Version of Record: https://www.sciencedirect.com/science/article/pii/S0040580915000908
Manuscript_0e85d4fecab1a2c9ce8232898c1506a9

Bayesian Pedigree Inference with Small Numbers of Single Nucleotide Polymorphisms via a Factor-Graph Representation

Eric C. Anderson[a,*], Thomas C. Ng[b]

[a] *Fisheries Ecology Division, Southwest Fisheries Science Center, National Marine Fisheries Service, National Oceanic and Atmospheric Administration, 110 Shaffer Road, Santa Cruz, CA 95060, USA*
[b] *Department of Biomolecular Engineering, University of California, Santa Cruz, CA*

Abstract

We develop a computational framework for addressing pedigree inference problems using small numbers (80–400) of single nucleotide polymorphisms (SNPs). Our approach relaxes the assumptions, which are commonly made, that sampling is complete with respect to the pedigree and that there is no genotyping error. It relies on representing the inferred pedigree as a factor graph and invoking the Sum-Product algorithm to compute and store quantities that allow the joint probability of the data to be rapidly computed under a large class of rearrangements of the pedigree structure. This allows efficient MCMC sampling over the space of pedigrees, and, hence, Bayesian inference of pedigree structure. In this paper we restrict ourselves to inference of pedigrees without loops using SNPs assumed to be unlinked. We present the methodology in general for multigenerational inference, and we illustrate the method by applying it to the inference of full sibling groups in a large sample ($n = 1,157$) of Chinook salmon typed at 95 SNPs. The results show that our method provides a better point estimate and estimate of uncertainty than the currently best-available maximum-likelihood sibling reconstruction method. Extensions of this work to more complex scenarios are briefly discussed.

Keywords: multigeneration pedigree inference; Sum-Product algorithm; relationship inference; full-sibling reconstruction

1. Introduction

Genetic markers have been used to infer the relationship or degree of relatedness of individuals for a variety of applications. Early uses relied on blood groups (Landsteiner and Levine, 1928) for simple paternity testing problems, but with the development of new and better forms of genetic markers, relationship and relatedness inference have been used to address a wide range of problems in numerous fields, such as the estimation of quantitative genetic parameters (Garant and Kruuk, 2005; Visscher et al., 2008), gene mapping (Purcell et al., 2007), captive breeding for animal conservation (Ivy et al., 2009), and understanding ecological processes (Blouin, 2003).

In some of these cases, inferred pedigree relationships provide an expected fraction of genome shared indentical by descent (IBD) between individuals, but it has been argued recently that the explosion of genome-wide single nucleotide polymorphism (SNP) data renders IBD-based concepts of relatedness of limited value (Speed and Balding, 2015). While this is an exciting prospect, there remain a number of contexts in which the inference of pedigrees remains central, such as when pedigree reconstruction is used for intergenerational tagging in wildlife management (Steele et al., 2013), or in the study of non-model organisms in ecological fields that have not yet enjoyed the same expansion of genomic-scale data as has medical genetics: fields collectively called "molecular ecology."

Until recently in molecular ecology, the genetic markers of choice for relationship inference were microsatellites (Tautz, 1989), but now SNP markers are being adopted more widely (Seeb et al., 2011). Although some molecular ecology studies employ next generation sequencing to type hundreds of thousands of SNPs across the genome, the number of individuals in such studies is still typically limited, and we predict that for reconstruction of pedigrees with many individuals, panels of about 100 to several hundred SNPs with high minor allele frequencies will become an increasingly common data type in molecular ecology. Fortunately, such data sets can provide considerable power for pedigree reconstruction (Anderson and Garza, 2006); consequently, here we focus on developing methodology useful for data sets with ≈80–400 SNPs.

Most problems of inferring relationships can be viewed as instances of pedigree reconstruction; however, different types of relationship-inference problems have often been treated separately. For example, the field of parentage inference (Meagher and Thompson, 1987; Roeder et al., 1989; Marshall et al., 1998; Nielsen et al., 2001) devel-

*Corresponding author: eric.anderson@noaa.gov

oped separately from the field of sibling-group inference (Painter, 1997; Almudevar and Field, 1999), and the inference of grandparents (Christie et al., 2011; Letcher and King, 2001) has been treated as something of a distinct problem. Ideally, all of these disparately-treated inference problems could be handled naturally as special cases of a more general multigeneration pedigree inference approach. The program COLONY (Wang, 2004; Wang and Santure, 2009) achieves this to a degree, jointly inferring sibling groups and parents, but its application is limited to, at most, two sampled generations.

By and large, however, the multigeneration pedigree reconstruction methods in use today (Almudevar, 2003; Cowell, 2009; Almudevar and LaCombe, 2012; Riester et al., 2009; Sheehan et al., 2014) rely on two assumptions that do not allow them to encompass the probability models used in, for example, sibling inference, and that make it difficult to apply them in molecular ecology scenarios. The first of these assumptions is that the sample is *complete* (see Almudevar 2003) with respect to the pedigree, which means that any observed founders of an inferred pedigree are assumed to be unrelated. This assumption arises because these methods tend not to model the genotypes of unsampled or latent individuals in the pedigree, and, consequently, do not provide a means for inferring pedigree links through unsampled individuals. The second assumption is that there is no genotyping error, or that it is negligible (which is rarely true in molecular ecology). The latter appears to be the case in Almudevar (2003), Cowell (2009), Almudevar and LaCombe (2012), and Sheehan et al. (2014). Riester et al. (2009), by contrast, include a correction in the genetic transmission probability from parent(s) to offspring to allow for genotyping error, but the correction is applied independently for every transmission in a way that does not allow the observed genotypes of an individual's offspring and parents to be jointly informative of the true state of the individual's imperfectly-observed genotype.

One way to relax each of these assumptions would be to explicitly include the true genotypes of sampled individuals (whose genotypes have been collected with some genotyping error) as latent variables in the model, and also to allow unsampled individuals (with fully unobserved genotypes) to be modeled and to occur within the inferred pedigrees of sampled individuals. Such an approach has been applied on the scale of one to two generations in at least two different software programs. COLONY explicitly considers the genotypes of unsampled individuals, and then marginalizes them out of the model by summing over all their possible genotypic states. This marginalization step has been highly optimized in COLONY, but it is still computationally challenging and can lead to long run times on data sets with many individuals. Accordingly, an approximate version of COLONY that does not carry out the full marginalization is offered (Wang, 2012). Likewise, the parentage software MASTERBAYES (Hadfield et al., 2006), includes an option for modeling the latent genotypes of sampled individuals, and sampling over them, via single-site Gibbs updating, in the course of doing Markov chain Monte Carlo (MCMC) over the space of pedigrees; though, apparently, this can slow mixing of the Markov chain, and the current implementation would likely incur a prohibitive computational cost if extended to inference of multigeneration pedigrees (J. Hadfield, pers. comm.).

Here we propose a computational framework for multigeneration pedigree reconstruction with moderate numbers of SNP markers that allows us to relax the assumption of complete sampling and the assumption of no genotyping error in a computationally efficient manner. Our approach is Bayesian and proceeds by MCMC sampling of pedigree structures from their posterior distribution. The framework allows the inclusion of unsampled members of the pedigrees, thus relaxing the complete-sample assumption and allowing the inference of pedigrees that may consist of multiple subpedigrees connected only through unsampled individuals. Computational efficiency is achieved by marginalizing over the latent genotypes of each individual using the Sum-Product algorithm for factor graphs (Kschischang et al., 2001). In this regard, the current restriction to SNPs, assumed to be unlinked, is key: because each SNP has only two alleles the number of genotypic states that must be summed over remains manageable. The Sum-Product algorithm utilizes storage of intermediate values that permit rapid calculation of the joint probability of all the genetic data conditional on a new pedigree configuration. This allows rapid calculation of acceptance probabilities for proposed changes to the current pedigree configuration, providing a way to perform MCMC over the space of pedigrees.

We introduce the idea of using the Sum-Product algorithm to make efficient updates to pedigrees in MCMC, and show how the formulation makes it possible to relax the complete-pedigree assumption and the assumption of no genotyping error. We start with a description of the probability model and its assumptions, the representation of pedigrees as factor graphs, and the Sum-Product algorithm. We then describe MCMC over the space of pedigrees and demontrate how unsampled individuals can be included in the MCMC sampling as graphical appendages we call *prongs*. In this paper we address inference of pedigrees without loops, and, while we develop the theory in the context of multigenerational pedigree construction, our current implementation is restricted to the problem of full-sibling reconstruction. We use this implementation to demonstrate a Bayesian full-sibling reconstruction method, showing it to be computationally feasible and more accurate than COLONY, while also providing a better estimate of uncertainty. Finally we briefly discuss some of the challenges that remain for a full implementation—most notably, efficient handling of inbreeding loops and marriage chains—and possible extensions of the method.

2. Methods

2.1. Notations and the probability model

The goal of our inference is the pedigree $\mathscr{P}$ connecting the individuals in our sample. A true pedigree that included observed individuals (in the sample) and unobserved individuals (not in our sample), connected through a sufficient number of generations back in the past, could conceivably include every individual in the sample as part of a single, fully-connected pedigree. However, with the sorts of genetic data we are interested in, it will be feasible to infer unsampled ancestors only one or two generations back in time. We let T denote the number of generations beyond which we will not attempt to infer the unsampled ancestors of sampled individuals. With this limitation, it makes sense to regard a pedigree $\mathscr{P}$ as consisting of some number $C \geq 1$ of connected components which are all mutually unconnected. That is $\mathscr{P} = \{\mathscr{P}^{(1)}, \ldots, \mathscr{P}^{(C)}\}$, where the elements of each $\mathscr{P}^{(i)}$ are connected, but $\mathscr{P}^{(i)}$ is unconnected to $\mathscr{P}^{(j)}$ $\forall i \neq j$.

We illustrate further notation referring to a simple example pedigree depicted as a marriage node graph (Thompson et al., 1978) in Figure 1(a). The gray filled nodes indicate that genotypes are observed (with error) on all of the individuals. Let x_i denote the true genotype (as if observed without error) of individual i (in the pedigree, $i \in \{1, \ldots, 10\}$), and let y_i denote the observed genotype of i, subject to genotyping error according to some genotyping error model ϵ. The genotypes of the founders (1, 2, 4, and 5) in this pedigree, if taken as unrelated, are *a priori* independent of one another, but depend on characteristics (for example allele frequencies) θ of the population. Throughout, we will assume that the genetic markers are unlinked.

A pedigree can be represented as an acyclic directed graph (DAG) (Lauritzen and Sheehan, 2003) in which directed edges connect parents to offspring (Figure 1(b)). Given the factorization implied by the DAG, the probability of the complete data, $x = (x_1, \ldots, x_{10})$, and $y = (y_1, \ldots, y_{10})$ can be written as:

$$\begin{aligned} p(x,y) &= p(x_1|\theta)p(x_2|\theta)p(x_4|\theta)p(x_5|\theta) \times \\ &\quad p(x_3|x_1,x_2) \times p(x_6|x_5,x_4)p(x_7|x_5,x_4) \times \\ &\quad p(x_8|x_3,x_4)p(x_9|x_3,x_4)p(x_{10}|x_3,x_4) \times \\ &\quad \prod_{i=1}^{10} p(y_i|x_i,\epsilon), \end{aligned} \tag{1}$$

where, $p(x_c|x_m, x_p)$ denotes the probability that parents with genotypes x_m and x_p would produce a child with genotype x_c.

We will write $p(x^{(k)}, y^{(k)}|\mathscr{P}^{(k)})$ to denote the joint probability of the complete data associated with a pedigree $\mathscr{P}^{(k)}$. Dependence on θ and ϵ is implied, though they are omitted from the notation. The likelihood of a pedigree $\mathscr{P}^{(k)}$ is the probability of the observed data on $\mathscr{P}^{(k)}$ found by summing over all the latent states, $x^{(k)}$:

$$L(\mathscr{P}^{(k)}) = p(y^{(k)}|\mathscr{P}^{(k)}) = \sum_{x^{(k)}} p(x^{(k)}, y^{(k)}|\mathscr{P}^{(k)}),$$

and the likelihood of a collection of unconnected pedigrees, $\mathscr{P} = \{\mathscr{P}^{(1)}, \ldots, \mathscr{P}^{(C)}\}$ is

$$L(\mathscr{P}) = p(y|\mathscr{P}) = \prod_{k=1}^{C} p(y^{(k)}|\mathscr{P}^{(k)}).$$

Given a prior distribution $p(\mathscr{P})$, the posterior distribution of a pedigree configuration $\mathscr{P}$ is given by $p(\mathscr{P}|y) \propto p(\mathscr{P})p(y|\mathscr{P})$ which cannot be computed exactly because the unknown normalizing constant involves an intractable sum over the space of all pedigrees with non-zero likelihood and prior. We note that several prior distributions over the space of possible pedigrees have been proposed, both in the multigenerational context (Sheehan and Egeland, 2007; Almudevar and LaCombe, 2012), as well as on the scale of a single generation (Koch et al., 2008; Wang and Santure, 2009), but we will not choose an explicit prior until a later example.

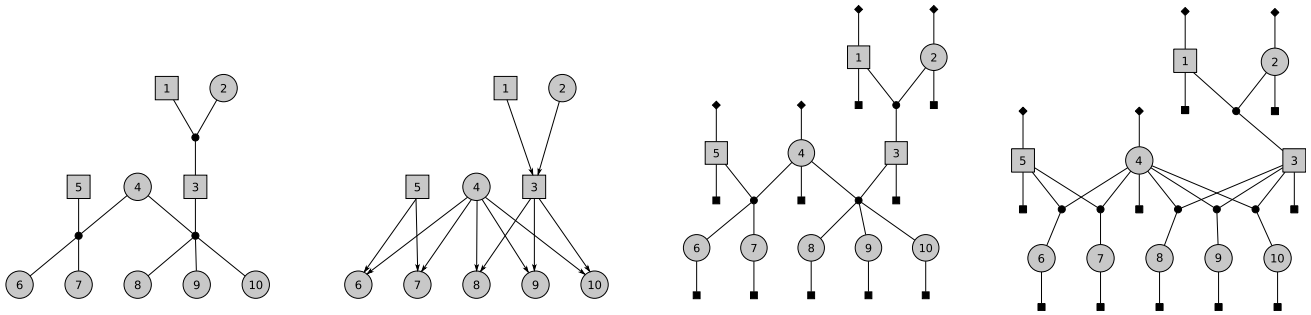
$p(\mathscr{P}|y)$ can be approximated via MCMC (Gelman et al., 2004) by devising a Markov chain that delivers a correlated sample of pedigrees from the posterior distribution. This can be done by a standard Metropolis-Hastings algorithm (Hastings, 1970): if $\mathscr{P}$ represents the current pedigree configuration in the chain, a new configuration $\mathscr{P}'$ is drawn from a proposal distribution $q(\mathscr{P}'|\mathscr{P})$ and that proposal is accepted with probability equal to the minimum of 1 or the Hastings ratio:

$$\frac{q(\mathscr{P}|\mathscr{P}')}{q(\mathscr{P}'|\mathscr{P})} \frac{p(\mathscr{P}')p(y|\mathscr{P}')}{p(\mathscr{P})p(y|\mathscr{P})}.$$

In some contexts, $q(\mathscr{P}'|\mathscr{P})$ can be chosen to be proportional to $p(\mathscr{P}')p(y|\mathscr{P}')$, possibly improving mixing of the chain. The key to doing this MCMC efficiently is being able to compute $p(y|\mathscr{P}')$ in the Hastings ratio quickly. Most of the subsequent paper is devoted to achieving a fast way of computing $p(y|\mathscr{P}')$.

2.2. Factor graph representation of data on a pedigree

The form of a factor graph depicts the way in which the joint probability of the complete data factorizes. A factor graph is a bipartite graph with two classes of vertices—"variable" nodes and "factor" nodes—and edges arranged such that variable nodes are adjacent only to factor nodes and factor nodes are adjacent only to variable nodes. The factorization implied by a factor graph is a product over factor nodes of functions of the states at the variable nodes adjacent to each factor node (Kschischang et al., 2001). Thus, if f_j is a factor node in the graph that is adjacent to variable nodes v_i, v_k and v_ℓ, then this implies there is a term $h_j(x_i, x_k, x_\ell)$ in the joint probability of the complete data. In the case of the *pedigree factor graph* we



(a) Marriage node graph (b) DAG (c) Pedigree factor graph (d) Fully factorized

Figure 1: A simple pedigree represented as (a) a marriage node graph, (b) an acyclic directed graph, (c) a pedigree factor graph, and (d) a "fully-factorized" factor graph, which, though it corresponds strictly to the factorization, leads to a factor-graph that is not singly-connected. Males are represented as squares, females as circles. See text for explanation.

define here, the variable nodes are the true, latent genotypes of the individuals, and we will distinguish between three different kinds of factor nodes: p-nodes which depict the prior on the genotypes of the founders, g-nodes which depict the likelihood of observed genotypes; and m-nodes which are topologically identical to marriage nodes.

Associated with each of these types of nodes is a function which has a counterpart in (1). A p-node adjacent to individual i in the pedigree represents a factor $h^{(p)}(x_i)$ in the joint probability that corresponds to the prior on the genotypes of the founders, thus $h^{(p)}(x_i) = p(x_i|\theta)$. The g-nodes denote factors in the joint probability that are likelihoods of the underlying genotypes given the observed genotypes: $h^{(g)}(x_i) = p(y_i|x_i)$ for a g-node adjacent to individual i. The m-nodes denote factors that are functions of the two parents and all the offspring adjacent to the m-node. For example, in the graph of Figure 1(c) the m-node beneath individuals 3 and 4 denotes a factor $h^{(m)}(x_3, x_4; x_8, x_9, x_{10})$ which is the probability of all the offspring genotypes given the parents' genotypes: $p(x_8, x_9, x_{10}|x_3, x_4) = \prod_{i\in\{8,9,10\}} p(x_i|x_3, x_4)$. These three types of nodes are summarized in Table 1. The pedigree factor graph in Figure 1(c) indicates a joint probablity function $p(x, y)$ that factorizes as follows:

$$\begin{aligned} p(x,y) \quad = \quad & h^{(p)}(x_1)h^{(p)}(x_2)h^{(p)}(x_4)h^{(p)}(x_5) \times \\ & h^{(m)}(x_1, x_2; x_3)h^{(m)}(x_5, x_4; x_6, x_7) \times \\ & h^{(m)}(x_3, x_4; x_8, x_9, x_{10}) \times \\ & \prod_{i=1}^{10} h^{(g)}(x_i) \end{aligned} \tag{2}$$

The functions $h^{(m)}$ associated with the m-nodes merit special explanation. If multiple offspring are adjacent to an m-node, then $h^{(m)}(\cdot, \cdot; \cdots)$ itself factorizes into a product over the offspring; the factor graph associated with it could have many more factor nodes—with separate ones connecting each offspring to its two parents. We don't depict the graph that way for two reasons: 1) it loses its close affiliation with the original marriage node graph, and 2) creating a separate factor node for each offspring creates cycles in the resulting factor graph (Figure 1(d))—in general, even a pedigree that includes no marriage chains or inbreeding loops will not be a polytree (Koller and Friedman, 2009) so long as at least one pair of parents has multiple offspring. While exact probability calculations on cyclic factor graphs can be executed using the junction tree algorithm [see Jensen and Kong (1999) or Lauritzen and Sheehan (2003) for a discussion of the junction tree algorithm applied to pedigrees], for the purposes of organizing intermediate calculations to sample over pedigrees we preserve the graphical structure of the original pedigree by using the pedigree factor graph and exploiting the additional factorization of the $h^{(m)}(\cdot, \cdot; \cdots)$ functions while computing (4).

2.3. The Sum-Product algorithm on factor graphs

The Sum-Product algorithm (Kschischang et al., 2001) provides an efficient way to calculate the marginal probability distribution of each variable node in a factor graph, conditional upon all of the observed data. It does this exactly for singly-connected factor graphs (factor graphs with one and only one path between every pair of nodes), and provides the basis for approximate iterative calculations on factor graphs with loops. In the context of pedigrees, the calculations of the Sum-Product algorithm on a singly-connected pedigree factor graph are closely related to the algorithm of Elston and Stewart (1971) and the peeling algorithm on a "zero-loop" pedigree (Thompson et al., 1978). However, while a run of the peeling algorithm delivers the marginal probability of a single reference individual on the pedigree, the Sum-Product algorithm computes the marginal for every individual on the pedigree in only twice the computational effort as computing it for a single individual (see also Totir et al. 2009).

The Sum-Product algorithm can be visualized as a process of passing messages between nodes of the factor graph. Factor nodes pass messages to variable nodes and variable nodes pass messages to factor nodes. Any message that gets passed into or out of a variable node is a nonnegative function defined on the values that the random variable associated with that variable node can take. In our case,

Table 1: Three types of factor nodes in the pedigree factor graph.

Type	Function	Graph Symbol	Example	Corresponding probability
p-node	$h^{(p)}(\cdot)$	◆	$h^{(p)}(x_1)$	$p(x_1\|\theta)$
g-node	$h^{(g)}(\cdot,\cdot)$	■	$h^{(g)}(y_1,x_1)$	$p(y_1\|x_1)$
m-node	$h^{(m)}(\cdot,\cdot;\cdots)$	●	$h^{(m)}(x_3,x_4;x_8,x_9,x_{10})$	$p(x_8\|x_3,x_4)p(x_9\|x_3,x_4)p(x_{10}\|x_3,x_4)$

these functions can be regarded as either likelihoods or probabilities.

Let $\mu_{v_i\longrightarrow f_j}(x_i)$ denote the message sent from variable node v_i to factor node f_j (which is adjacent to v_i). Since this message originates from v_i it is a function that assigns a nonnegative value to every possible value of the discrete variable x_i, associated with v_i, in its domain, $\mathscr{X}_i$. And let $\mu_{f_j\longrightarrow v_i}(x_i)$ denote the message sent from a factor node f_j to variable node v_i. Since this message is passed to variable node v_i, it is also a nonnegative function of x_i.

Messages sent from variable nodes to factor nodes are the simplest to calculate, being a product of the messages coming into v_i from all the *other* factor nodes adjacent to v_i, *i.e.*,:

$$\mu_{v_i\longrightarrow f_j}(x_i) = \prod_{f_k\in\eta(v_i)\backslash f_j} \mu_{f_k\longrightarrow v_i}(x_i) \tag{3}$$

where $\eta(v_i)\backslash f_j$ denotes all the neighbors of variable node v_i, excluding factor node f_j.

Messages sent from factor nodes to variable nodes are a little more complicated since they involve a sum to marginalize out the other variable nodes adjacent to the factor node. Let C denote the set of indices of the variable nodes adjacent to factor node f_j—that is $C = \{k : v_k \in \eta(f_j)\}$—and let x_C denote the state of all the variables adjacent to f_j. Now, imagine we wish to send a message from f_j to a particular one of its neighbors, v_i. Let the indices of the remaining neighbors of f_j that are not v_i be denoted $C\backslash i$ and the states of those variables be denoted by $x_{C\backslash i}$. The message sent from f_j to v_i is a function of x_i, the state of the variable associated with v_i, obtained by marginalizing the function $h_j(x_{C\backslash i}, x_i)$ (the factor in the joint probability corresponding to factor node f_j) by summing over all possible states of $x_{C\backslash i}$, each one weighted by the product of messages incoming to the factor node f_j from $v_{C\backslash i}$:

$$\mu_{f_j\longrightarrow v_i}(x_i) = \sum_{x_{C\backslash i}\in\mathscr{X}_{C\backslash i}} h_j(x_{C\backslash i}, x_i) \prod_{k\in C\backslash i} \mu_{v_k\longrightarrow f_j}(x_k). \tag{4}$$

In interpreting (4) it is important to recognize that the x_k values for $k \in C\backslash i$ change with every value of $x_{C\backslash i}$ that is being summed over. If factor node f_j has no neighbors other than v_i, (*i.e.*, $C\backslash i = \emptyset$) then the message is merely the value of the function associated with f_j, that is:

$$\mu_{f_j\longrightarrow v_i}(x_i) = h_j(x_i). \tag{5}$$

From the foregoing it is clear that sending a message from a factor node f_j to a variable node v_i requires that messages have come to f_j from all of f_j's neighbors except v_i. Likewise, a message cannot be sent from a variable node v_i to factor node f_j until v_i has received messages from all of its neighbors apart from f_j. Thus, any node can send a message to a neighbor if it is not waiting for incoming messages from any other neighbors. This provides a simple rule for running the Sum-Product algorithm: at each step, send messages from any node n_i (we use n_i instead of f_i or v_i to denote that the node could be either a factor node or a variable node) to its neighbor n_{i^*} so long as:

1. a message has not yet been sent from n_i to n_{i^*}
2. node n_i has already received messages from all its neighbors except n_{i^*}, or n_i has no neighbors except n_{i^*}.

When these scheduling rules are applied to a singly-connected factor graph, the process will terminate after a finite number of steps—the sum-product algorithm is $O(n)$ on number of individuals and number of unlinked loci—and two messages (going in different directions) will have been sent along every edge in the graph. The product of all the messages incoming to a variable node v_i, normalized to sum to one over the states of x_i is then the marginal probability of x_i. Furthermore, the summed product of the two messages along any edge of the graph is equal to the joint probability of all the observed data. More specifically, for an edge between any v_i and any of its neighbors, say f_j, on a singly-connected pedigree factor graph, $\mathscr{P}^{(k)}$,

$$p(y|\mathscr{P}^{(k)}) = \sum_{x_i\in\mathscr{X}_i} \mu_{v_i\longrightarrow f_j}(x_i) \times \mu_{f_j\longrightarrow v_i}(x_i). \tag{6}$$

Visual examples of the steps of the Sum-Product algorithm are given in Figures 2 and 3 in which message values were computed using a script in R that had been verified for correctness with the library `libDAI` (Mooij, 2010).

2.4. MCMC over pedigrees

We exploit the fact that (6) holds for every edge in a singly-connected pedigree factor graph to make rapid calculation of $p(y|\mathscr{P}')$ for a new pedigree $\mathscr{P}'$ that can be reached from the current state $\mathscr{P}$ by making certain types of changes that involve breaking and reattaching pedigree

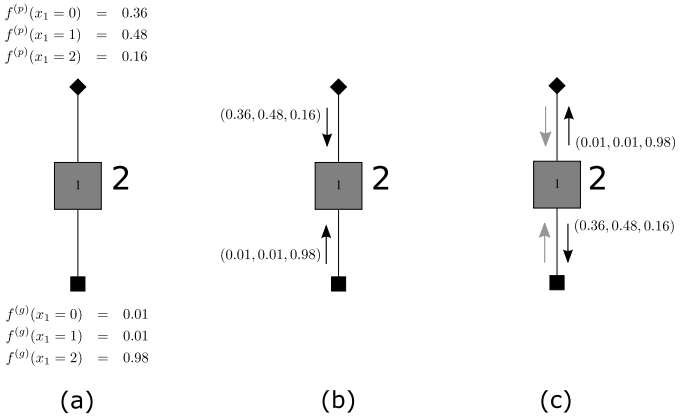


Figure 2: The Sum-Product algorithm on the simplest pedigree factor graph—a single individual. This example assumes a single SNP with two alleles, a and A. The genotype is recorded as 0, 1, or 2, being the number of A alleles carried by the individual. (a) The frequency of the A allele in the population is 0.4, so *a priori* the genotype x_1 of the individual has probabilities 0.36, 0.48, and 0.16, as shown. The observed genotype of the individual is 2, and the genotyping error model is assumed to be simple (2% probability of an error, in which case the observed genotype is equally likely to be either of the two incorrect ones) which gives values for $h^{(p)}$ as shown in the lower part of the panel. (b) The first step of the Sum-Product algorithm. Each of the factor nodes has only one neighbor so they send messages inward toward the single variable node (eq. 5). Each message is a three-vector. (c) Step 2. Once the variable node has received incoming messages, it can send outgoing messages to each factor node adjacent to it (eq. 3). Messages passed in the current step are depicted as black arrows with their values shown. Previously-passed messages are gray arrows. Their values are not shown, but are the same as when they were originally passed. It is easy to verify that the normalized product of all arrows incoming to the individual gives the posterior probability of his genotype.

edges or switching the labeling of individuals. Rapid calculation of $\mathscr{P}$ allows fast acceptance or rejection of proposed moves, since $p(\mathscr{P})p(y|\mathscr{P}')$ appears in the Hastings ratio (Hastings, 1970). In brief, $p(y|\mathscr{P}')$ can be computed by applying (6) over messages that traversed edges that were broken but then were reattached in moving from $\mathscr{P}$ to $\mathscr{P}'$. In this way, calculating the probability of the data given a new pedigree configuration does not require an entire run of a peeling algorithm, but rather, at a single locus, only 3 floating point multiplications and 2 floating point additions.

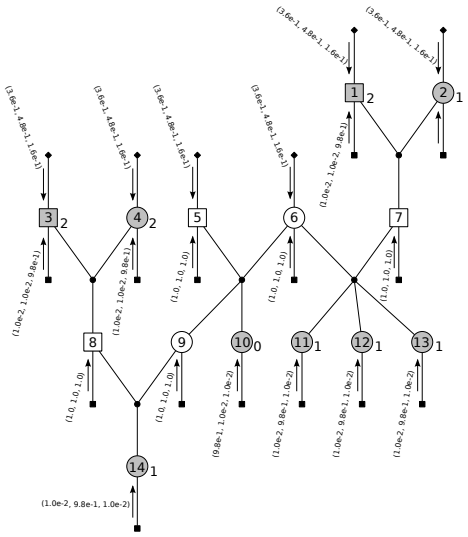
There are numerous types of moves from $\mathscr{P}$ to $\mathscr{P}'$ that might be proposed. In order to describe a simple pair of moves that suffice to make an irreducible MCMC chain over the space of singly-connected pedigree factor graphs, we introduce more terminology (Fig. 5).

Once the Sum-Product algorithm has been performed and we break a pedigree factor graph along one of its edges, two *spikes* are created. Spikes are edges which are broken to become two different edges, each one adhering to one of the two nodes the edge previously connected. The message that was outgoing from the node to which a spike is still attached remains with the spike. A spike attached to a variable node is called a variable-spike and a spike attached to a factor node is called a factor-spike. Fig. 5(b) shows the seven possible types of spikes. Spikes formed by breaking a pedigree can be reattached to spikes resulting from breaking another edge according to rules that follow naturally from what a pedigree factor graph represents. Ascending spikes can only be reattached to downward-facing m-spikes or to p-spikes, descending spikes can only be reattached to upward-facing m-spikes, and likelihood spikes can only be reattached to g-spikes. And, of course, an edge cannot be reattached in such a way that it would create a directed cycle in the DAG implied by the pedigree (Almudevar, 2003).

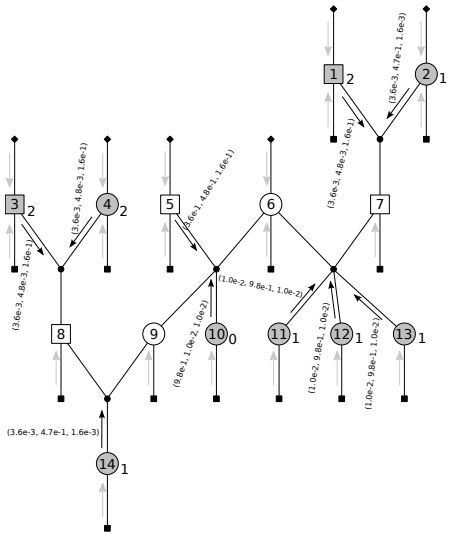
The messages associated with spikes provide the means for quickly computing the probability of pedigrees created by connecting two previously unconnected pedigrees. Let $\mathscr{P}^{(i)}$ and $\mathscr{P}^{(j)}$ be two unconnected pedigree factor trees upon which the Sum-Product algorithm has been run. Allow $\mathscr{P}^{(i)}$ to be broken at an edge so that it creates two unconnected components, $\mathscr{P}_a^{(i)}$ which has a spike a_i where the edge was broken, and $\mathscr{P}_b^{(i)}$ which has a spike b_i. Also, allow $\mathscr{P}^{(j)}$ to be broken into $\mathscr{P}_a^{(j)}$ with spike a_j and $\mathscr{P}_b^{(j)}$ with spike b_j. If a_i and b_j are spikes that can be attached to one another, and $\mathscr{P}_a^{(i)}$ and $\mathscr{P}_b^{(j)}$ are reattached via those spikes, then the joint probability of all the genetic data on the resulting connected component can be calculated by applying (6) to the messages on a_i and b_j. This follows from the fact that the messages passing in opposite directions along an edge in a factor tree are independent of one another, so that (6) can be computed for any reattached spikes without any need to recompute the actual messages flowing along the spikes. We introduce a notation for this. If $\mathscr{P}^{(k)}$ is the connected component formed by attaching $\mathscr{P}_a^{(i)}$ and $\mathscr{P}_b^{(j)}$ along spikes a_i and b_j, and μ_{a_i} and μ_{b_j} are the messages associated with those spikes, which are functions defined over a set of states $\mathscr{X}$, then

$$P(y^{(k)}|\mathscr{P}^{(k)}) = \sum_{x\in\mathscr{X}} \mu_{a_i}(x)\mu_{b_j}(x) = \langle a_i, b_j\rangle. \qquad (7)$$

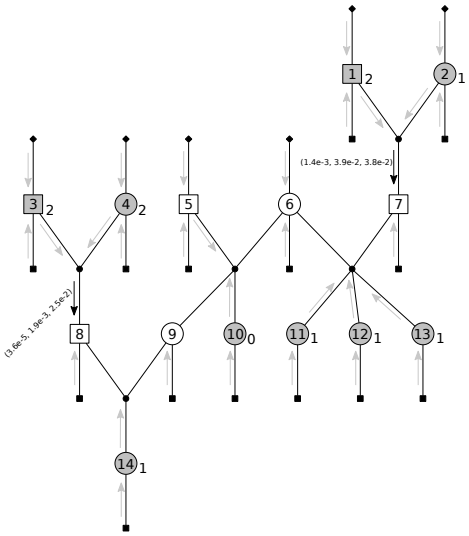
To facilitate moves between pedigree configurations, it is sometimes helpful to add edges (and neighboring nodes) to the pedigree that may themselves be broken and reattached. These additional nodes and edges can be added in such a manner that they do not alter the joint probability of the data on the pedigree. *Stubs* are edges that we may add to variable nodes, and which terminate on the other end at a factor node whose function, $h(x)$, returns 1 for all values of x. As Fig. 5(c) shows, the factor node on a stub may be regarded as a g-node or an m-node (called a g-stub or m-stub respectively). Since the factor associated with a stub returns 1 for all values of x, it should be clear that the addition of a stub does not alter the probability of the data on a pedigree. A *prong* is an additional individual, bearing no genetic data, added to a pedigree—effectively an individual with a g-stub attached to it. It is permissible to attach other stubs to prongs, and, if placed in a founder position on the pedigree, then a prong should have



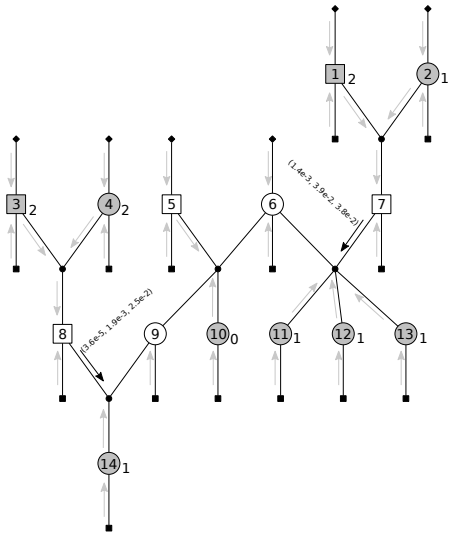
(a) Step 1



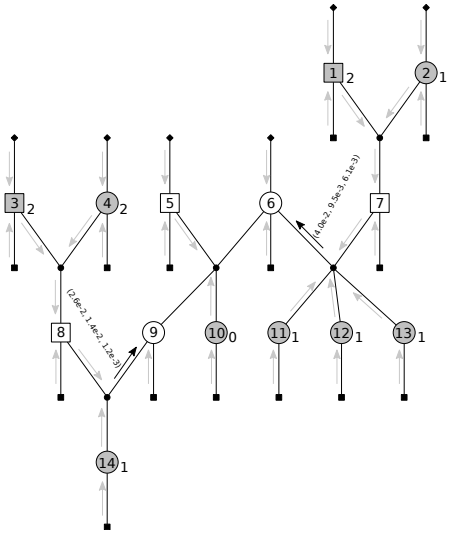
(b) Step 2



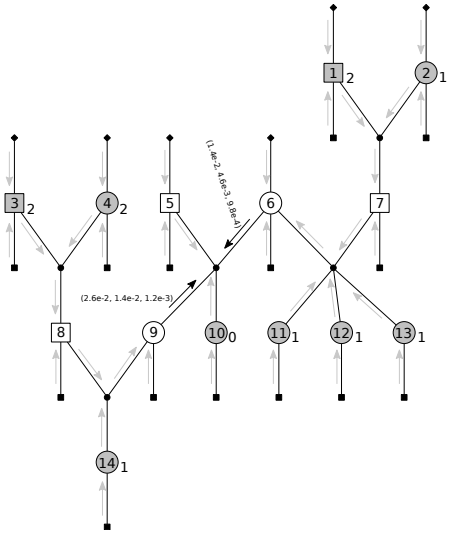
(c) Step 3



(d) Step 4

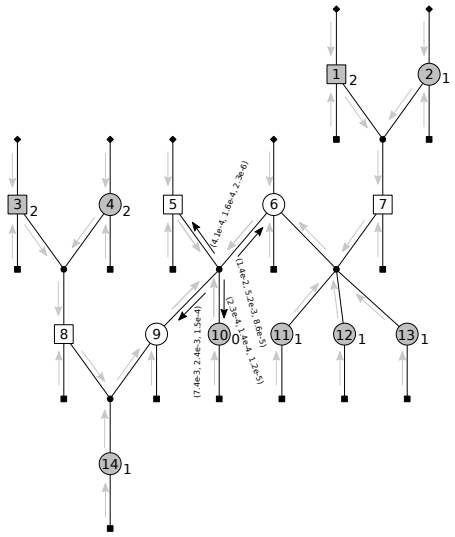


(e) Step 5

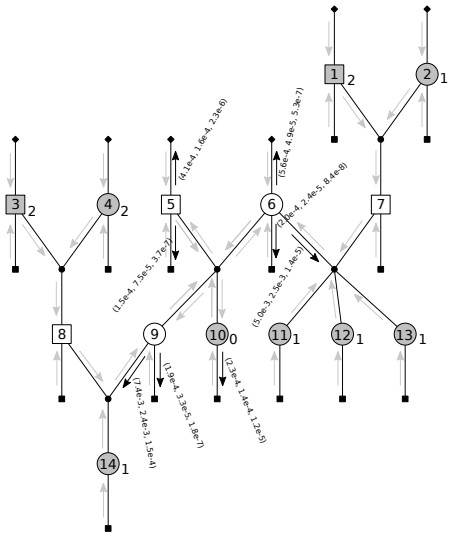


(f) Step 6

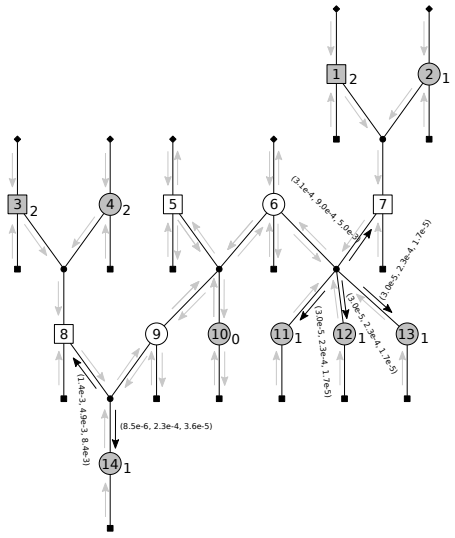
Figure 3: The Sum-Product algorithm for the single SNP from Figure 2 on a 14-member pedigree. over 4 generations. (*a–f*) the first 6 steps of the algorithm. Messages passed during a step are denoted as black arrows and their values are given. Messages passed in previous steps are colored gray. The genotype values of observed individuals appear as numbers to their right.



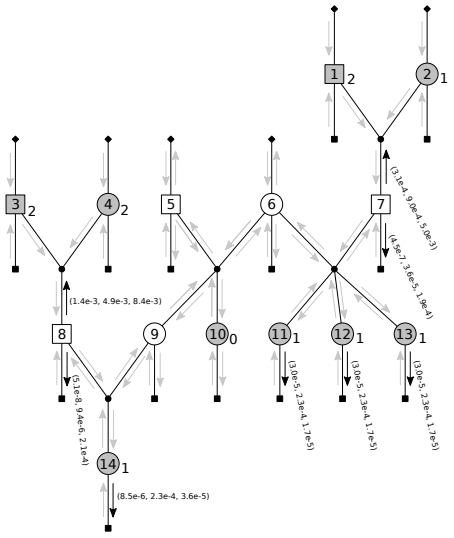
(a) Step 7



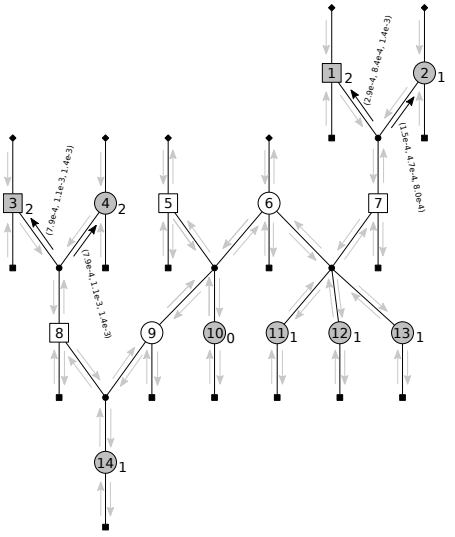
(b) Step 8



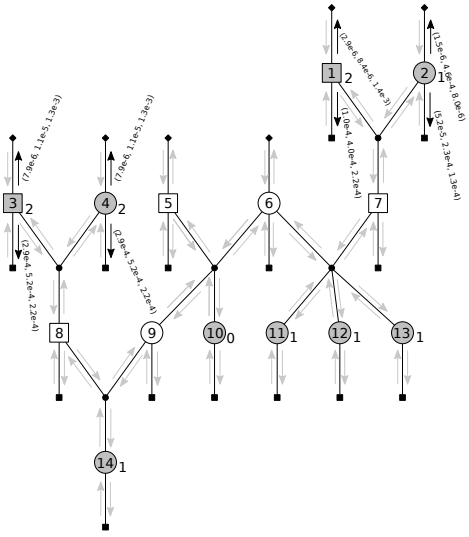
(c) Step 9



(d) Step 10



(e) Step 11



(f) Step 12

Figure 4: Continuation of Figure 3. (*a–f*) steps 7–12 of the algorithm. At its conclusion it can be verified that the dot product of the two messages going in different directions along any edge gives the joint probability of all the observed data.

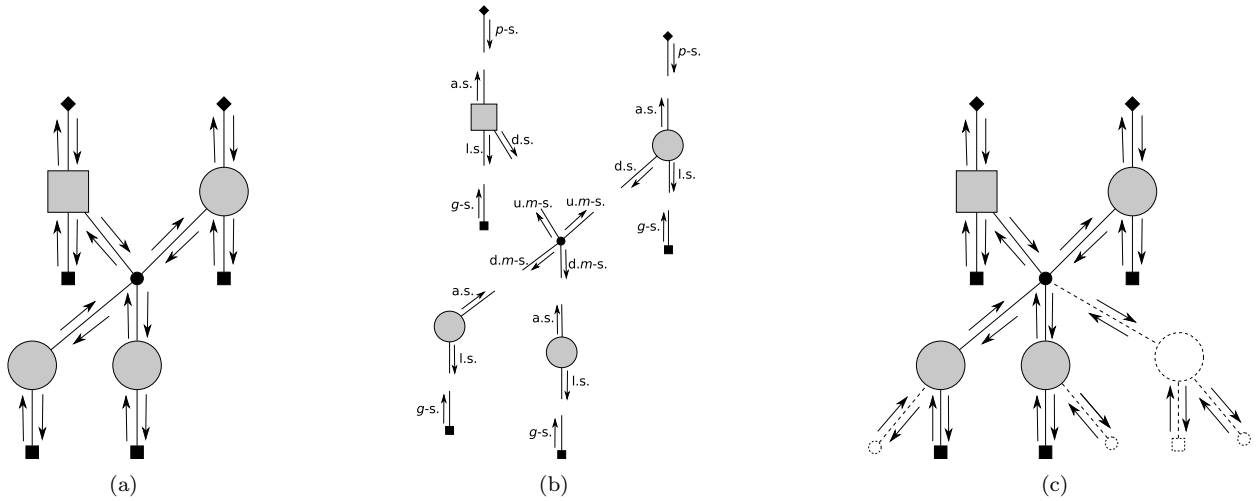


Figure 5: Constructs to help visualize and compute probabilities of proposed changes to pedigrees. (*a*) A simple 4-member pedigree with arrows showing messages passed in the Sum-Product algorithm. (*b*) *Spikes* are created when edges are broken. Outgoing messages from the attached node remain with each spike. There are 7 types of spikes; three "variable spikes" connected to variable nodes: a.s. = ascending spike, d.s. = descending spike, l.s. = likelihood spike; four "factor spikes" connected to factor nodes: p-s. = p-node spike, g-s. = g-node spike, u.m-s. = upward-facing m-node spike, d.m-s. = downward-facing m-node spike. (*c*) *Stubs* are factor nodes with constant functions (*i.e.*, $h(x) = 1\ \forall x$) that may be attached to variable nodes. These are drawn with dashed lines. There are four in the figure: three m-stubs and one g-stub. *Prongs* are unsampled individuals (with no genotype data) that are drawn with dashed lines when added to a pedigree. There is one female prong in the figure with two stubs attached to it. Prongs are useful for creating spikes that allow the inference of pedigree links through unsampled individuals.

as a neighbor a p-node signifying the allele frequency in the population. Prongs play an important role within the context of MCMC— notably they allow for sampling over pedigrees in which some individuals are unobserved, letting the pedigree connecting observed individuals include relatives that have not been observed.

In preparation for the two proposals that will be described in the next section, it must be the case that: every variable node in the pedigree should have one descending stub attached to it (regardless of the number of marriages it is already involved in); every m-node must have two parents, of which none, one, or both may be prongs; every m-node must be attached to at least one offspring node that is a prong with no descendants; and every founder in the pedigree (including prongs that are founders) must be attached to a p-node. These requirement ensure that there are edges available to break so as to propose reconnections that allow the chain to be irreducible. The Sum-Product algorithm should have been run on each such connected component, since the last time it was modified in a proposal.

2.4.1. Pedigree-altering proposals

We are now in position to consider proposals to move a Markov chain over the space of tree-shaped pedigree factor graphs. While many varieties would be possible, we focus on a simple, yet general, pair of proposals.

Simple pedigree split. This most basic of proposals proceeds as follows: 1) choose a connected component $\mathscr{P}^{(i)}$; 2) choose an edge attached to an m-node within $\mathscr{P}^{(i)}$ upon which to break it into two parts, $\mathscr{P}_a^{(i)}$ and $\mathscr{P}_b^{(i)}$; 3) to each resulting spike, a_i and b_i, add a factor node, stub, or prong, as follows: upward- or downward-facing marriage spikes get reattached to prongs, descending spikes get reattached to m-stubs, and ascending spikes get reattached to p-spikes. (Note that there is an unlimited supply of p-spikes, stubs, and prongs to add to a pedigree as needed.) Denote the spike, stub, or prong attached to a_i as a_i^*, likewise for b_i, b_i^*. The joint probability of all the data on $\mathscr{P}_a^{(i)}$ and $\mathscr{P}_b^{(i)}$ is now $\langle a_i, a_i^* \rangle \langle b_i, b_i^* \rangle$, which can be used to compute the Hastings ratio.

Leaf-edge pedigree join. This proposal provides the reverse move to a simple pedigree split. 1) Choose two connected components $\mathscr{P}^{(i)}$ and $\mathscr{P}^{(j)}$. 2) From $\mathscr{P}^{(i)}$ choose an edge to break from amongst all edges which are either adjacent to a p-node or an m-stub, or which are between an m-node and a prong that is, itself, an outermost variable node (that is, it is either a founder in the pedigree, or it has no children in the pedigree). Break this edge, yielding spikes a_i and b_i. One of those spikes (assume it is b_i in this case) will either be a disconnected p-spike, or will be attached to a prong and have a message value of 1 for all x. 3) Choose an edge from within $\mathscr{P}^{(j)}$ in the same manner as one was chosen from $\mathscr{P}^{(i)}$, yielding spikes a_j and b_j, the latter of which is either a disconnected p-spike, or is attached to a prong. 4) If it is not allowable to attach the spike a_i to a_j, reject the move. Otherwise, join a_i to a_j, let the joined pedigree so formed be the proposal and note that the joint probability of the data upon it can be calculated as $\langle a_i, a_j \rangle$.

When one allows that a single individual can be attached to two parent prongs, and that every m-node will be attached to at least one offspring prong that provides an edge which may be broken to add individuals to a sibling group, the above two proposals are sufficient to traverse the space of all possible tree-shaped pedigrees, and, in fact, will always preserve them as trees. It may be that other proposals, such as those that swap branches or individuals between pedigrees, have better mixing properties in some contexts—for example proposals that would switch parents with offspring, in cases in which an individual's generation is unknown. We have been intentionally vague about the process of choosing sub-pedigrees and edges within them, as there are a number of ways in which this might be accomplished. We will demonstrate some on the example problem that follows.

2.4.2. Inferring pedigrees through unsampled individuals

In this framework, the ability to infer pedigrees that include unsampled individuals in the interior of the pedigree is a direct consequence of attaching prongs to individuals in the inferred pedigrees and then including the edges attached to those prongs amongst those that may be broken and reattached when performing pedigree splits and joins. The Sum-Product algorithm, by propagating all the information that is known from the pedigree about the unobserved genotype of the individual represented by the prong, provides a simple and general way of ensuring that evidence for the existence of that individual is evaluated appropriately. There isn't a restriction on how many prongs one might hypothesize or how many generations ancestral to an observed individual one may wish to extend a chain of prongs, but, obviously they should not extend beyond T generations back in time, and their extent should be appropriate to the amount of data (number of markers) available. Figure 6 provides a visual representation of how prongs might be utilized in a problem of grandparentage inference.

3. Application: inference of full-sibling groups

We have implemented a version of the above algorithm restricted to the problem of full sibling inference in the Rcpp-based R (Eddelbuettel and François, 2011; R Core Team, 2014) package FULLSNIPLINGS. Given a sample of individuals, all of the same cohort, FULLSNIPLINGS infers the groups of full siblings under a model in which members of the sample are either unrelated or full siblings; the model does not account for half-sibling or cousin relationships.

3.1. MCMC in FULLSNIPLINGS

Prior to running the MCMC, several values are computed that help restrict the set of proposals that will be considered. Briefly, the likelihood ratio for the hypothesis of full-sibling versus the hypothesis of unrelated is computed for every pair of individuals in the sample. Let $\Lambda_{i,j} = \Lambda_{j,i}$ denote the likelihood ratio for the pair of individuals i and j. 10^5 independent values of the likelihood ratio are then simulated conditional upon the observed allele frequencies and an assumed value of the genotyping error rate. (FULLSNIPLINGS assumes a per-gene-copy error rate of ϵ and the error model described in the appendix of Anderson and Garza (2006).) The value of the 0.001-th quantile, denoted $\Lambda^{(0.001)}$ is calculated, and for each individual i, the set $S_i = \{j : j \neq i,\ \Lambda_{i,j} > \Lambda^{(0.001)}\}$ is compiled and stored. S_i represents the set of individuals that are *not* very likely *non*-siblings of individual i.

The MCMC itself is initialized with a configuration in which every individual in the sample is hypothesized to belong to their own full-sibling group of size one. Each of these sibling groups is represented as the individual attached to three prongs: two parents and a sibling (Fig 7). Changes to the current configuration $\mathscr{P}$ are made by proposing changes to the placement of a focal individual i. The only types of changes allowed are those moving individual i to a new connected component by placing it upon the "sibling prong" of another full-sibling family, or removing it from a family and reinstating it as a sibling group of size 1. This is achieved using a Gibbs sampler. We let Q_i denote the set of connected components in the current configuration $\mathscr{P}$ that contain at least one individual in S_i. Denoting by $\mathscr{P}_j$ the configuration in which individual i is moved to a family j in Q_i, we can compute the probability of proposing each as

$$q(\mathscr{P}_j|\mathscr{P}) \propto p(\mathscr{P}_j)p(y|\mathscr{P}_j).$$

Likewise, the probability of proposing to remain at the current state is proportional to $p(\mathscr{P})p(y|\mathscr{P})$. And, if i is already a sibling of someone in the current state, then one can propose moving i to a family of just a single individual with probability proportional to $p(\mathscr{P}^*)p(y|\mathscr{P}^*)$ where $\mathscr{P}^*$ is that configuration where i has been moved into a family with no siblings. The conditional probability of the data in each of these expressions is quickly calculated by applying (7) for each $\mathscr{P}_j,\ j \in Q_i$, and $\mathscr{P}$ and $\mathscr{P}^*$. Having done so, the proposal distributions can be be normalized to sum to one and a single proposal chosen from those probabilities. It is straightforward to show that the Hastings ratio reduces to 1, always, in this case.

3.2. The prior $p(\mathscr{P})$ in FULLSNIPLINGS

Computing $p(\mathscr{P})p(y|\mathscr{P})$ also requires specifying $p(\mathscr{P})$. Since the full sibling inference problem can be interpreted as a problem of inferring a partition of a sample into full sibling groups, a natural prior would be the multivariate Ewens distribution (Ewens, 1972; Johnson et al., 1997), which is particularly convenient for Gibbs sampling because it provides an easily-calculated predictive distribution, known in genetics as Hoppe's Urn (Hoppe, 1984). We considered the Ewens distribution, but feeling that it would not extend easily to more complex situations (such as the inclusion of both maternal and paternal half-siblings) we chose to experiment in FULLSNIPLINGS with

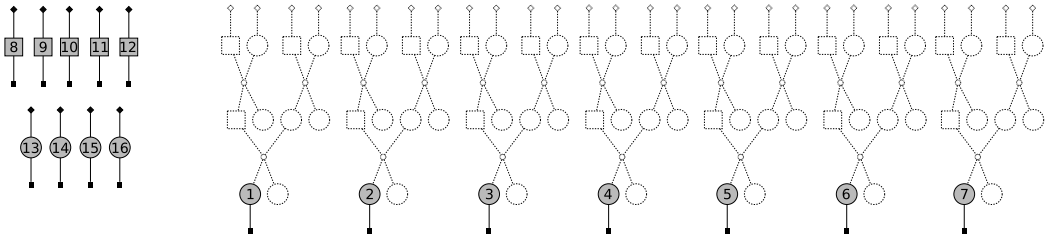


Figure 6: Visual depiction of how prongs can be invoked to handle a problem in grandparentage inference. Imagine that individuals 8–16 are a (possibly incomplete) sample of candidate grandparents of individuals 1–7 (represented here, without loss of generality, as females). The goal is to infer the pedigree connecting individuals 1–7 two generations into the past and to include individuals 8–16 on that pedigree when any of them is a grandparent. Note that 1–7 could be full- or half-siblings or full- or half-cousins of one another, and that should be reflected in the inferred pedigree. The figure shows an arrangement of prongs that would allow the space of these possibilities to be explored through simple pedigree splits and joins.

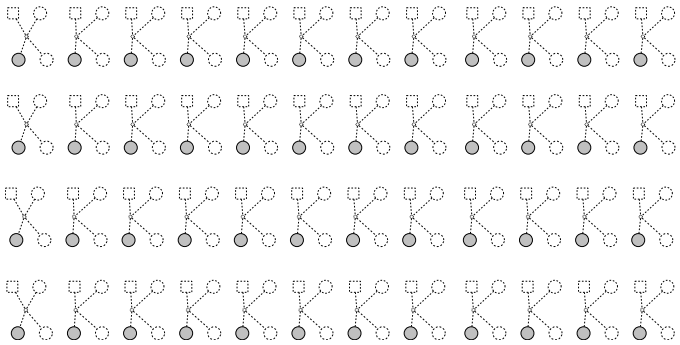


Figure 7: Depiction of the starting configuration for FULLSNIPLINGS. Every observed individual in the sample is represented for simplicity as a female (though they could be of either sex). Each individual has two unsampled-parent-prongs and a sibling prong.

specifying a "pseudo-prior" in terms of an intuitively reasonable predictive distribution, which may, itself, not be derivable as the conditional distribution of an actual joint distribution. Doing so leads our sampler to be a "pseudo-Gibbs" sampler, an approach that has proved beneficial in other genetics problems (Stephens et al., 2001).

We derive the pseudo-prior used for $p(\mathscr{P})$ in terms of the predictive distribution required for Gibbs updates. Imagine that we are going to propose a move of individual i. For any configuration $\mathscr{P}_j$, $\mathscr{P}$, or $\mathscr{P}^*$, we can think of decomposing $p(\mathscr{P})$ into a product of two terms: a prior on the configuration of all the individuals with i removed, $p(\mathscr{P}_{[-i]})$, and the conditional (or predictive) distribution of i's placement, conditional on the configuration of the rest, $p(i|\mathscr{P}_{[-i]})$. Since $p(\mathscr{P}_{[-i]})$ is constant across all possible placements of i into different families, the only part that is relevant for computing the proposal distribution is the predictive distribution $p(i|\mathscr{P}_{[-i]})$, and we compute it thus: the predictive probability that i belongs in its own full sibling group is β, defined to be the proportion of full sibling groups in $\mathscr{P}_{[-i]}$ that include only a single individual. And the predictive probability that i belongs in an existing sibling group j that has n_j individuals (not including i) is $(1-\beta)n_j/n$, where n is the number total number of sampled individuals, excluding i. In practice, since we start in a configuration in which every individual is in a full sibling group of size 1 we define β to be the minimum of 0.995 or the proportion of full sibling groups (excluding i's) of size 1.

3.3. Application to an empirical data set

We applied FULLSNIPLINGS to a a sample of 1157 real Chinook salmon known from parentage inference to be either full-siblings or unrelated to one another at a degree closer than full cousins (*i.e.*, half-siblings were intentionally not present in the sample). The sizes of the full-sibships varied from 12 fish to 1 fish. FULLSNIPLINGS was run for 100 sweeps of burn in (in each sweep, every individual was proposed to be moved exactly once) and 500 sweeps of data collection. This required 13.3 minutes on a single core of a 2.66 GHz Intel Xeon processor in a Mac computer. The 500 samples from the posterior were condensed into a single estimate of a partition by a simple greedy algorithm—sibling groups were ordered from highest to lowest based on the number of occurrences (out of 500 sweeps), and if two sibling groups had the same number of occurrences, the largest was sorted first. Then, proceeding through the sorted list, each newly encountered sibling group that did not include any individuals already appearing in the final condensed list of inferred sibling groups was added to the final condensed list of inferred groups and assigned a posterior probability equal to the number of its occurrences in the MCMC sample, divided by 500.

The same data set was analyzed using COLONY's full likelihood method, which required 21.0 minutes on the same computer. The data were analyzed two different ways, first they were run using COLONY's default "no prior" setting, and second they were run using COLONY's "sibling-prior" option. The "sibling-prior" option in COLONY employs an Ewens distribution, but requires that the user input a value for the mean sibship size (rather than estimating that during the inference procedure). Accordingly, we supplied the program with the mean sibship size from the "no-prior" run.

COLONY returns an estimated "posterior probability" for each inferred sibling group. This is reported as two different values relating to whether individuals were incorrectly included in or excluded from the sibling group. Since COLONY appears to overestimate the degree of certainty in its estimates, we took the lesser of the two values as COLONY's estimate of the posterior probability of each inferred sibling group.

The accuracy of all three methods was assessed using the partition distance (PD) (Painter, 1997; Gusfield, 2002) between the inferred and the true partition (Table 2), and the relationship between inferred sibship size, posterior probability, and the incidence of different kinds of errors was plotted (Fig. 8). On the basis of the PD, FULLSNIPLINGS (PD = 35) performed slightly better than COLONY with the "sibling-prior" (PD = 38), but considerably better than COLONY with the default "no-prior" (PD = 66). The contrast between FULLSNIPLINGS and COLONY is particularly clear in the estimated posterior probabilities from each program. COLONY estimates a posterior probability of 1.0 for many incorrect sibling groups, and the estimated posterior probability does not appear to reflect the actual chance that the sibling group was incorrectly inferred. The posterior probability estimated by FULLSNIPLINGS, however, does a much better job of reflecting uncertainty in sibgroup assignments—few sibgroups with high posterior are incorrectly-inferred, while errors are more common amongst the inferred sibgroups with lower posterior probability.

One well-known behavior of COLONY's maximum likelihood full-sibling reconstruction algorithm is its lack of statistical error control. In short, especially with uninformative genetic data (for example, with few markers) a higher likelihood can often be found by forming spurious sibling groups (Almudevar and Anderson, 2012) than by leaving unrelated individuals separate. To compare the behavior of FULLSNIPLINGS to that of COLONY with progressively less informative data, we ran the Chinook data set using just the first 85, 75, ..., 25, SNP markers. The results are striking (Table 2, Fig. 9). With fewer markers, COLONY infers many incorrectly large full sibling groups and estimates a high posterior probability for most of them. FULLSNIPLINGS behaves very differently—when using fewer SNPs, fewer sibling groups of size ≥ 1 are inferred, and fewer sibling groups are inferred with high posterior probability.

Table 2: Partition distances between inferred sibling groups and the truth with different numbers of loci L. COLONY-sp is COLONY with the "sibling prior" set at a mean of 3.06 (the mean from an initial run at $L = 95$.)

L	FULLSNIPLINGS	COLONY-sp	COLONY
95	35	38	66
85	31	50	80
75	47	75	101
65	70	111	152
55	113	171	208
45	209	262	296
35	374	405	444
25	545	585	618

4. Discussion

We have illustrated a factor-graph representation for genetic data on a pedigree. Using this representation, we have explained how the Sum-Product algorithm on singly-connected pedigree factor graphs computes the joint probability of all the genetic data under a proposed modification to the pedigree. This provides a method for rapidly evaluating proposed changes to the inferred pedigree structure so as to sample pedigrees, using MCMC, according to their posterior probability, which, in turn, provides a computationally efficient method for Bayesian pedigree inference. We have illustrated the method by applying it to the full-sibling reconstruction problem and have shown in application to a real data set that it outperforms the best available maximum likelihood full-sibling reconstruction method, both in terms of the quality of its point estimate, and especially in terms of its assessment of uncertainty.

Although our illustration of the approach on full sibling reconstruction involves pedigrees of only two generations, it shows that a Bayesian approach to pedigree reconstruction may have some advantages over maximum likelihood approaches. Further, the theory underlying the approach extends immediately to multigenerational pedigrees in a way that would relax the assumptions of complete sampling and no genotyping error that have hindered application of existing multigeneration pedigree inference methods in molecular ecology contexts. It also would provide a prinicipled and computationally efficient approach for inferring multigeneration pedigrees that include unobserved individuals.

It should be noted that data other than genetic data might also be profitably modeled in the context of the pedigree factor graph, for example sex. Though sex is often certain in studies of humans, in molecular ecology, sex of the sampled individuals may be completely unknown or observed with error. This uncertainty can be accommodated by giving each individual a latent sex state that can be either "male" or "female," and adding a factor node representing the likelihood that an individual is male or female to individuals with sex observed (possibly with error). Then, the sex of individuals can be jointly inferred with the pedigree structure. In a similar fashion, the proposed method could easily be extended to estimate unknown parameters such as genotyping error rates, or, if trait data were available, quantitative genetic parameters given the sampled pedigrees.

In this paper, attention has been restricted to proposals and algorithms for sampling pedigrees that yield singly-connected pedigree factor graphs. This corresponds

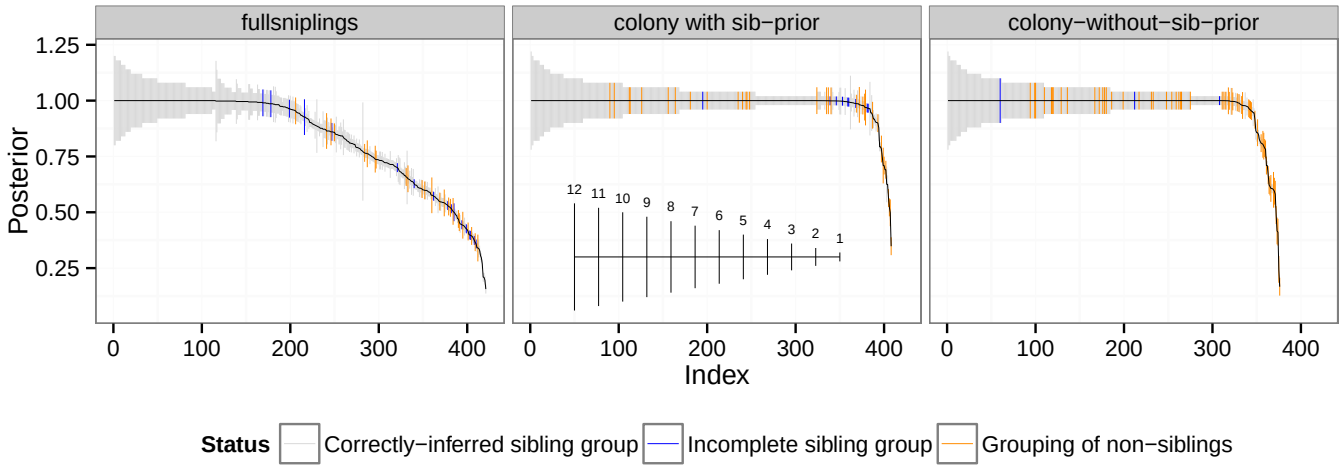


Figure 8: Comparison of FULLSNIPLINGS' inference to COLONY'S on 1,157 Chinook salmon typed at 95 SNPs. Each thin vertical line represents a sibling group whose size (number of individuals) is denoted by its vertical extent (see scale in lower half of middle panel), and whose posterior probability (as estimated by each program, respectively) is denoted by the vertical height of its midpoint (where it is intersected by the black curve). The roughly 400 inferred sibling groups (of size ≥ 1) have been arranged in left-to-right order sorted by posterior probability and, within identical values of posterior probability, by size. Gray vertical bars denote inferred full-sibling groups that are correct; blue are inferred full sibling groups that are not complete (*i.e.*, all members are full siblings of one another, but they do not include all the members of the full sibling group present in the sample); orange are inferred full sibling groups that include individuals that are not truly full siblings. Of note is the fact that FULLSNIPLINGS' estimate of posterior probability more closely reflects the chance that an inferred sibling group is incorrect.

to doing inference over the space of zero-loop pedigrees (Thompson et al., 1978). Of course, the true pedigree may include loops, either those due to inbreeding or those due to marriage chains (Sheehan, 2000). Inbreeding and polygamy occur in many populations, so we are currently investigating and developing methods for inferring pedigrees with loops. The Sum-Product algorithm is not exact on factor graphs with loops, which complicates its application; however, by simulating (and fixing for the current iteration) values of genotypes for cut-set individuals (see Thompson et al. 1978), the loops in the graph may be broken, allowing rapid calculation of the Sum-Product algorithm on the subgraphs. This approach can also be used to break up large connected componenents, so that running the Sum-Product algorithm does not require running it over entire, large connected components. Such an approach is similar to the use of cut-sets to handle loops in the Elston-Stewart algorithm (Elston and Stewart, 1971), and is related to the blocking-Gibbs-sampling approach taken by Jensen and Kong (1999) to improve mixing of MCMC over the space of unobserved genotypes given a fixed pedigree. Another approach to dealing with loops might be to use loopy-belief propagation, though in that case there is no guarantee that the product of messages traversing an edge properly represent the joint probability of all the data. Alternatively, the sum-product algorithm could be run upon a forest of singly-connected pedigree factor graphs whose union includes all the edges in the loopy pedigree, and then the messages actually used for the MCMC updates could be appropriately reweighted versions of the messages on each of the singly-connected graphs, borrowing, for example, from similar methods in the analysis of large complex pedigrees (Bouchard-Côté and Kirkpatrick, 2012).

We have assumed that the L SNPs used for data are unlinked. This assumption is certain to be violated with large numbers of SNPs; however, unless L is much larger than the number of chromosomes, it may provide a reasonable approximation. The data set with 1,157 Chinook salmon included the true physical linkage of the markers, since the data were from real salmon. In that application, the assumption of unlinked markers does not appear to hamper the inference excessively. If the markers were mapped, and not in linkage disequilibrium, and recombination rates between them were available, then it might be possible to formulate an approximation that extends the factor-graph approach by modeling the latent states with a hidden Markov chain. Such calculations would not be exact since the the latent genotypes within a pedigree context do not follow a hidden Markov chain along the chromosome. Exact calculations would require an approach like the algorithm of Lander and Green (1987), and thus is unlikely to be practical on large pedigrees. It appears it would be difficult to extend the pedigree factor graph approach to dense genomic data. Within molecular ecology, however, it would be valuable to extend the method to multiallelic markers to allow inference from short read data from a small number of sites throughout the genome, or from microsatellite markers. Doing so, it is unlikely to be computationally feasible to model every allelic state

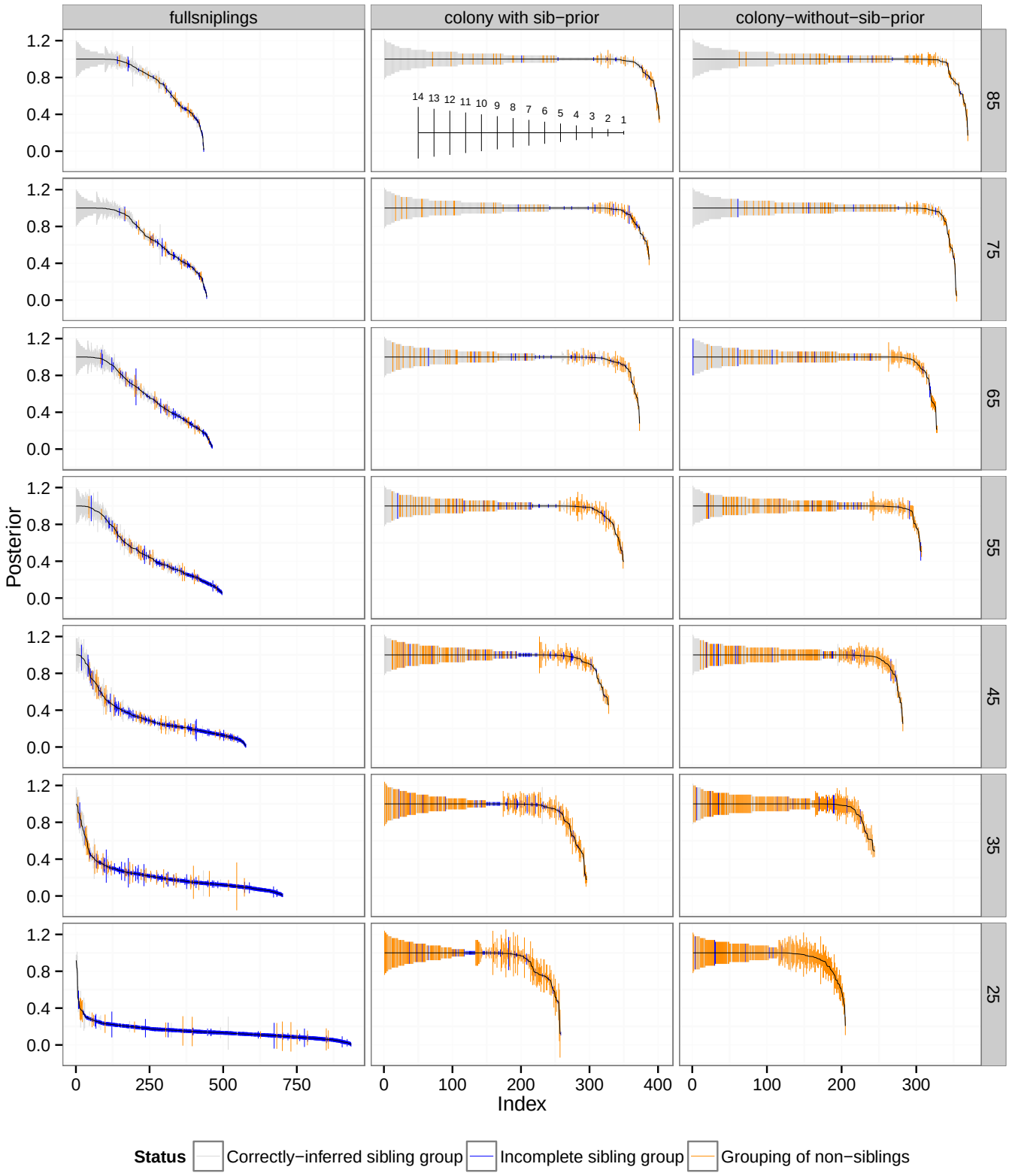


Figure 9: Figures as in Fig. 8 for differing numbers of SNPs, L. Each row shows results for a different value of L, from 85 to 25. Note that x-scales differ between the columns. For all values of L the mean sibship size for COLONY's "sibling-prior" option was set to 3.06, the value obtained when analyzing the data with COLONY and "no-prior" using all 95 loci.

explicity, but some approximation may be possible.

Overall, the factor graph approach provides a useful way of organizing the computations for multigeneration pedigree inference in the presence of unsampled relatives and genotyping error. While additional work will be required to deal with inference of multigenerational pedigrees and pedigrees with loops, or to develop approximations to handle linked SNPs or multiallelic markers, the encouraging performance of the methodology when applied to the full-sibling reconstruction problem indicates that it will be a fruitful area of future research.

Data Accessibility

- R package FULLSNIPLINGS: `https://github.com/eriqande/fullsniplings`
- All data and code to reproduce the analyses of 1,157 Chinook typed at 95 SNPs are available at: `https://github.com/eriqande/tpb-colony-compare`.

Acknowledgements

I gratefully acknowledge helpful discussions with Matthew Stephens, Jarrod Hadfield, Anthony Almudevar, Jinliang Wang and Nuala Sheehan. Two anonymous referees provided valuable criticism that improved the paper considerably. R code for interfacing with the `glpk` package to compute the partition distances in Table 2 was provided by Anthony Almudevar. The Chinook salmon data were provided by Anthony Clemento and Carlos Garza. I thank the organizers of the workshop on "Statistical and computational methods for relatedness and relationship inference from genetic marker data" at the Internaional Centre for Mathematical Statistics, Edinburgh, Scotland, where this work was first presented. This work was supported by Letter of Agreement funding of the Pacific Salmon Commission, Chinook Technical Committee, US Section, and by NSF-OCE-1260693.

References

Almudevar, A., 2003. A simulated annealing algorithm for maximum likelihood pedigree reconstruction. Theor Popul Biol 63, 63–75.

Almudevar, A., Anderson, E.C., 2012. A new version of PRT software for sibling groups reconstruction with comments regarding several issues in the sibling reconstruction problem. Molecular Ecology Resources 12, 164–178.

Almudevar, A., Field, C., 1999. Estimation of single-generation sibling relationships based on DNA markers. Journal of Agricultural, Biological, and Environmental Statistics 4, 136–165.

Almudevar, A., LaCombe, J., 2012. On the choice of prior density for the Bayesian analysis of pedigree structure. Theoretical Population Biology 81, 131–143.

Anderson, E.C., Garza, J.C., 2006. The power of single nucleotide polymorphisms for large-scale parentage inference. Genetics 172, 2567–2582.

Blouin, M.S., 2003. DNA-based methods for pedigree reconstruction and kinship analysis in natural populations. Trends in Ecology & Evolution 18, 503–511.

Bouchard-Côté, A., Kirkpatrick, B., 2012. Bayesian pedigree analysis using measure factorization, in: Advances in Neural Information Processing Systems, pp. 2897–2905.

Christie, M.R., Marine, M.L., Blouin, M.S., 2011. Who are the missing parents? Grandparentage analysis identifies multiple sources of gene flow into a wild population. Molecular Ecology 20, 1263–76.

Cowell, R.G., 2009. Efficient maximum likelihood pedigree reconstruction. Theoretical population biology 76, 285–91.

Eddelbuettel, D., François, R., 2011. Rcpp: Seamless R and C++ integration. Journal of Statistical Software 40, 1–18. URL: `http://www.jstatsoft.org/v40/i08/`.

Elston, R.C., Stewart, J., 1971. A general model for the genetic analysis of pedigree data. Human Heredity 21, 523–542.

Ewens, W.J., 1972. The sampling theory of selectively neutral alleles. Theoretical Population Biology 3, 87–112.

Garant, D., Kruuk, L.E.B., 2005. How to use molecular marker data to measure evolutionary parameters in wild populations. Molecular Ecology 14, 1843–59.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 2004. Bayesian Data Analysis, 2nd edition. Chapman and Hall, New York.

Gusfield, D., 2002. Partition-distance: A problem and class of perfect graphs arising in clustering. Information Processing Letters 82, 159–164.

Hadfield, J.D., Richardson, D.S., Burke, T., 2006. Towards unbiased parentage assignment: combining genetic, behavioural and spatial data in a Bayesian framework. Molecular Ecology 15, 3715–30.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109.

Hoppe, F., 1984. Polya-like urns and the Ewen's sampling formula. Journal of Mathematical Biology 20, 91–94.

Ivy, J.A., Miller, A., Lacy, R.C., Dewoody, J.A., 2009. Methods and prospects for using molecular data in captive breeding programs: an empirical example using parma wallabies (Macropus parma). J Hered 100, 441–54.

Jensen, C.S., Kong, A., 1999. Blocking Gibbs sampling for linkage analysis in large pedigrees with many loops. The American Journal of Human Genetics 65, 885–901.

Johnson, N.L., Kotz, Z., Balakrishnan, N., 1997. Discrete Multivariate Distributions. Wiley & Sons, New York.

Koch, M., Hadfield, J.D., Sefc, K.M., Sturmbauer, C., 2008. Pedigree reconstruction in wild cichlid fish populations. Mol Ecol 17, 4500–11.

Koller, D., Friedman, N., 2009. Probabilistic Graphical Models. MIT Press.

Kschischang, F.R., Frey, B.J., Loeliger, H.A., 2001. Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47, 498–519.

Lander, E.S., Green, P., 1987. Construction of multilocus genetic linkage maps in humans. Proceedings of the National Academy of Sciences 84, 2363–2367.

Landsteiner, K., Levine, P., 1928. On the inheritance of agglutinogens of human blood demonstrable by immune agglutinins. The Journal of experimental medicine 48, 731–749.

Lauritzen, S.L., Sheehan, N.A., 2003. Graphical models for genetic analysis. Statistical Science 18, 489–514.

Letcher, B.H., King, T.L., 2001. Parentage and grandparentage assignment with known and unknown matings: application to Connecticut River Atlantic salmon restoration. Can J Fish Aquat Sci 58, 1812–1821.

Marshall, T.C., Slate, J., Kruuk, L.E.B., Pemberton, J.M., 1998. Statistical confidence for likelihood-based paternity inference in natural populations. Molecular Ecology 7, 639–655.

Meagher, T.R., Thompson, E.A., 1987. Analysis of parentage for naturally established seedlings within a population of *Chamaelirium luteum* (Liliaceae). Ecology 68, 803–812.

Mooij, J.M., 2010. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. Journal of Machine Learning Research 11, 2169–2173. URL: `http://www.jmlr.org/papers/volume11/mooij10a/mooij10a.pdf`.

Nielsen, R., Mattila, D.K., Clapham, P.J., Palsboll, P.J., 2001. Sta-

tistical approaches to paternity analysis in natural populations and applications to the North Atlantic humpback whale. Genetics 157, 1673–82.

Painter, I., 1997. Sibship reconstruction without parental information. Journal of Agricultural, Biological, and Environmental Statistics 2, 212–229.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A.R., Bender, D., Maller, J., Sklar, P., de Bakker, P.I.W., Daly, M.J., Sham, P.C., 2007. PLINK: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet 81, 559–75.

R Core Team, 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL: `http://www.R-project.org/`.

Riester, M., Stadler, P., Klemm, K., 2009. FRANz: Reconstruction of wild multi-generation pedigrees. Bioinformatics 25, 2134–2139.

Roeder, K., Devlin, B., Lindsay, B.G., 1989. Application of maximum likelihood methods to population genetic data for the estimation of individual fertilities. Biometrics 45, 363–380.

Seeb, J.E., Carvalho, G., Hauser, L., Naish, K., Roberts, S., Seeb, L.W., 2011. Single-nucleotide polymorphism (SNP) discovery and applications of SNP genotyping in nonmodel organisms. Molecular Ecology Resources 11 Suppl 1, 1–8.

Sheehan, N., 2000. On the application of Markov chain Monte Carlo methods to genetic analyses on complex pedigrees. International Statistical Review 68, 83–110.

Sheehan, N.A., Bartlett, M., Cussens, J., 2014. Improved maximum likelihood reconstruction of complex multi-generational pedigrees. Theor Popul Biol 97, 11–9.

Sheehan, N.A., Egeland, T., 2007. Structured incorporation of prior information in relationship identification problems. Ann Hum Genet 71, 501–18.

Speed, D., Balding, D.J., 2015. Relatedness in the post-genomic era: is it still useful? Nat Rev Genet 16, 33–44. URL: `http://dx.doi.org/10.1038/nrg3821`.

Steele, C.A., Anderson, E.C., Ackerman, M.W., Hess, M., Campbell, N.R., Narum, S.R., Campbell, M.R., 2013. A validation of parentage-based tagging using hatchery steelhead in the Snake River basin. Canadian Journal of Fisheries and Aquatic Sciences 70, 1046–1054.

Stephens, M., Smith, N.J., Donnelly, P., 2001. A new statistical method for haplotype reconstruction from population data. American Journal of Human Genetics 68, 978–989.

Tautz, D., 1989. Hypervariability of simple sequences as a general source for polymorphic DNA markers. Nucleic Acids Research 17, 6463–6471.

Thompson, E.A., Cannings, C., Skolnick, M.H., 1978. Ancestral inference. I. The problem and the method. Ann Hum Genet 42, 95–108.

Totir, L.R., Fernando, R.L., Abraham, J., et al., 2009. An efficient algorithm to compute marginal posterior genotype probabilities for every member of a pedigree with loops. Genetics Selection Evolution 41, 52.

Visscher, P.M., Hill, W.G., Wray, N.R., 2008. Heritability in the genomics era–concepts and misconceptions. Nat Rev Genet 9, 255–66.

Wang, J., 2004. Sibship reconstruction from genetic data with typing errors. Genetics 166, 1963–79.

Wang, J., Santure, A.W., 2009. Parentage and sibship inference from multilocus genotype data under polygamy. Genetics 181, 1579–94.

Wang, J.L., 2012. Computationally efficient sibship and parentage assignment from multilocus marker data. Genetics 191, 183–194.