheritance in inbreeding generations can be modeled as an aggregated indicator.

3.1 Modeling inbreeding generations

During inbreeding, offspring are produced by sibling matings for many generations. At each generation, four new haplotypes are formed by recombining the four haplotypes from the previous generation. The inbreeding process at a single site is shown in Figure 1a. We denote the beginning generation of inbreeding as generation I_0. Observe that, at each site, because of the symmetry of inbreeding structure, the four alleles at generation I_0 have equal probabilities to be passed down to any haplotypes after I_1. Thus, for a descendant haplotype at generation I_k $(k>2)$, we can simply replace the lattice of binary inheritance indicators by a single quaternary indicator. Each choice of the quaternary indicator has $1/4$ probability. Two quaternary indicators are needed for the two haplotypes of a I_k descendant (Fig. 1b). However, the two quaternary indicators are not independent as the two haplotypes share the same inbreeding history until I_{k-1}. To model this dependency between the two quaternary indicators, we find out the transition events and probabilities of the pair of indicators. The grouped pair is then used as an aggregated inheritance indicator as discussed above.

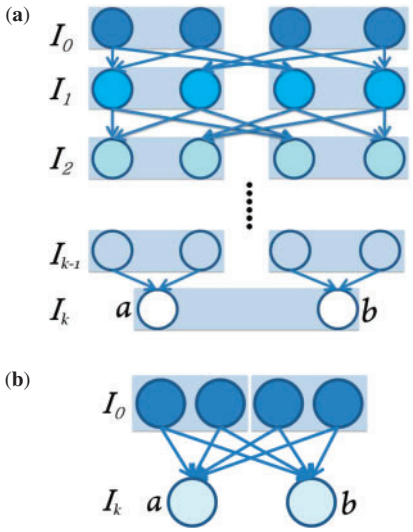

Fig. 1. (**a**) Lattice of binary inheritance indicators representing the inheritance pattern of an inbreeding process at a single site. (**b**) An equivalent quaternary indicator representation.

We label the four I_0 haplotypes as $1,2,3,4$. We then denote by a,b the two I_k descendant haplotypes and $S(a_l),S(b_l)$ are their I_0 sources at site l, i.e., $S(a_l),S(b_l)\in\{1,2,3,4\}$. Their I_0 sources along the chromosome is denoted by $S(a),S(b)\in\{1,2,3,4\}^L$. A transition happens in $S(a)$ between site l and $l+1$ if $S(a_l)\neq S(a_{l+1})$. We consider, between two adjacent sites, l and $l+1$, all the possible transitions from $S(a_l),S(b_l)$ to $S(a_{l+1}),S(b_{l+1})$ (Table 1).

Note that:

$$P_{EE0}+P_{EN1}+P_{EE2}+P_{EN2}=P(S(a_l)=S(b_l))=$$
$$P_{EE0}+P_{EE2}+P_{NE1}+P_{NE2}=P(S(a_{l+1})=S(b_{l+1}))$$

and

$$P_{NE1}+P_{NN0}+P_{NN1}+P_{NN2}+P_{NE2}=P(S(a_l)\neq S(b_l))=$$
$$P_{EN1}+P_{EN2}+P_{NN0}+P_{NN1}+P_{NN2}=P(S(a_{l+1})\neq S(b_{l+1}))$$

The prior probability $P(S(a_l)=S(b_l))$ at any site l is called the *inbreeding coefficient* (Wright, 1922). To calculate the probability, let IC_k denote the inbreeding coefficient at generation I_k. IC_k can be computed recursively using $IC_k=\sum_{j=0}^{k-2}\left(\frac{1}{2}\right)^{k-j}\times(1+IC_j)$.

Next, we derive the probabilities in Table 1. Consider that any transition in $S(a)$ or $S(b)$ is caused by one or more recombinations in the inbreeding process (Fig. 1a). Our calculation is based on the assumption that the recombination fraction, θ, is reasonably small. Hence, for any haplotype c at generation I_j $(1\leq j\leq k)$, we assume that any single transition in $S(c)$ is solely caused by one recombination in generating c or its ancestor haplotypes. In other words, a single transition in $S(c)$ is not the result of multiple recombinations in the pedigree. Our assumption is generally true for dense SNP markers where θ is usually well below 0.001. Under the assumption, if a

Table 1. All possible transitions of $S(a),S(b)$

Site l	Possible transitions	Site $l+1$	Denote By
$S(a_l)=S(b_l)$	Neither $S(a)$ or $S(b)$ transitions.	$S(a_{l+1})=S(b_{l+1})$	P_{EE0}
	Either $S(a)$ or $S(b)$ transitions, but not both.	$S(a_{l+1})\neq S(b_{l+1})$	P_{EN1}
	Both $S(a)$ and $S(b)$ transitions to same value.	$S(a_{l+1})=S(b_{l+1})$	P_{EE2}
	Both $S(a)$ and $S(b)$ transitions, but to different values.	$S(a_{l+1})\neq S(b_{l+1})$	P_{EN2}
$S(a_l)\neq S(b_l)$	Neither $S(a)$ nor $S(b)$ transitions.	$S(a_{l+1})\neq S(b_{l+1})$	P_{NN0}
	Either $S(a)$ or $S(b)$ transitions, but not both. $S(a)$ and $S(b)$ become equal after the transition.	$S(a_{l+1})=S(b_{l+1})$	P_{NE1}
	Either $S(a)$ or $S(b)$ transitions, but not both. $S(a)$ and $S(b)$ remain different after the transition.	$S(a_{l+1})\neq S(b_{l+1})$	P_{NN1}
	Both $S(a)$ and $S(b)$ transition. $S(a)$ and $S(b)$ remain different after the transition.	$S(a_{l+1})\neq S(b_{l+1})$	P_{NN2}
	Both $S(a)$ and $S(b)$ transition. $S(a)$ and $S(b)$ become the same after the transition.	$S(a_{l+1})\neq S(b_{l+1})$	P_{NE2}

Each type of transition is denoted by three characters. First two letters indicate the equality of $S(a),S(b)$ before and after the transition. Then followed by a digit indicating the number of transitions in $S(a),S(b)$.

transition in $S(c)$ is caused by a recombination in generating c itself, we define this to be a *lead transition*. Intuitively, a lead transition is one not inherited from its ancestors. A lead transition in c will change the I_0 source of c and all descendant haplotypes inheriting the transition. A lead transition is only possible when the two parental haplotypes of c have different I_0 sources. Hence, between two sites, a haplotype at generation j has a lead transition with probability $\theta\times(1-IC_{j-1})$.

With the inbreeding coefficients calculated, we can derive the marginal probability of observing transition in one of the I_k haplotypes, $P_{1T}=P(S(a_l)\neq S(a_{l+1}))=P(S(b_l)\neq S(b_{l+1}))$. Without loss of generality, we consider $P(S(a_l)\neq S(a_{l+1}))$ for haplotype a. $S(a)$ will transition if a itself or any of its ancestor haplotypes has a lead transition. At generation k, the lead transition happens with probability $\theta\times(1-IC_{k-1})$. For generation $k-1$, there are two possible ancestor haplotypes, each with $\frac{1}{2}\theta\times(1-IC_{k-2})$ chance of causing a transition in $S(a)$. For each generation j from 1 to $k-2$, there are four possible ancestor haplotypes with probability $\frac{1}{4}\theta\times(1-IC_{j-1})$. Consider that, at one site, any two haplotypes from the same generation cannot both be the ancestor of a. Thus, for any generation j, the expected probability of causing transition in $S(a)$ is $\theta\times(1-IC_{j-1})$. Under our assumption, $P(S(a_l)\neq S(a_{l+1}))$ can be expressed by $1-\prod_{j=1}^{k}(1-\theta\times(1-IC_{j-1}))$.

We then derive the probability P_{EE2} that $S(a)$ and $S(b)$ have equal state at site l, and both transition to another state at site $l+1$. This event happens only if a haplotype c at some previous generation is the common ancestor of a,b and c has a lead transition. The probability of c at generation j being the common ancestor of a and b is $\frac{1}{4}IC_{k-j}$. The probability that c has a lead transition is $\theta\times(1-IC_{j-1})$. Again, consider the fact that, at one site, any two haplotypes from the same generation cannot both be the common ancestor of a and b. Thus, the probability of $EE2$ event caused by lead transition at I_j $(1\leq j\leq k-2)$ is $\theta\times(1-IC_{j-1})IC_{k-j}$. Assuming a small θ, P_{EE2} can be calculated by $1-\prod_{j=1}^{k-2}(1-\theta\times(1-IC_{j-1})IC_{k-j})$.

Lastly we consider the probability P_{NN1}. To simplify our discussion, assume that the transition happens in $S(a)$ [i.e. $S(a_l)\neq S(a_{l+1})$] and it inherits a lead transition in haplotype c of generation j. Since $S(a_l)$, $S(a_{l+1})$ and $S(b_l)$ all have different I_0 ancestry, alleles from at least three distinct I_0 haplotypes should be observed at generation $j-1$. Let $P_{\text{Distinct}}(m,j)$ be the probability of observing exactly m distinct I_0 alleles at generation j. $P_{\text{Distinct}}(3,j)$ and $P_{\text{Distinct}}(4,j)$ can be computed recursively using:

$$P_{\text{Distinct}}(4,j)=\frac{1}{4}P_{\text{Distinct}}(4,j-1)$$

$$P_{\text{Distinct}}(3,j)=\frac{1}{2}P_{\text{Distinct}}(3,j-1)+\frac{1}{2}P_{\text{Distinct}}(4,j-1)$$

Then, P_{NN1} is the probability that (i) at least three distinct I_0 alleles are present at generation $j-1$ and (ii) a's ancestor c at generation j has a lead transition between sites l and $l+1$ which is inherited by a (iii) before and after transition, the I_0 source of c is different from that of b. Due to space limitation, we omit the detailed discussion of c at different generations.

Under our assumption of a small θ, P_{NN2},P_{NE2},P_{EN2} are all sufficiently small and can be ignored in calculating other probabilities. The intuition is as follows: if k is small, there are few animals in the inbreeding lattice and the chance of observing multiple transitions is rare; when k becomes larger, the probability $P(S(a_l)\neq S(b_l))$ approaches 0 rapidly and P_{NN2},P_{NE2},P_{EN2} are much smaller than $P(S(a_l)\neq S(b_l))$. With P_{1T}, P_{EE2} and P_{NN1} derived, we can easily solve all the rest probabilities in Table 1:

$$P_{NE1}=P_{EN1}=\frac{1}{2}(2\times(P_{1T}-P_{EE2})-P_{NN1})$$

$$P_{EE0}=IC_k-P_{EE2}-P_{EN1}$$

$$P_{NN0}=1-IC_k-P_{NE1}-P_{NN1}$$

P_{NN2},P_{NE2},P_{EN2} are approximated by a small probability $P_{NE1}\times P_{NE1}$. We use simulation to validate the probabilities derived above. The results are shown in Figure 2. For θ around 0.01, our method gives reasonably close approximation. For θ below 0.001, our method is very accurate. The recombination fraction between dense SNP markers is usually well below 0.001.

So far we have derived all event probabilities in Table 1. The transition probability from $(S(a_l),S(b_l))$ to $(S(a_{l+1}),S(b_{l+1}))$ is the corresponding probability in Table 1 conditioned on $P(S(a_l)=S(b_l))$ or $P(S(a_l)\neq S(b_l))$.

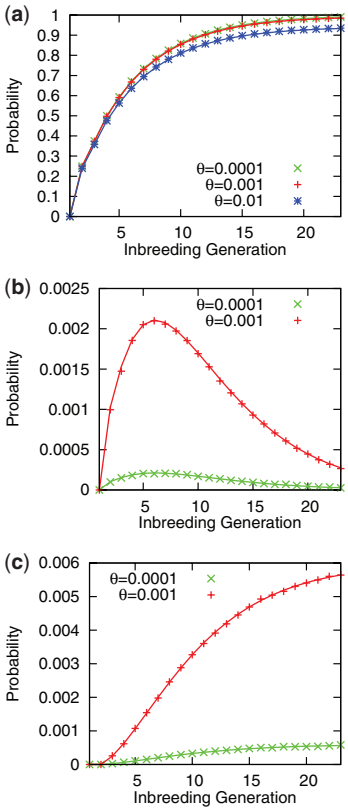

Fig. 2. Comparison of predicted probabilities and observed probabilities from 10 000 000 simulations. The data points in the figures are observed probabilities from simulations. The curves are derived from our formulas. (**a**) Predicted and simulated P_{EE0} for $\theta=0.01,0.001,0.0001$. (**b**) Predicted and simulated $P_{EN1}=P_{NE1}$ for $\theta=0.001,0.0001$. (**c**) Predicted and simulated P_{EE2} for $\theta=0.001,0.0001$. We do not plot the case of $\theta=0.01$ in (b) and (c) because the values are much larger than that of the other two θ values.

3.2 Integrating the inbreeding model

We have argued that each inbreeding process can be modeled by two quaternary indicators and their transition probabilities can be accurately approximated when θ is small. It is then straightforward to integrate the inbreeding model into the original Lander–Green model. We encode the two quaternary indicators using 4 binary bits in the inheritance vector. Consider a pedigree containing i inbreeding processes and n' other members not involved in inbreeding. The inheritance vector v_l at every site l now has length $2\times n'+4\times i$. Each possible realization of v_l is a hidden state in HMM. The transition probability from v_l to v_{l+1} is the product of transition probabilities of all binary indicators and pairs of quaternary indicators. We can then solve the HMM using standard routine:

$$\begin{aligned} P(v_l|G_D) &= \frac{P(G_D|v_l)P(v_l)}{P(G_D)} \\ &= \frac{P(g_1,\ldots,g_l|v_l)P(g_{l+1},\ldots,g_L|v_l)P(v_l)}{P(G_D)} \\ &= \frac{P(g_1,\ldots,g_l,v_l)P(g_{l+1},\ldots,g_L|v_l)}{P(G_D)} \\ &= \frac{\alpha(v_l)\beta(v_l)}{P(G_D)} \end{aligned}$$

where

$$\begin{aligned} \alpha(v_l) &= P(g_1,\ldots,g_l,v_l) \\ \beta(v_l) &= P(g_{l+1},\ldots,g_L|v_l) \end{aligned}$$

$\alpha(v_l)$ and $\beta(v_l)$ can be solved recursively:

$$\begin{aligned} \alpha(v_{l+1}) &= \sum_{v_l}\alpha(v_l)P(v_{l+1}|v_l)P(g_{l+1}|v_{l+1}) \\ \beta(v_l) &= \sum_{v_{l+1}}\beta(v_{l+1})P(v_{l+1}|v_l)P(g_{l+1}|v_{l+1}) \end{aligned}$$

$P(G_D)$ is obtained from the calculated $\alpha(v_l)$ and $\beta(v_l)$ at any site l:

$$P(G_D)=\sum_{v_l}\alpha(v_l)\beta(v_l)$$

The genome ancestry at site l is, for every founder pair (F_A,F_B),

$$P(\text{Ancestry}(g_l)=(F_A,F_B)) = \sum_{v_l}P(v_l|G_D)$$

for all v_l s.t. g_l is inherited from (F_A,F_B)

Note that, if we place the bits of quaternary indicators at the end of inheritance vector, the recursive calculation of α and β can still greatly benefit from the Elston–Idury algorithm (Idury and Elston, 1997).

4 MODELING THE CC

The CC is a large panel of reproducible, recombinant-inbred mouse lines proposed by the Complex Trait Consortium (Churchill *et al.*, 2002). Over a thousand of mouse lines have been started among which several hundred lines are kept inbreeding. All mouse lines are generated using eight genetically diverse founders via a common breeding scheme designed to randomize the genomic contribution of each founder. It provides an ideal platform for testing our approach.

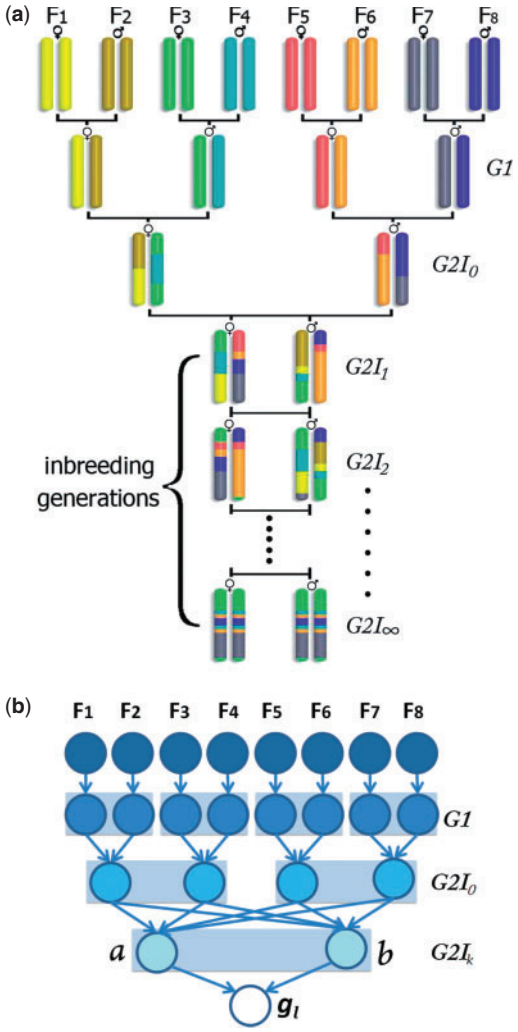

Fig. 3. (**a**) CC breeding scheme: an example derivation of chromosomes by recombining chromosomes from eight ordered founders. $G1$ and $G2I_0$ are two generations of crosses. $G2I_1$ to $G2I_\infty$ are multiple generations of inbreeding. (**b**) The inheritance indicators used to represent the inheritance flow at a SNP site.

4.1 The breeding scheme

CC mice are derived from eight fully inbred founders using the eight-way funnel breeding scheme shown in Figure 3a. The chromosomes of the eight founders (shown in different colors) are combined by two generations of crosses (labeled $G1$ and $G2I_0$), followed by at least 20 inbreeding generations ($G2I_1$ to $G2I_\infty$).

The positions of the eight founders are not fixed. Permutations of the founders are used to randomize the genomes and balance the founder contributions to the resulting CC lines. This variation in initial positions imposes different ancestry constraints on each line. Without loss of generality, we assume a founder order of $F_1F_2F_3F_4F_5F_6F_7F_8$ as shown in Figure 3a.

4.2 Modeling the genome of $G2I_k$ generation

In a CC pedigree, any recombination in the formation of $G1$ haplotypes can be virtually ignored since all founders are fully

inbred. Hence, at each SNP site, we only need four inheritance indicators for $G2I_0$ haplotypes and two quaternary indicators for the two haplotypes in a resulting $G2I_k$ descendant. The structure of the inheritance indicators is shown in Figure 3b.

$G2I_1$ mice are an exception which only involve one generation of inbreeding. For a $G2I_1$ mouse, we simply let the two quaternary indicators revert back to binary indicators. This becomes a standard Lander–Green model and it can be seen that the two $G2I_1$ haplotypes are restricted to be from the left and right half of the funnel, respectively.

5 EXPERIMENTS

In this section, we evaluate the proposed model on both simulated data and real CC genotype data. We implement our model GAIN (Genome Ancestry with INbreeding) for CC using C++. GAIN is compared with MERLIN (Abecasis *et al*., 2002) and HAPPY (Mott *et al*., 2000). MERLIN is a widely used pedigree analysis software based on Lander–Green algorithm and can handle large number of markers. HAPPY is a QTL mapping tool suite and can analyze genome ancestry based on only founder and descendant genotype data, i.e it ignores pedigree structure. Both softwares estimate the genome ancestry directly or indirectly.

5.1 Experiments on simulated data

As ground truth is generally unavailable for real data, we evaluate the accuracy of genome ancestry analysis using simulated data. We simulate the genotype of a $G2I_k$ mouse by recombining real CC founder haplotypes according to the CC pedigree structure. Given the founder genotypes, the founder haplotypes can be obtained trivially since all founders are fully inbred. At each generation we choose recombination position randomly. To simulate genotyping errors, we also introduce random errors to the resulting genotype sequence. When a site is selected to represent an error, we flip its value to heterozygous if it is homozygous originally. If a heterozygous site is selected, we change it to one of the homozygous state randomly. This resembles the fact that most genotyping errors are between heterozygous and homozygous states, instead of between the two homozygous states.

We simulate 20 test cases for each generation from $G2I_1$ to $G2I_{20}$. The number of markers ranges from 6000 to 10 000. As MERLIN does not output probability distribution for each inheritance vector, we first compare the best founder ancestry pair estimated by each method against the true answer. The error rate is measured by the percentage of sites where the estimated best founder ancestry does not match the ground truth. Figure 4 shows the error rate of all three methods in the simulated data with and without errors. Results of MERLIN are only available for the first four generations as the running time grows exponentially with the size of pedigree. No results can be generated within reasonable running time (3 h) for generations beyond $G2I_4$. By incorporating pedigree information, both GAIN and MERLIN infer accurate estimates (error rate <2%). In contrast, HAPPY has much higher error rates and is more sensitive to noise.

As mentioned previously, an accurate solution to the genome ancestry problem is important to subsequent studies such as QTL analysis. In such studies, not only the most likely genome ancestry is desired, but also the probabilities of each founder pair are wanted.

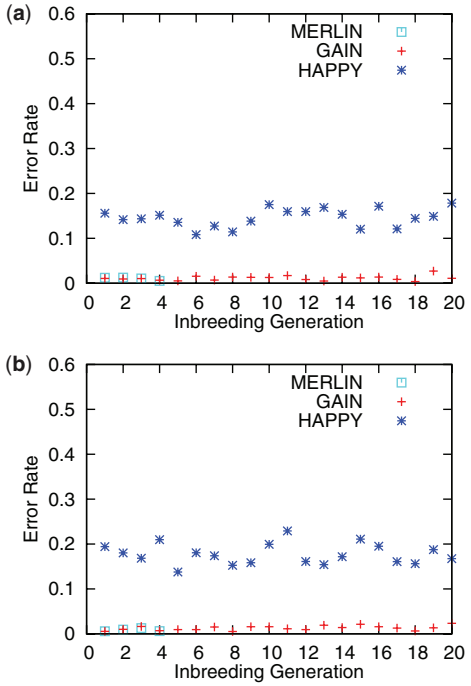

Fig. 4. (**a**) Comparison of error rates of GAIN, MERLIN and HAPPY on a simulated dataset with no noise. (**b**) Comparison on a simulated dataset with 1% noise.

Hence, it is also important to evaluate the probability distribution generated by each method. Both GAIN and HAPPY compute a probability distribution of each founder pair being the ancestry at a SNP site. We investigate the proportion of probabilities assigned to wrong founder ancestry. The result in Figure 5 shows that the knowledge of pedigree structure is indispensable in solving the genome ancestry problem. While HAPPY infers the most probable ancestry correctly for >80% of the markers, it assigns near 60% of the total probabilities to wrong ancestry choices. The misassigned probabilities could hamper further studies. With pedigree structure modeled, GAIN can resolve most ambiguities and assigns only <4% of the total probabilities to wrong ancestry.

5.2 Experiments on real CC data

Our dataset consists of genotypes of all autosomes from 96 mice of generation $G2I_5$ to $G2I_{12}$. The number of SNP markers on each chromosome ranges from 4122 to 35 172. Due to the running time constraint of MERLIN, we only compare GAIN with HAPPY which does not consider pedigree structure. Since the true genome ancestry is unknown, we investigate the difference between the results of the two approaches.

We compare both the best ancestry estimated and the full probability distribution of each possible ancestry. The first comparison (Fig. 6a) shows the percentage of sites of which the best ancestry estimated by the two methods do not agree. The difference in best ancestry choice is very similar to that of our experiments on simulated data with random error: the results from the two methods differ by 20%. We further measure the difference in probability distributions quantitatively using Jensen-Shannon divergence (JSD, Lin, 1991) which is a smoothed and bounded divergence based

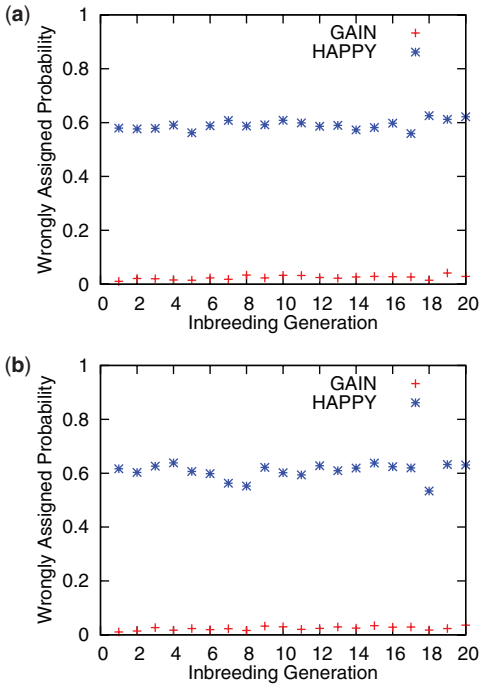

Fig. 5. (**a**) Proportion of probabilities assigned to wrong ancestry by GAIN and HAPPY on a simulated dataset with no noise. (**b**) Proportion of probabilities assigned to wrong ancestry by GAIN and HAPPY on a simulated dataset with 1% noise.

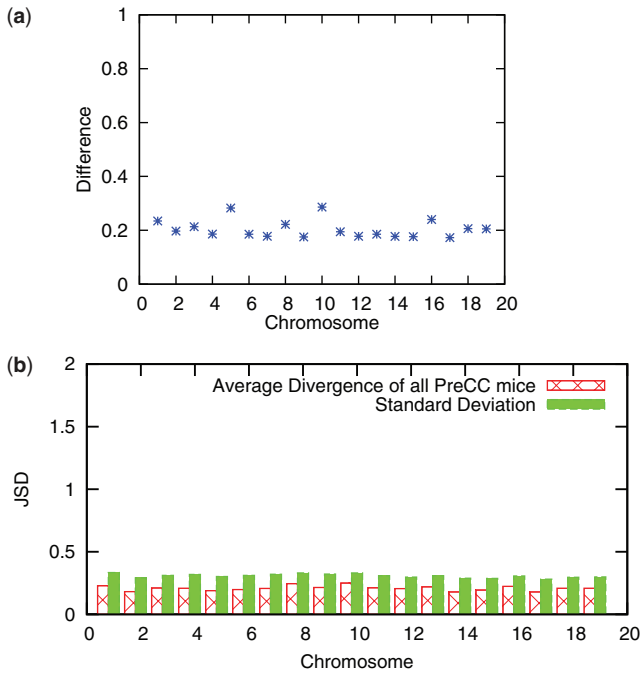

Fig. 6. (**a**) The difference in best ancestry estimated by GAIN and HAPPY. (**b**) The average JSD between results from GAIN and HAPPY on chromosomes 1 to 19 of 96 real CC mice.

on Kullback–Leibler divergence. The JSD between two probability distributions p_1 and p_2 is defined as:

$$\mathrm{JSD}(p_1||p_2) = \sum_i p_1(i)\log_2 \frac{p_1(i)}{\frac{1}{2}p_1(i)+\frac{1}{2}p_2(i)} + \sum_i p_2(i)\log_2 \frac{p_2(i)}{\frac{1}{2}p_1(i)+\frac{1}{2}p_2(i)}$$

A low JSD indicates high similarity between $p1$ and $p2$. The JSD ranges between 0 and 2. Figure 6b compares the mean and SD of the JSD between HAPPY's results and ours over all markers and all 96 mice, grouped by chromosomes.

Though we cannot compare the results against the ground truth for real CC data, the source of difference are further investigated. Consider again the CC pedigree in Figure 3a. The initial four founder-mating pairs (F_1,F_2),(F_3,F_4), (F_5,F_6), (F_7,F_8) cannot serve as ancestry for any genotypes of $G2I_k$ descendants. This is because any genetic material passed from a founder mating pair is carried by a single haplotype in the $G2I_0$ generation. These four founder pairs are thus invalid ancestry choices if the pedigree structure is considered. As an example to show the improved inference due to incorporating pedigree knowledge, the ancestry of chromosome 7 of a $G2I_6$ mouse inferred by GAIN and HAPPY are shown in Figure 7a and b respectively. The most probable founder pair inferred by HAPPY agrees with our result at most sites. But their actual probabilities are often different. To quantify the extent to which HAPPY assigns positive probabilities to invalid ancestry, at each site l, we aggregate the probabilities of invalid ancestry and plot this 'pedigree inconsistency' measure in Figure 7c. We can see that, the difference between Figure 7a and b is largely influenced by the 'pedigree inconsistency'. Moreover, the probability distributions of ancestry choices at neighboring sites are not independent. Probabilities assigned to pedigree-inconsistent ancestry can substantially influence the, choice of ancestry at neighboring sites. Such 'propagated error' is sometimes the main cause of the JSD between HAPPY's results and ours. As an example, Figure 7d shows a region in chromosome 1 from another $G2I_6$ mouse where the propagated error is the main cause of divergence. In this region, HAPPY does not assign significant probabilities to invalid ancestry choice, except for a few sites at both ends of this region. But, in the middle part, HAPPY favors ancestry choices that are one recombination away from these invalid ancestry choices.

To sum up, even partial pedigree knowledge causes a big difference in analyzing genome ancestry. Though HAPPY can conduct analysis rapidly, its results on complex pedigrees can be biased. On the other hand, our method can provide a pedigree consistent inference in comparable running time.

5.3 Running time performance

For a pedigree containing i inbreeding processes and n' members not involved in inbreeding, the time complexity of GAIN is $O(L\times n'\times 2^{2n'}\times 2^{8i})$ where L is the number of SNP markers. For any $G2I_k$ animal in CC pedigree, the time complexity remains the same. The running time does not depend on the error rate of genotype data either. Figure 8 shows the running time comparison of GAIN, MERLIN and HAPPY.

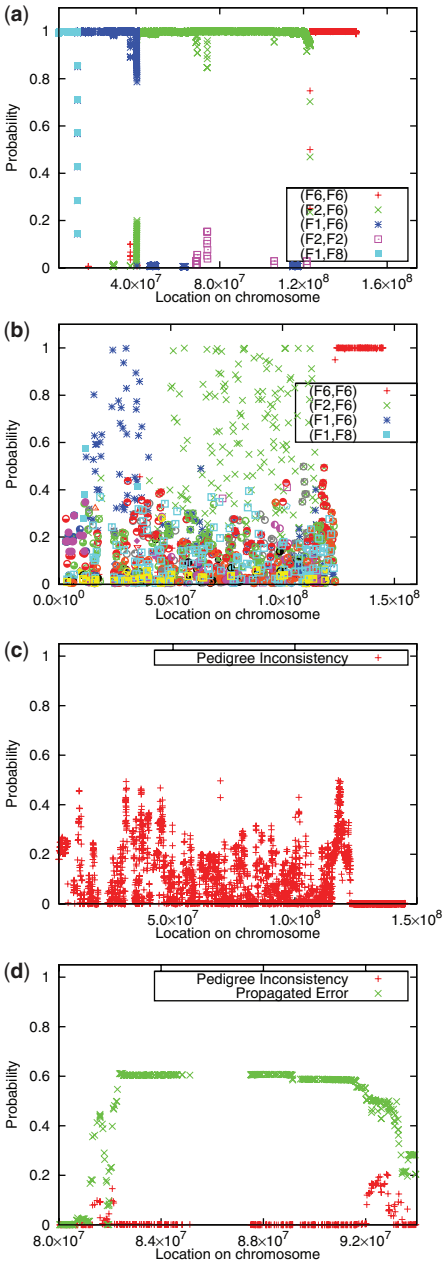

Fig. 7. (**a**) Ancestry inference on chromosome 7 of a $G2I_6$ mouse by GAIN. (**b**) Ancestry inference on chromosome 7 of the same mouse by HAPPY. (**c**) The pedigree inconsistency in (b), i.e. the aggregated probability assigned to ancestry that violates pedigree knowledge. (**d**) A region in chromosome 1 from another $G2I_6$ mouse where propagated error is the main cause of divergence.

6 DISCUSSION

The development of high-density SNP technology makes model animal resources a powerful tool for studying genetic variations. It also makes any analysis on such resources computationally challenging. In this article, we demonstrate that modeling repetitive substructure of a pedigree can provide significant improvement in efficiency without compromising accuracy. We introduce a novel method for modeling the inbreeding process. Integrated into the HMM framework originally introduced by the Lander–Green algorithm, our method can handle large pedigrees such as CC efficiently. The inbreeding substructure model alone does not speed up the ancestry inference for all types of pedigrees, but, as we have shown with the CC, the computational benefit can be crucial for analyzing many model animal resources. In analyzing such data, our method outperforms previous methods in terms of accuracy and efficiency. We believe that substructure modeling is a promising approach for large pedigree analysis, especially when specific types of pedigree are of interest. In the future, we plan to investigate other common substructures and build a more general framework to allow efficient computation on more types of pedigrees.

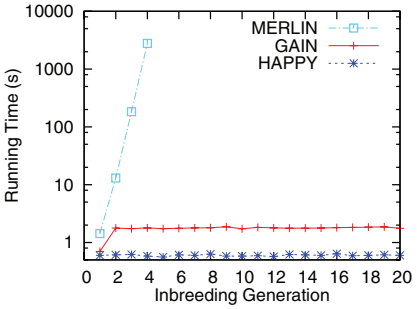

Fig. 8. Average running time of the three methods on dataset containing 6644 markers. The experiment is conducted on an Intel desktop with 2.66 Ghz CPU and 8 GB memory.

Funding: National Science Foundation (IIS0448392, IIS0812464, partially); National Institutes of Health (GM076468, partially)

Conflict of Interest: none declared.

REFERENCES

Abecasis,G.R. *et al.* (2002) MERLIN-rapid analysis of dense genetic maps using sparse gene flow trees. *Nat. Genet.*, **30**, 97–101.

Browning,S., and Browning,B.L. (2002) On Reducing the Statespace of Hidden Markov Models for the Identity by Descent Process. *Theor. Popul. Biol.*, **62**, 1–8.

Chia,R. *et al.* (2005) The origins and uses of mouse outbred stocks. *Nat. Genet.*, **37**, 1181–1186.

Churchill,G.A. *et al.* (2002) The Collaborative Cross, a community resource for the genetic analysis of complex traits. *Nat. Genet.*, **36**, 1133–1137.

Donnelly,K.P. (1983) The probability that related individuals share some section of genome identical by descent. *Theor. Popul. Biol.*, **23**, 34–63.

Gabriel,S.B. *et al.* (2002) The structure of haplotype blocks in the human genome. *Science*, **296**, 2225–2229.

Geiger,D. *et al.* (2009) Speeding up HMM algorithms for genetic linkage analysis via chain reductions of the state space. *Bioinformatics*, **25**, 196–203.

Gudbjartsson,D.F. *et al.* (2005) Allegro version 2. *Nat. Genet.*, **37**, 1015–1016.

Idury,R.M., and Elston,R.C. (1997) A faster and more general hidden Markov model algorithm for multipoint likelihood calculations. *Hum. Hered.*, **47**, 197–202.

Jensen,C.S. and Kong,A. (1999) Blocking Gibbs sampling for linkage analysis in large pedigrees with many loops. *Am. J. Hum. Genet.*, **65**, 885–901.

Kruglyak,L. *et al.* (1996) Parametric and nonparametric linkage analysis: a unified multipoint approach. *Am. J. Hum. Genet.*, **58**, 1347–1363.

Lander,E.S. and Green,P. (1987) Construction of multilocus genetic linkage maps in humans. *Proc. Natl Acad. Sci. USA*, **84**, 2363–2367.

Li,J. and Jiang,T. (2005) Computing the minimum recombinant haplotype configuration from incomplete genotype data on a pedigree by integer linear programming. *J. Comput. Biol.*, **12**, 719–739.

Lin,J. (1991) Divergence measures based on the Shannon entropy. *IEEE Trans. Inform. Theory*, **37**, 145–151.

McPeek,M.S. (2002) Inference on pedigree structure from genome screen data. *Stat. Sin.*, **12**, 311–335.

Mott,R. *et al*. (2000) A new method for fine-mapping quantitative trait loci in outbred animal stocks. *Proc. Natl Acad. Sci. USA*, **97**, 12649–12654.

Pasaniuc,B. *et al.* (2009) Inference of locus-specific ancestry in closely related populations. *Bioinformatics*, **25**, 213–221.

Piccolboni,A. and Gusfield,D. (2003) On the complexity of fundamental computational problems in pedigree analysis. *J. Comput. Biol.*, **10**, 763–773.

Qian,D., and Beckmann,L. (2002) Minimum-recombinant haplotyping in pedigrees. *Am. J. Hum. Genet.*, **70**, 1434–1445.

Sankararaman,S. *et al.* (2008) Estimating local ancestry in admixed populations. *Am. J. Hum. Genet.*, **8**, 290–303.

Schwartz,R. *et al*. (2004) Inferring piecewise ancestral history from haploid sequences. *Lect. Notes Bioinform.*, **2983**, 62–73.

Sobel,E., and Lange,K. (1996) Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics. *Am. J. Hum. Genet.*, **58**, 1323–1337.

Sundquist,A. *et al.* (2008) Effect of genetic divergence in identifying ancestral origin using HAPAA. *Genome Res.*, **18**, 676–682.

Tang,H. *et al.* (2006) Reconstructing genetic ancestry blocks in admixed individuals. *Am. J. Hum. Genet.*, **79**, 1–12.

Valdar,W. *et al.* (2006) Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nat. Genet.*, **38**, 879–887.

Wright,S. (1922) Coefficients of inbreeding and relationship. *Am. Nat.*, **56**, 330–338.

Wu,Y., and Gusfield,D. (2007) Improved algorithms for inferring the minimum mosaic of a set of recombinants. *Lect. Notes Comput. Sci.*, **4580**, 150–161.

Zhang,K. *et al.* (2002) A dynamic programming algorithm for haplotype block partitioning. *Proc. Natl Acad. Sci. USA*, **99**, 7335–7339.

Zhang,Q. *et al.* (2009) Inferring genome-wide mosaic structure. In *Proceedings of PSB*, pp. 150–161.

Zhang,Q. *et al.* (2008) Genotype sequence segmentation: handling constraints and noise. *Lect. Notes Comput. Sci.*, **5251**, 271–283.