Graphical Models for Genetic Analyses

Steffen L. Lauritzen[*]
Aalborg University and MaPhySto[†]

Nuala A. Sheehan
Institute of Genetics, University of Leicester

September 23, 2003

Abstract

This paper introduces graphical models as a natural environment in which to formulate and solve problems in genetics and related areas. Particular emphasis is given to the relationships between various local computation algorithms which have been developed within the hitherto mostly separate areas of graphical models and genetics. The potential of graphical models is explored and illustrated through a number of example applications where the genetic element is substantial or dominating.

Key words: Bayesian network; forensic genetics; linkage analysis; local computation; peeling; probability propagation; QTL analysis.

1 Introduction

In the current climate of rapid development in biological research with modern DNA technology and computing power, genetic analyses involving complex models and family data are becoming both feasible and interesting. Specialised algorithms and software have been developed for the purpose of performing required calculations such as, for example, FASTLINK (Cottigham, Idury and Schäffer 1993), GENEHUNTER (Kruglyak, Daly, Reeve-Daly and Lander 1996), and VITESSE (O'Connell 2001), all of which are programs for the analysis of genetic linkage. The set of familial relationships among a group of individuals forms what is commonly known as a *pedigree* and a

[*]Address for correspondence: Department of Mathematical Sciences, Aalborg University, Fredrik Bajers Vej 7G, DK–9220 Aalborg, Denmark. Email: steffen@math.auc.dk.

[†]MaPhySto — A Network in Mathematical Physics and Stochastics, funded by The Danish National Research Foundation.

variety of graphical representations have been developed for handling pedigrees in a precise and consistent manner. As several authors, including Kong (1991), Heath (2003), and others have acknowledged, such representations logically tempt an exploitation of *graphical models* (Lauritzen 1996) for the description and analysis of genetic problems associated with pedigrees. It is the aim of this paper to explore the potential of a graphical model approach to such genetic analyses, an approach which is also behind the efficient linkage analysis software SUPERLINK developed by Fishelson and Geiger (2002). Here, however, we attempt to exploit the flexibility and generality of graphical models even further.

Many complex genetic computations can only be performed approximately and involve repeated random sampling techniques, typically in the form of Markov chain Monte Carlo (MCMC) methods (Thompson 1994, Jensen, Kjærulff and Kong 1995, Sheehan 2000, Thompson 2000, Thompson 2001, Fernandez, Fernando, Guldbrandtsen, Totir and Carriquiry 2001, Heath 2003). However, this paper has its focus on exact computational methods, noting that these are also of crucial importance to many of the steps within any efficient MCMC algorithm.

The paper is organised as follows. In Section 2 we introduce basic concepts from genetics and graphical models, in Section 3 we give a relatively detailed account of the basic local computation algorithms used in genetics and graphical models and discuss how they relate to each other. In Section 4 we illustrate the approach and its use in a number of applications. Finally, in Section 5, we discuss further perspectives.

2 Preliminaries

This section introduces fundamental and classical concepts in genetics and graphical models with the primary purpose of introducing newcomers to either of the fields, but also to define and describe the terminology used.

2.1 Graphs and pedigrees

We will formally define a pedigree or *genealogy* to be a group of individuals together with a full specification of all the relationships between them (Thompson 1986). It is conventional, but not necessary, to assume that every individual has either both parents in the pedigree, or neither. We define a pair of pedigree members to be *spouses* only if they have mutual offspring in the pedigree and every such pairing is called a *marriage*. Those without parents are called the *founders* of the pedigree and these, by definition, are unrelated. Founders either belong to some baseline generation back to which ancestry has been traced or have married into the pedigree more recently. Pedigrees are commonly represented graphically, although not always strictly as a graph. A standard representation is shown in Figure 1.

A *graph* associated with any particular model is a set of vertices or *nodes* representing the variables in the model and a set of *edges* representing the links between these variables. Edges can be either *directed* with arrows indicating the direction of the link, or *undirected.* Directed edges are also called *arcs.* A pedigree can easily be expressed as a directed graph (Lange and Elston 1975), the simplest of these depicted in Figure 2 where the nodes denote individuals and the arcs connect individuals to their offspring. We shall henceforth refer to this graph as the *relationship graph* and note that it is a standard representation of the transmission of genes from parents to offspring.

A natural extension leads to the *marriage node* graph of Figure 3 (Thomas 1985) which has two kinds of node *and* two kinds of arc (Lange and Elston 1975, Cannings, Thompson and Skolnick 1978). Here, individuals and marriages are represented as nodes, and the connecting arcs are *marriage arcs*, directed from an individual to his marriages, and *descent arcs*, directed from a marriage to the resulting offspring. When drawing a marriage node graph, it is conventional to omit the directions on the arcs since direction is always *down* from parents to offspring via the relevant marriage node.

2.2 Some basic genetics

In diploid individuals, the basic genetic material or DNA in each normal cell is packaged into pairs of homologous strings or *chromosomes.* Human beings, for example, have 23 such pairs, 22 of which are called the *autosomes* and the remaining pair are the sex chromosomes. For a given individual, one chromosome in each pair derives from the DNA of his mother and the other from the DNA of his father. A segment of chromosome which codes for some functional protein is known as a *locus* and we typically refer to the individual's DNA at this locus as his *gene*. Different forms that can be assumed by the DNA at a locus, or different variants of a gene, are called *alleles.* The word "gene" is sometimes used to refer to the actual locus ("the ABO gene") or to the allelic type predisposing an individual to a particular disease ("the breast cancer gene") but here we will follow Thompson (2001) and use the term as Mendel (1866) intended it, to refer to the entity transmitted from parent to offspring. The (unordered) pair of alleles at any locus (one on each chromosome in the pair) is known as the *genotype* and the potentially observable characteristic (if this is the locus for a functional gene) is the *phenotype* (e.g. affected/normal, height, blood type, etc.). With modern DNA technology, the observable phenotype determines the genotype completely in many cases.

Sometimes it is convenient to keep track of the parental origins of each allele in a genotype. The term *ordered genotype*, or *full genotype*, can be used to make this distinction but in this paper, genotypes will be unordered. If both alleles are of the same type, we say that the genotype is *homozygous* while differing allelic types yield a *heterozygous* genotype. If a homozygous genotype (or *homozygote*) has the same phenotype as a heterozygous type with which it has an allele in common, we say that the common allele *dominates* the other allele in the heterozygote. Alternatively, the allele that is not shared is said to be *recessive* to the common one. Table 1 displays the genotypes, associated phenotypes and dominance patterns for the human ABO blood group, simplified to a 3-allele genetic system. The discussion in this paper will be centered around autosomal traits. Although completely analogous, the treatment of sex-linked inheritance is a little different and will not be dealt with here.

According to Mendel's First Law which has set the basis for modern genetics, any given characteristic of an individual is determined by two discrete *factors*, or genes, one of which is a copy of one of the corresponding pair in his mother and the other a copy of one of the paternal pair. Furthermore an individual passes a copy of i.e. *segregates* a randomly chosen one of his two genes to each of his children, independently for different segregations and independently of segregations from the other parent. When genes segregate with equal probability 1/2, we have what is known as *Mendelian segregation*. This is often assumed for autosomal traits. Mendel's Second Law states that segregations of genes at different loci are independent. This is now known not to be true in general: these segregations may be correlated if the loci are close together on the same chromosome or *linked*.

During gamete formation in a process called *meiosis* (see Figure 4), the maternal and paternal copies of a particular chromosome in an individual pair up. Breaks occur at several random positions which allow for the exchange of segments of chromosome within the pair. This is called *crossing over* and refers to the interchange of genetic material between the two homologous chromosomes. The resulting chromosomes, which are mixtures of the maternal and paternal chromosome segments, separate and one of each pair is passed to the *gamete*—the genetic contribution from a single parent to the next generation. The term *haplotype* is often used to refer to a listing of alleles in a single gamete at a given number of loci. They may all lie on the same chromosome (see Figure 4), but not necessarily. The alleles appearing in a haplotype are said to be *in phase* and haplotype information on an individual is known as the phase of that individual's meioses. The correlation in segregations between linked loci is due to the fact that it is highly unlikely that a crossover will occur between two loci which are physically close on the chromosome. Loci which are "far apart" (or on different chromosomes) are more likely to segregate independently in accordance with Mendel's Second Law.

The *recombination fraction* r between two loci is defined as the probability that the genes segregating to the gamete at these loci come from different parental chromosomes. For loci close together, $r \approx 0$ so the two alleles in the gamete (and hence the future offspring) will tend to have the same grand-parental origins. Under assumptions of the meiosis model for most diploid species, the maximum value r can assume is $\frac{1}{2}$ indicating that the loci are segregating independently. A recombination occurs between two loci if there is an odd number of crossovers between them in that meiosis.

The *genetic map distance* between two loci (Haldane 1919) is defined as the expected number of crossovers to occur between them in a gamete and is measured in units called Morgans (or often centiMorgans, for convenience). Various mapping functions exist (Ott 1999) relating map distance to recombination fractions. The one we will use in this paper is due to Haldane and assumes that crossovers occur as a Poisson process with rate 1 per Morgan and that numbers of crossovers in non-overlapping intervals are independent. Under this model of *no-interference*, the relationship between λ, the genetic distance between any two loci, and the corresponding recombination fraction r is given by

$$r = \frac{1}{2}(1 - e^{-2\lambda})$$

with inverse function

$$\lambda = -\frac{1}{2}\log(1 - 2r).$$

Because they are expectations, map distances are additive and hence may be more convenient to work with from a computational viewpoint. In a probabilistic model, however, it is more natural to think of linkage using recombination fractions. For this reason, it is common to vary between the two quantities within an analysis using an appropriate mapping function and the two terms are often used interchangeably.

In genetic linkage studies, the aim is to localise the genes for some trait of interest by mapping their positions relative to known *marker* loci within the pedigrees being studied. A genetic marker locus can be defined as a position on a particular chromosome which is characterised by a specific DNA sequence or observable variations in the sequence. In linkage studies, marker loci are assumed not to have any effect on the trait under consideration. Good estimation of the recombination fraction is often restricted by the pedigree size and structure and so a linkage analysis is generally viewed as a first step in the mapping process with the aim of identifying a general chromosomal region of interest. The precise location of the gene is then determined by a study giving finer resolution using *linkage disequilibrium* mapping, for example (Heath 2003).

A genetic trait for which the expressed phenotype corresponds to the genotype at a single locus is often called a *Mendelian* or single-locus trait. Such traits are generally well-understood and many have been successfully mapped over the last two decades using standard techniques. Examples include cystic fibrosis (Riordan, Rommens, Kerem, Alon, Rozmahel, Grzelczak, Zielenski, Lok, Plavsic, Chou, Drumm, Iannuzzi, Collins and Tsui 1989) and Duchenne's muscular distrophy (Monaco, Bertelson, Middlesworth, Colletti, Aldridge, Fischbeck, Bartlett, Pericak-Vance, Roses and Kunkel 1985). The human ABO blood group of Table 1 is an example of a discrete Mendelian trait whereby the observed phenotypes can be classified into distinct categories. A *quantitative* trait has a phenotype which is affected by the simultaneous segregation of many genes at many loci (we call this *polygenic* variation), and may in addition have some non-genetic variation superimposed (Falconer and Mackay 1996). Quantitative traits can exhibit variation on a continuous scale (e.g. height, weight etc.) but can also be discrete as in threshold traits. A *quantitative trait locus* (QTL) can be thought of as a segment of chromosome affecting a quantitative trait but whose effect is not large enough to cause an observable discontinuity and is hence not detectable using Mendelian methods. More generally, *complex* genetic traits are those for which the simple correspondence between genotype and phenotype breaks down (Lander and Schork 1994). They include discrete, continuous and quantitative traits and may also have multivariate phenotypes measured on either discrete or continuous scales. They can also have interaction effects in that the underlying genotype effects on the trait phenotype may vary with age and sex, for example, and various environmental factors may have to be accounted for. Coronary heart disease is an example of such a trait: despite the strong evidence for a genetic component to heart disease, few genes have been identified which clearly influence the risk of developing the condition (Thompson and Wijsman 1990).

For a more detailed discussion of the basic genetic concepts introduced in this section, see Thompson (2000) and Sham (1998).

2.3 Elements of Bayesian networks

A probabilistic approach to dealing with uncertainty in expert systems began with the realisation that calculations on seemingly intractable high dimensional problems can be efficiently performed when a set of simplifying conditional independence assumptions is imposed (Pearl 1988, Lauritzen and Spiegelhalter 1988). These assumptions essentially split the problem into small manageable components. The immediate advantage is that a complex problem can be represented in a graphical form which can then inform the development of efficient computational algorithms for performing calculations. The most common of these *graphical models* (Lauritzen 1996) are the *Bayesian networks* (Pearl 1986, Jensen 1996) but more general types of network may be appropriate and lend themselves to essentially the same computational simplifications (Cowell, Dawid, Lauritzen and Spiegelhalter 1999).

It is unfortunate for our applications that the terminology traditionally used in this area derives from genetics (Sheehan, Guldbrandtsen, Lund and Sorensen 2002). For instance, for nodes labelled a and b, we say that a is a *parent* of b, or b is a *child* of a, if there is a directed edge from a to b. In contrast with the biological interpretation of these terms, a node in a graph can have more than two parents (e.g. Figure 5). Usually it will be clear from the context whether terms such as 'parent' refer to the graph-theoretic notion or its biological analogue. When there can be ambiguity, we will further qualify the term using expressions like *graph parent* or *bio parent*.

Recall that a graph is a collection of nodes and edges which can either be directed (arcs) or undirected. A *trail* in a graph is defined to be a sequence of edges, each having a node in common with both preceding and succeeding edges. A *path* is a trail with no edges violating the direction of the trail. If all edges of the trail are undirected, the path is *undirected* and the path is *directed* if all edges are directed. If there is a path from node a to node b (i.e. we can arrive at b by following arrows from a), we say that a is a (graph) *ancestor* of b and b is a (graph) *descendant* of a. A trail beginning and ending with the same node is a *cycle* or *loop*. If all the edges of a graph are directed, it is a directed graph and if it has no directed cycles, it is a *directed acyclic graph* or DAG (Cowell et al. 1999). A graph is *connected* if there is a trail between any pair of nodes. A connected graph with no cycles is a *tree*. In this paper, unless otherwise stated, it will be assumed that all graphs are connected.

Returning to the marriage node graph representation of our simple pedigree in Figure 3, note that individuals $5, 6, 7$ and 8 are all connected by marriage edges. The corresponding trail is known as a *marriage chain*. A directed path in a marriage node graph is an alternating sequence of marriage and descent arcs and since an individual cannot be his own biological ancestor or descendant, there are no directed cycles in a pedigree. Hence, a marriage node graph is a DAG and so is the relationship graph of a pedigree (e.g. Figure 2). However, loops arise more easily in the relationship graph as can be seen in Figure 2. The loop $14 - 10 - 13 - 9 - 14$ formed by siblings 13 and 14 does not feature in the marriage node graph representation of Figure 3. Pedigrees are described as either *looped* or *unlooped* according to whether or not their marriage node graphs have undirected cycles. An *inbreeding* loop arises for example when two biologically related individuals marry causing two separate paths of descent from a common ancestor to the node representing their marriage. Other loops include marriage rings, exchange loops, multiple marriage loops and all kinds of interconnecting combinations of the above (Cannings et al. 1978).

A *Bayesian network* is a DAG with node set V, where the nodes represent random variables, $X = (X_v)_{v \in V}$, having some joint probability distribution function of the form:

$$f(x) = \prod_{v \in V} f(x_v \mid x_{\mathrm{pa}(v)}) \tag{1}$$

with $\mathrm{pa}(v)$ denoting the set of parent nodes of the node v and $x_A = (x_v)_{v \in A}$ for any subset $A \subseteq V$. It then holds that any node, given the values at its parents, is conditionally independent of all nodes which are not descendants. This is known as the *directed local Markov property.* Further independences can be deduced from the *global directed Markov property* which gives a complete description of independence relationships associated with a Bayesian network. In fact, the factorization (1) is equivalent to either of the local and global directed Markov properties; see Lauritzen (1996) for details concerning these Markov properties. Note that through (1), the joint distribution of a Bayesian network is completely specified from the associated DAG and the conditional distributions of each node given its parents.

It may be worth emphasizing that the term 'Bayesian network' has no direct reference to 'Bayesian inference'; it is referring to its common usage in expert systems, where the networks were designed for efficient calculation of 'reverse conditional probabilities' as in Bayes' formula.

2.4 Bayesian network representations for pedigrees

In order to express a pedigree as a Bayesian network, the graph nodes should represent random variables for which a joint probability distribution can be defined satisfying the factorization in (1). There are several ways of designing such a network and these various representations have different properties. The visually most parsimonious representation, although not necessarily the most useful, is the *genotype* network. This uses the relationship graph (see Figure 2) as the underlying DAG, the nodes now representing the genotypes of the individuals rather than the individuals themselves. We will later return to this representation but initially we describe the *segregation network* since this is the most direct and complete representation of the inheritance relationships in a pedigree. We will discuss the various representations by way of example, using a single-locus discrete genetic trait. As in Sheehan (2000) we consider a pedigree of m individuals uniquely labelled by the integers $1, \ldots, m$.

The segregation network This is a Bayesian network constructed as follows. For each individual i, we have two nodes which represent the maternally and paternally inherited genes. The underlying random variables can assume any of the a allelic types in the system. Following common usage (Thompson 2001) we will use 0 to label maternal inheritance and 1 for paternal inheritance. Thus the node labelled i^1 is identified with the random variable L_{i^1} assigning the allelic type of the gene inherited by individual i from his father.

For each non-founder, arcs are directed from the two genes in the father to the paternal gene in the individual and, similarly, the individual's maternal gene is a (graph) child of the two genes in his mother. Additional nodes representing the *meiosis* or *segregation* indicators (Thompson 1994, Sobel and Lange 1996) are then added as parents to each gene node. These are binary nodes assuming the value 1 to denote that a copy of the paternal gene in the corresponding parent was inherited and 0 to indicate inheritance from the maternal gene. In this way, each allelic type of a non-founder is a deterministic function of its (graph) parents. For the paternally inherited gene:

$$L_{i^1} = f(l_{p_i^1}, l_{p_i^0}, s_{p_i,i}) = \begin{cases} l_{p_i^1} & \text{if } s_{p_i,i} = 1 \\ l_{p_i^0} & \text{if } s_{p_i,i} = 0 \end{cases} \tag{2}$$

and similarly for the maternally inherited gene:

$$L_{i^0} = f(l_{m_i^1}, l_{m_i^0}, s_{m_i,i}) = \begin{cases} l_{m_i^1} & \text{if } s_{m_i,i} = 1 \\ l_{m_i^0} & \text{if } s_{m_i,i} = 0, \end{cases} \tag{3}$$

where m_i and p_i are the labels of the mother and father of individual i and $S_{m_i,i}$ and $S_{p_i,i}$ are binary random variables assigning indicators for the segregations to i from the mother and father, respectively.

The laws of inheritance can now be encoded by letting the segregation indicators be independent with *transmission probabilities*

$$P(S_{p_i,i} = 1) = \sigma_1, \text{ and } P(S_{m_i,i} = 1) = \sigma_0. \tag{4}$$

In the simplest case of Mendelian inheritance, we have $\sigma_1 = \sigma_0 = 1/2$, and this assumption is reasonable for many autosomal traits. But we emphasize that the segregation network has sufficient detail to represent more complex inheritance laws.

The assumption of random union of gametes is fully incorporated in the segregation network of Figure 6 with the graph clearly indicating that founder genes are independent of each other and of the segregation indicators. This implies *Hardy–Weinberg proportions* for founder genotypes. The probability with which founder genes arrive into the pedigree could be $P(L_{i^1} = l) = P(L_{i^0} = l) = f_l$, where f_l is the population frequency of the allelic type l, for instance. These allele frequencies may of course differ between individuals if there are known differences in race, breed, species etc.

The allele network The segregation network provides the most detailed description of the inheritance relationships in a pedigree at a single locus. In many contexts it is unnecessary to keep track of the full details and the segregation network can then conveniently be reduced to what we term an *allele network*, which is obtained from the segregation network by removing the segregation indicators and the associated arcs. Conditional probability distributions of the allelic types given their parents are easily derived from (2), (3) and (4) to become

$$P(L_{i^1} = l \,|\, l_{p_{i^1}}, l_{p_{i^0}}) = \begin{cases} \sigma_1 & \text{if } l = l_{p_{i^1}} \\ 1 - \sigma_1 & \text{if } l = l_{p_{i^0}} \end{cases},$$

and similarly for the maternally inherited gene. Since (2) and (3) define the transmission to a given node as deterministic functions of its (graph) parents in the allele network and independent noise variables (the segregation indicators), Theorem 2.20 of Lauritzen (2001) ensures that the reduction from segregation to allele network preserves the local Markov property of the corresponding DAG and is hence a Bayesian network. Figure 7 shows the allele network for a single locus corresponding to our simple pedigree of Section 2.

We note that what we have termed an allele network is also a standard representation and figures similar to Figure 7 can be found in Jensen (1997), Thompson and Heath (2000), and Thompson (2001) for example. In particular, Jensen (1997) uses the term "gene representation" for our allele network and Thompson and Heath (2000) call it the "gene pedigree".

The genotype network For certain purposes it may be advantageous to make yet a further reduction to obtain what we term a *genotype network*. Again, we note that this representation is called a "genotype representation" in Jensen (1997) and also features in Heath (2003) and Spiegelhalter (1990). Here we let G_i denote the individual's genotype at the locus of interest, so that $G_i = \{L_{i^1}, L_{i^0}\}$ and as the basic DAG we use the representation graph of Figure 2 with each node i in the DAG representing the random variable assigning a genotype to individual i. The graph parents of non-founder nodes represent the genotypes of the biological parents of the individual.

To verify that the inheritance model represented by the allele and segregation networks implies the local Markov property for the genotype network, we must ensure that the genotype of individual i, G_i, is conditionally independent of the genotypes of his non-descendants, given the genotypes of his parents. In contrast to the segregation and allele networks, *this holds only under the Mendelian inheritance model.* Otherwise, the haplotype (or phase) information on the parental genotype will be informative for segregation to children.

As an example of this, consider a diallelic trait and a nuclear family with a homozygous mother, i.e. with genotype $g_m = \{A, A\}$, and heterozygous father $g_p = \{A, a\}$. The conditional probability of a second child getting genotype $G_2 = \{A, A\}$, given the genotype g_1 of the first child becomes

$$P(G_2 = \{A, A\} \mid g_m, g_p, g_1) = \sigma_1^2 + (1 - \sigma_1)^2 \text{ when } g_1 = \{A, A\} \quad (5)$$

with both children inheriting a copy of the same gene from their father, and

$$P(G_2 = \{A, A\} \mid g_m, g_p, g_1) = 2\sigma_1(1 - \sigma_1) \text{ when } g_1 = \{A, a\} \quad (6)$$

when they inherit copies of different paternal genes. Hence, genotypes of siblings are *not* conditionally independent given the genotypes of their parents unless $\sigma_i = 1/2$ in which case, the right hand sides of (5) and (6) are identical and equal to 1/2.

Although the genotype network is visually less complex than the allele network, a price has been paid, not only by restricting the inheritance model to be Mendelian, but also by increasing the state spaces at each node: For a genetic system with a alleles, the genotypes G_i can assume any of $a(a+1)/2$ distinct states, whereas the nodes in an allele network only have a states. This can be of substantial importance for computational issues and will be discussed in further detail in Section 3.

For the genotype network, we need to derive probabilities both for founder genotypes and for the inheritance of genotypes from parental genotypes, the latter also referred to as *transmission probabilites* in the literature. If we denote the probabilities of the founder genotypes by π and the transmission probabilities by

$$\tau(g_i \,|\, g_{m_i}, g_{p_i}) = P(G_i = g_i \,|\, G_{m_i} = g_{m_i}, G_{p_i} = g_{p_i})$$

the factorization (1) yields the familiar expression

$$P(g_1, \ldots, g_m) = \prod_{i \in \mathcal{F}} \pi(g_i) \prod_{j \notin \mathcal{F}} \tau(g_j \,|\, g_{m_j}, g_{f_j}), \tag{7}$$

where $\mathcal{F}$ denotes the set of individuals which are founders of the pedigree and the transmission probabilities, τ, are Mendelian. See Thompson (1986), for example.

Phenotypic information Each of the types of network above specifies the inheritance relationships without referring to the observational situation in any given context. Although the genotypes may be identifiable from the phenotypes in many cases, they often are not and only partial information is available in some situations. To accommodate such data, an extra node is added for each individual where phenotypic information is available, and possibly also for other individuals, depending on the purpose of the analysis. We let Y_i denote the variable associated with the phenotype of individual i. In the allele and segregation networks, the node carrying the phenotype Y_i has the two alleles of individual i as parents, whereas in the genotype network, Y_i has the genotype G_i as its only parent. If the genotype is itself observable, $Y_i = G_i$, and we can omit this extra node in the genotype network.

The local Markov property of the genotype network augmented with phenotypic information is ensured by the phenotype Y_i of any individual being conditionally independent of other variables in the network, given the genotype G_i of that individual. When the paternal and maternal alleles influence the phenotype differently (*genetic imprinting*), this conditional independence is violated. The allele and segregation networks contain sufficiently detailed information for the Markov property to hold in the augmented network. Alternatively, one could define a network in terms of *ordered* genotypes where the Markov property would then hold.

The conditional distribution of the phenotype Y_i given its (graph) parents is known as the *penetrance distribution*. This may often be given through a deterministic relationship, for example when $Y_i = G_i$, or through a more complicated function such as the penetrance function for the ABO blood group system of Table 1. Figure 8 shows an allele network augmented with phenotype nodes for two individuals.

3 Local computation on graphs

Almost every problem associated with pedigree analysis or other complex genetic problems involves a difficult computation. This could be the computation of a likelihood, the probability of an individual having a specific allele, genotype or haplotype, or some other characteristic of the system under investigation. Superficially such computations seem too complex to be feasible at all and indeed many are not. However, there are a number of related computational algorithms which exploit the local structure of the system, including that of the pedigree. These algorithms yield drastic reductions in the computational complexity. In genetic applications, such computation is typically referred to as 'peeling' (Elston and Stewart 1971, Cannings et al. 1978, Lander and Green 1987). See also Thompson (2000, 2001) and Heath (2003), for further discussion.

The peeling algorithms are special cases, or variants, of general algorithms for so-called *local computation* on graphs (Cowell et al. 1999). In this section we describe and explain the general types of algorithm and their relation to peeling algorithms.

3.1 General algorithm

The general algorithm for local computation can be seen as having two phases. During the first phase, a suitable computational structure is established. In the second phase, the computations themselves are executed. The first phase is sometimes referred to as *compilation,* the latter as *propagation.* In the genetic literature it is more usual to describe the peeling algorithm with these two phases executed simultaneously.

3.1.1 Compilation

The compilation process involves the collection of groups of variables into *cliques* so that computations can be performed locally to these, i.e. only involving functions of sets of variables belonging to the same clique. At the next stage, these cliques are organized in a tree structure, the *junction tree*, which is used to coordinate the local computations in a consistent way to yield the desired correct global result. Finally the numbers to be used in the calculations are associated with the relevant location in the junction tree. The various steps of the compilation process are as follows, to be described in further detail below:

- From Bayesian network to undirected graph.
- Triangulation.
- Constructing the junction tree.
- Loading the junction tree.
- Incorporation of observations

From Bayesian network to undirected graph The local computation algorithms are most conveniently and efficiently described in terms of undirected graphs, thereby displaying their full flexibility and symmetries. The first step, therefore, is to transform the Bayesian network into an undirected graph. This is done by removing the directions from the existing edges and adding further undirected edges between all pairs of graph parents with a common (graph) child node. The latter process is referred to as *moralising* the graph i.e. by 'marrying' the (graph) parents. In the resulting *moral graph*, all sets of the form $\{v\} \cup \mathrm{pa}(v)$ are *complete* in the graph, meaning that all pairs of elements are connected with edges. The factorization (1) can therefore be written as

$$f(x) = \prod_{v \in V} f(x_v \mid x_{\mathrm{pa}(v)}) = \prod_{C \in \mathcal{C}} \phi_C(x_C), \tag{8}$$

where $\mathcal{C}$ denotes the set of *cliques* of the moral graph i.e. the maximal complete subsets of nodes, and the functions ϕ are the *potentials*. To obtain this factorization we just collect factors $f(x_v \mid x_{\mathrm{pa}(v)})$ with $\{v\} \cup \mathrm{pa}(v)$ in the clique C so the potential ϕ_C is a product of these factors. Since $\{v\} \cup \mathrm{pa}(v)$ is complete in the moral graph, this can always be done. Heath (2003) uses the term *dependency graph* for the moral graph.

Figure 8 shows a Bayesian allele network corresponding to a modification of the pedigree in Figure 1, where individuals 12 and 14 have married and the phenotypes of their common offspring (15) and of individual 11 have been explicitly represented. The corresponding moral graph is displayed in Figure 9. Note that it is the graph parents (i.e. the allele pairs) which are married in this graph and not the bio parents.

Triangulation The next step of the compilation process is to *triangulate* the graph by adding *fill-in* edges to the moral graph until all cycles involving more than three nodes have *chords*. For a pedigree without loops, this step is unnecessary in the case of a single locus analysis, as the moral graph will then satisfy this requirement automatically. However, in linkage analysis problems, for example, even an unlooped pedigree may induce long *chordless* cycles in the moral graph of the corresponding Bayesian network representation. Computational difficulties associated with pedigree analysis are related to these cycles rather than to the loops of the pedigree graph. An example of such a cycle in a simple *half-sib* design is given later in Section 4.2. The technical reason for triangulating the graph is that the cliques of an undirected graph can only be arranged in a junction tree after this has taken place. Further details will be presented below.

The moral graph in Figure 9 has a chordless cycle, comprising the nodes $10^m, 6^m, 11^m, 11^p, 7^m, 12^p, 15^p, 15^m, 14^m, 10^m$. Note that if the phenotypic information on individual 15 in Figure 8 were not represented, 15^p and 15^m would not be connected and the allele network would have no chordless cycles in its moral graph, despite the fact that the marriage node graph of the corresponding pedigree has loops.

A triangulation of the graph in Figure 9 is displayed in Figure 10, where 6 fill-in edges have been added. Such a triangulation is most often found by using an ordering for node *elimination*; when a node is eliminated, fill-in edges are added between any pairs of neighbours of the node which are not already connected by an edge. The node is then removed together with all its neighbours. Adding the fill-in edges produced in this way to the original set of edges in the moral graph will always produce a triangulated graph. Conversely, for any triangulation of the graph there is an elimination ordering which produces it; cf. Proposition 2.17 of Lauritzen (1996). The notion of an elimination ordering is identical to what is known in the genetics literature as a *peeling sequence,* and the term 'peeling' refers to the elimination process. The fill-in edges ensure that the 'cutsets' of Cannings et al. (1978) become complete sets so the corresponding R-functions created during the associated computations are local to these cutsets (Heath 2003). In this way, a triangulated graph is also being created during a standard peeling process, although it is typically not represented in explicit form. The factorization (8) clearly implies a similar factorization

$$f(x) = \prod_{C \in \mathcal{C}} \phi_C(x_C), \tag{9}$$

where $\mathcal{C}$ now denotes the set of cliques in the triangulated graph, since cliques in the moral graph are complete in any graph with more edges.

Figure 10 also displays the elimination order used to produce the given triangulation. A triangulation is not unique and the goal is to generate *cliques* (maximal sets of pairwise connected nodes) which are as small as possible. Optimizing this step is known to be NP-complete (Yannakakis 1981), but there are several heuristic algorithms which find good triangulations (George and Liu 1989, Kjærulff 1992, Amestoy, Davis and Duff 1996). In fact, there are also algorithms, which in most cases run at reasonable computational speed and are guaranteed to return an optimal triangulation. Such an algorithm has been implemented in version 6 of the commercially available software HUGIN (Andersen, Olesen, Jensen and Jensen 1989). This algorithm is based on work of Shoiket and Geiger (1997), Berry, Bordat and Cogis (2000), and Bouchitté and Todinca (2001), and it is described in Jensen (2002).

The triangulation step is crucial, as the computational complexity is essentially determined by the total size of the state spaces associated with the cliques of the resulting triangulated graph. This determines whether exact computations are at all feasible and whether approximate methods such as MCMC methods will be required. Since the total size is exponential in the size of the cliques, the clique with largest state space makes a dominating contribution to the complexity. For the triangulation in Figure 10, the clique with the largest state space is $\{6^p, 6^m, 10^m, 11^m\}$ with the associated state space having a^4 states for a model with a distinct allelic types. For comparison, a genotype representation of the same pedigree yields the triangulation shown in Figure 11 with a largest clique state space of $(a(a+1)/2)^3$. So, even when $a = 2$ the allele network is preferable here.

Constructing the junction tree Once the graph has been triangulated the cliques can easily be identified and connected in what is known as a *junction tree*; see Sections 4.3 and 4.4 in Cowell et al. (1999) for a description of the construction algorithms. This is a tree having the set $\mathcal{C}$ of cliques of a triangulated graph as nodes, and satisfying the further property that

$$C \cap D \subseteq E \text{ for all } C, D, E \in \mathcal{C} \text{ with } E \text{ between } C \text{ and } D, \tag{10}$$

where E is between C and D if it lies on the unique path from C to D. This property is crucial for the correctness of the propagation algorithm described later and the set of cliques in an undirected graph can be arranged in a tree with property (10) if and only if the graph is triangulated (Theorem 4.6 of Cowell et al. (1999)). A junction tree for the triangulated graph in Figure 10 is displayed in Figure 12. The junction tree property (10) is, for example, reflected through $C_{11} \cap C_{18} = \{12^p\}$ being contained in all of the cliques C_{16},C_{17}, C_{20} and C_{19}, placed between C_{12} and C_{16} in the junction tree.

Loading the junction tree The next step is to identify the *potentials* ϕ_C in the factorization (9). This is done by collecting factors of the form $f(x_v \,|\, x_{\mathrm{pa}(v)})$ in (8) into cliques which contain both v *and* $\mathrm{pa}(v)$. For each node v at least one such clique exists and we choose one of them, say C, and *assign* the node v to C. If $V(C)$ denotes the set of nodes which are assigned to C, we let $\phi_C(x_C) \equiv 1$ for $V(C) = \emptyset$ and else

$$\phi_C(x_C) = \prod_{v \in V(C)} f(x_v \,|\, x_{\mathrm{pa}(v)}),$$

whereby (9) is clearly satisfied with the joint distribution expressible as a product of potentials over the cliques. This concludes the general part of the compilation process. Table 2 gives the full list of cliques, assignments, and potentials for the triangulated allele network of Figure 10.

For a trait with a distinct alleles and where the phenotype of the ith individual, Y_i, is equal to the genotype G_i, the total size of the computational structure in Table 2 with 21 cliques is equal to

$$2a^4 + 2a^3(a+1)/2 + 17a^3$$

, whereas the structure for the corresponding genotype network (triangulated in Figure 11) has total size equal to $11(a(a+1)/2)^3$, since it has 11 equally sized cliques. Table 3 indicates how the sizes of the state spaces for both networks vary with the number of alleles in the genetic system.

Incorporating observations The compilation process described above has not yet taken account of the available data for the analysis in question. The representation (9) gives the joint probability of an arbitrary configuration of variables in the network. However, we want the probability of configurations which are consistent with the observations. This can be obtained if for all v where $X_v = x_v^*$ is observed we find a clique C with $v \in C$ and modify the potential there to ϕ_C^* by changing appropriate values to zero. More precisely, we let

$$\phi_C^*(x_C) = \begin{cases} \phi_C(x_C) & \text{if } x_v = x_v^* \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

This then implies that $\prod_{\mathcal{C}} \phi_C^*(x_C)$ is equal to the joint probability of an arbitrary configuration x which is consistent with the observations. The process of forming ϕ^* from ϕ is often referred to as *entering evidence.* If we denote the set of observed nodes with E we have

$$f(x \mid x_E^*) = \frac{\prod_{C \in \mathcal{C}} \phi_C^*(x_C)}{Z(x_E^*)}, \tag{12}$$

where the normalizing constant $Z(x_E^*)$ is the probability of the observations, obtained by summing over all configurations which are consistent with the observations:

$$f(x_E^*) = Z(x_E^*) = \sum_{x : x_E = x_E^*} \prod_{C \in \mathcal{C}} \phi_C(x_C) = \sum_x \prod_{C \in \mathcal{C}} \phi_C^*(x_C). \tag{13}$$

This also yields the *likelihood* when comparing different models.

In the example considered, we may have observed phenotypes $Y_{11} = y_{11}^*$ and $Y_{15} = y_{15}^*$, and the potentials in the two first cliques in Table 2 should therefore be modified to incorporate this information by setting their values equal to zero for other values of Y_{11} or Y_{15}.

3.1.2 Propagation

In the second part of the algorithm, often referred to as *propagation of evidence*, the actual computations with numbers are made, and the probabilities of interest are calculated. In particular, the sum in (13) must be calculated with more sophisticated techniques than brute force, since the number of terms in the sum grows exponentially with the number of nodes in the network.

There are several variants of the general algorithm of which we choose to describe two in some detail: the Shafer-Shenoy procedure (Shenoy and Shafer 1990) and the HUGIN procedure (Jensen, Lauritzen and Olesen 1990), which represents a refinement of the algorithm in Lauritzen and Spiegelhalter (1988). The Shafer–Shenoy procedure appears to be closest to what is known as 'peeling', but it includes the more general variant used in Thompson (1981) to derive gene probabilities for all individuals in the pedigree. In Thompson (2000), page 98, this variant is referred to as 'reverse peeling' although 'simultaneous peeling' (to all individuals) would seem a more appropriate term.

Peeling towards a root To help clarify the relationship between genetic peeling and these algorithms, we initially describe a scheme which essentially is a junction tree formulation of peeling as described in Cannings et al. (1978).

As a first step, we choose one of the cliques to be a *root* R of the tree, and the algorithm then proceeds by passing appropriate *messages* towards this root. Initially the messages are sent from the leaves of the junction tree, where a *leaf* of the tree is any clique other than the root which has only a single neighbouring clique. The messages gradually progress towards the root as the leaves are 'peeled' off the junction tree.

More precisely, we denote a generic clique potential of the junction tree by ψ_C and we initially have $\psi_C = \phi^*_C$, $C \in \mathcal{C}$. When a *message* is sent from a leaf L to its neighbour D, the potential ψ_L is *marginalized* to $S = L \cap D$ as

$$\psi_L^{\downarrow S}(x_S) = \sum_{y_{L \setminus D}} \psi_L(x_S, y_{L\setminus D}). \tag{14}$$

Furthermore, the neighbouring clique D *absorbs* the message from L by modifying its potential ψ_D to $\tilde{\psi}_D$ as

$$\tilde{\psi}_D = \psi_D \psi_L^{\downarrow S}. \tag{15}$$

Finally the leaf L is removed, i.e. it is 'peeled' off the junction tree.

The marginal $\psi_L^{\downarrow S}$ in (14) and (15) is exactly the R-function of Cannings et al. (1978) and the separator S is the cutset. The important fact is that these calculations are 'local' to the cliques L and D and therefore performed with relatively few variables, as long as the cliques are small. Therefore it is crucial for the algorithm to obtain small cliques in the triangulation process described earlier.

Now let $L' = L \setminus D$, $V' = V \setminus L'$ and $\mathcal{C}' = \mathcal{C} \setminus \{L\}$, where V is the set of nodes and $\mathcal{C}$ is the set of cliques as before. We then have that

$$V = L' \cup V' \text{ and } V' = \bigcup_{C \in \mathcal{C}'} C,$$

since any node v in the leaf L which also belongs to another clique, must be in D by the junction tree property (10). As we shall see shortly, after a message has been sent and absorbed as above, it holds that

$$f(x_{V'} \,|\, x_E^*) = \frac{\prod_{C \in \mathcal{C}'} \tilde{\psi}_C(x_C)}{Z(x_E^*)}, \tag{16}$$

and

$$Z(x_E^*) = f(x_E^*) = \sum_x \prod_{C \in \mathcal{C}'} \tilde{\psi}_C(x_C),$$

where $\tilde{\psi}_C$ are the potentials modified through the passing of the message from L to D. Note that only ψ_D is modified so $\tilde{\psi}_C = \psi_C$ for $C \neq D$. In effect, the summation is only over configurations consistent with the evidence x_E^* because initially $\psi_C = \phi_C^*$ for all C as in (11) and all other values are equal to zero.

Equation (16) holds because

$$\begin{aligned} f(x_{V'} \,|\, x_E^*) &= \sum_{y_{L'}} f(y_{L'}, x_{V'} \,|\, x_E^*) = \sum_{y_{L'}} \frac{\prod_{C \in \mathcal{C}} \psi_C(y_{L' \cap C}, x_{V' \cap C})}{Z(x_E^*)} \\ &= \frac{\prod_{C \in \mathcal{C}' \setminus \{D\}} \psi_[C(x_C)}{Z(x_E^*)} \sum_{y_{L'}} \psi_L(y_{L'}, x_S) \psi_D(x_D) \\ &= \frac{\prod_{C \in \mathcal{C}' \setminus \{D\}} \tilde{\psi}_C(x_C)}{Z(x_E^*)} \psi_D(x_D) \psi_L^{\downarrow S}(x_S) \\ &= \frac{\prod_{C \in \mathcal{C}'} \tilde{\psi}_C(x_C)}{Z(x_E^*)}. \end{aligned}$$

Proceeding in this fashion, all cliques other than the root R are eventually peeled so we finally have

$$f(x_R \,|\, x_E^*) = \psi_R(x_R)/Z(x_E^*) \text{ and } Z(x_E^*) = \sum_{x_R} \psi_R(x_R), \tag{17}$$

where ψ_R now refers to the modified potential after all messages have been sent. Thus we have succeeded in calculating Z and the conditional joint probability of all configurations of nodes in the root clique R. Their individual conditional probabilities can then be obtained by a further simple summation. This message passing scheme is illustrated in Figure 13.

The Shafer–Shenoy procedure If the marginal probability $f(x_E^*)$ is the only quantity of concern, the peeling procedure described is fully satisfactory and no further computation is needed. However, in many cases it is also of interest, for example, to calculate updated probabilities $f(x_v \mid x_E^*)$ for nodes v which are not elements of the chosen root R. In principle, one could then find another clique C with $v \in C$ and repeat the full scheme with a new root $R' = C$. However, many quantities would then be recalculated several times. With a bit of clever bookkeeping, as also noted by Thompson (1981), this repetition can be avoided. A systematic approach to this has been described in Shenoy and Shafer (1990) and proceeds as follows.

With every branch of the junction tree between neighbours C and D we associate a *separator* $S = C \cap D$ and place two *mailboxes* along S, one for messages from C to D and one for messages in the reverse direction, from D to C. We initialize all the mailboxes to be *empty* and they become *full* when a message is placed in them. A clique C may now send a message to D if and only if all its incoming mailboxes other than that coming from D are full. Thus at first, only the leaves are allowed to send messages as in the peeling procedure just described.

The structure of a message $\mu_{C \to D}$ from a clique C to its neighbour D along the separator $S = C \cap D$ is calculated as

$$\mu_{C \to D} = \left(\psi_C \prod_{A \in \mathrm{ne}(C) \setminus \{D\}} \mu_{A \to C} \right)^{\downarrow S},$$

where $\mathrm{ne}(C)$ are the neighbours of C in the junction tree and $\mu_{A \to C}$ are messages from A to C. Essentially, the message which C sends to D is a suitable marginalization of the product of its own potential with all the messages that C has received from its other neighbours. The absorption of messages in (15) is initially avoided and the message is just stored in its appropriate mailbox which then changes its status to being full. The procedure stops when all mailboxes are full, i.e. when exactly two messages have been sent along every branch. The Shafer–Shenoy procedure is illustrated in Figure 14.

Each of the messages $\mu_{C \to D}$ is identical to the 'R-function' in (14) which would have been calculated during a process of peeling towards a root R of the junction tree where D is located between R and C or possibly D is equal to R. Thus we have implicitly succeeded in peeling towards all cliques simultaneously.

The final step follows by observing that when all mailboxes are full, it holds that for any clique C

$$f(x_C \mid x_E^*) = \frac{\psi_C(x_C) \prod_{A \in \mathrm{ne}(C)} \mu_{A \to C}(x_{A \cap C})}{Z(x_E^*)},$$

where

$$Z(x_E^*) = f(x_E^*) = \sum_{x_C} \left(\psi_C(x_C) \prod_{A \in \mathrm{ne}(C)} \mu_{A \to C}(x_{A \cap C}) \right).$$

The HUGIN procedure The algorithm used in the software HUGIN also makes specific use of the separators but stores only a single potential ψ_S along every branch of the junction tree. Initially all these separator potentials are set to be identically equal to unity, so the factorization (12) implies that

$$f(x \mid x_E^*) \propto \frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \psi_S(x_S)}, \tag{18}$$

where $\mathcal{S}$ is the set of separators and, initially, $\psi_C = \phi_C^*$ after evidence has been entered.

When a message is sent from C to D via the separator $S = C \cap D$, the following operations are performed:

$$\begin{aligned} \psi_C^{\downarrow S}(x_C) &= \sum_{y_{C \setminus S}} \psi_C(x_S, y_{C \setminus D}) \\ \tilde{\psi}_D(x_D) &= \psi_D(x_D) \frac{\psi_C^{\downarrow S}(x_S)}{\psi_S(x_S)} \\ \tilde{\psi}_S(x_S) &= \psi_C^{\downarrow S}(x_S), \end{aligned}$$

i.e. first the *S-marginal* $\psi_C^{\downarrow S}$ of ψ_C is calculated by summing out over all variables not in S, then the clique potential ψ_D is modified by multiplication with the 'likelihood ratio' $\psi_C^{\downarrow S}/\psi_S$, and finally the separator potential ψ_S is replaced with $\psi_C^{\downarrow S}$. The potential from the clique which sends the message is unmodified, i.e. $\tilde{\psi}_C = \psi_C$. Since we have that

$$\frac{\tilde{\psi}_C(x_C)\tilde{\psi}_D(x_D)}{\tilde{\psi}_S(x_S)} = \frac{\psi_C(x_C) \left(\psi_D(x_D) \frac{\psi_C^{\downarrow S}(x_S)}{\psi_S(x_S)} \right)}{\psi_C^{\downarrow S}(x_C)} = \frac{\psi_C(x_C)\psi_D(x_D)}{\psi_S(x_S)},$$

the factorisation (18) remains valid at all times during the computational procedure.

Messages are now sent between neighbours in the tree according to a specific *schedule*. An efficient message passing schedule allows a clique to send exactly one message to each of its neighbours and only after it has already received messages from all its other neighbours. Such a message passing schedule can be implemented via a local control using the same rule as in the Shafer-Shenoy procedure. Alternatively, corresponding to the actual implementation in HUGIN, one can use a global control by first choosing a root R, then making an inward pass through the junction tree, known as

CollectEvidence, by which messages are sent from the leaves inwards towards R, and subsequently making an outward pass, DistributeEvidence, which sends messages in the reverse direction from the root towards the leaves. The two phases are illustrated in Figures 15 and 16.

The CollectEvidence procedure is identical here to what was described as 'peeling towards a root' since all separator potentials are initialised to unity. DistributeEvidence seems closer to 'reverse peeling' than the procedure used by Thompson (1981) which is more like the 'simultaneous peeling' of the Shafer-Shenoy procedure.

When exactly two messages have been sent along every branch of the junction tree in an efficient schedule, it holds that

$$f(x_A \mid x_E^*) = \psi_A(x_A)/Z(x_E^*), \qquad \text{for all } A \in \mathcal{C} \cup \mathcal{S}.$$

The marginal probability and normaliser Z can therefore be found as

$$f(x_E^*) = Z(x_E^*) = \sum_{x_S} \psi_S(x_S)$$

from any of the separator potentials ψ_S. In particular, the separator with the smallest associated state space can be chosen.

3.2 Random and other propagation schemes

Some generalizations of the message passing schemes described above use different definitions of the marginalization operation $\downarrow$ and the multiplication used in the basic factorization and message computation, but work otherwise in essentially the same fashion (Shenoy and Shafer 1990, Lauritzen and Jensen 1997). For example, replacing summation with maximization in (13) and in (14) still yields a valid propagation scheme, known as max-propagation. Then, after CollectEvidence, the (max) normalisation constant, Z, is the probability of the most probable configuration of all variables in the network, and this configuration will be identified after DistributeEvidence (Dawid 1992). Since the relation (18) remains invariant in the HUGIN procedure (also under max-propagation), one can easily switch between propagation modes.

Another important generalisation is the *random propagate* algorithm described by Dawid (1992). This begins with CollectEvidence to a root R using sum-marginalisation, i.e. with peeling to R, but in the reverse step a Monte-Carlo sample is drawn as follows. After CollectEvidence, the potential ψ_R is proportional to the conditional probability distribution of the variables in the root clique, given the evidence, cf. (17). Hence a random configuration $\check{x}_R$ can readily be sampled according to this distribution. The root clique now passes this configuration on to each of its neighbours C as $\check{x}_{R\cap C} = \check{x}_S$, where $S = C \cap R$ is the separator between C and R. After this has been done, each of the neighbouring cliques C chooses a random configuration $\check{x}_{C\setminus S}$ of the remaining variables according to a probability distribution which is proportional to $\psi_C(x_{C\setminus S}, \check{x}_S)$. When the neighbouring cliques have sampled their configurations in this way, they in turn pass on the chosen configuration to their neighbours, and so on. When the sampling stops at the leaves of the junction tree, a configuration $\check{x}$ has been correctly generated from the conditional distribution $f(x \mid x^*_E)$, given the evidence. This procedure is the general version of what Thompson (2000), page 95, describes as a variation of the Baum (1972) algorithm and forms an essential step in many Monte Carlo based computational schemes which are relevant for genetic analyses. In particular, any sampling scheme which carries out a block update on several variables jointly, conditionally on the values of the remaining variables in the network, makes use of the *random propagate* algorithm.

3.3 Computational shortcuts

Computational issues have been considered by geneticists for a long time. As a result, a number of shortcuts have been developed which speed up computations beyond the efficiency intrinsic to the local computation algorithms themselves. These shortcuts are all associated with preprocessing before the compilation and propagation steps, and have the purpose of eventually leading to a reduction of the total size of the state spaces associated with the cliques of the final junction tree. Several of these preprocessing steps are, for example, described in Sheehan (2000), Fishelson and Geiger (2002) and Heath (2003).

Below we give a brief description—from a graphical model perspective—of preprocessing steps which lead to computational savings. This may have interest also beyond genetic applications, as the savings provided are often quite considerable and therefore useful in more general cases.

Trimming One way of reducing the computational problems in a Bayesian network is to remove any unobserved *terminal* node v, i.e. an unobserved node which has no (graph) children. In genetics this is typically a phenotype node or a node representing the genotypes or alleles of a *final* individual. This removal can be made without loss of information in any Bayesian network unless the updated probabilities for the node itself are of interest. In genetic counselling problems, where the purpose of the analysis is to provide individual risk probabilities or similar cases, such nodes may be of interest in themselves and must then be kept in the network, but in many other genetic problems, say linkage analysis, they are redundant. Repeated removal of nodes of this type will eventually lead to reducing the Bayesian network to the smallest *ancestral subset* containing all nodes which are either observed or of interest. For example, if the risk probabilities of individual 13 in the network in Figure 8 were not of separate interest, and the only observed nodes were the phenotypes Y_{11} and Y_{15}, the nodes 13^p and 13^m could be removed as they are not part of the ancestral sets of the observed node. Also, if Y_{15} were neither observed nor of specific interest, all nodes corresponding to individuals 5, 8, 9, 10, 12, 13, 14, and 15 would similarly be redundant.

A related reduction can be made by removing any groups of (graph) founder nodes with a single common (graph) child, provided this child has no other (graph) parents. In general this has the associated drawback that the probability distribution of the variable associated with the child node must be calculated, but in a genetic model which implies stable gene frequencies, such computation can be avoided. In general, repeated application of this process can simplify the Bayesian network considerably. If we return to Figure 8, the alleles of individuals 1, 2, 3, 4, 5, and 8 could be removed in this fashion.

Forcing and excluding In Bayesian networks with many deterministic or close to deterministic relationships, the values at some nodes may strongly restrict the state spaces at neighbouring nodes. When a value at a given node can be determined exactly by the known values at some neighbouring nodes, we speak of *forcing* and the forced value at such a node may simply be added to the list of observations. For example, if genotypes are observable and we have that Y_{15} in Figure 8 is homozygous, the alleles 15^p and 15^m are completely identified and we may include these as observations even though they were intiially unknown.

Similarly the values at some nodes can be restricted in the sense that certain values can be *excluded* as possibilities, given the observations at neighbouring nodes. In this way the individual state space at a particular node can be reduced. Such forcings and exclusions can be repeatedly and recursively applied node by node in the network. The derivation of all globally implied forcings and exclusions in a given network is in its essence solving a constraint satisfaction problem and has the same computational complexity as the full computation itself. But it can still be very effective to derive the forcings and exclusions which follow directly from local considerations.

Delayed triangulation To take full advantage of the forcing, and to speed up computation in general, it is advantageous to incorporate the evidence into the factorization of the joint probability already at the stage of the moral graph, rather than after triangulation and setting up the junction tree. If we recall the factorisation (8)

$$f(x) = \prod_{v \in V} f(x_v \,|\, x_{\mathrm{pa}(v)}) = \prod_{C \in \mathcal{C}} \phi_C(x_C)$$

where $\mathcal{C}$ now denote the cliques of the moral graph, we may also write

$$f(x_{V\setminus E} \,|\, x_E^*) \propto \prod_{C\in\mathcal{C}} \phi_C(x_{C\setminus E}, x_{E\cap C}^*).$$

Since x_E^* is fixed, we can let $\mathcal{C}^*$ denote the set of maximal subsets among $C\setminus E, C\in\mathcal{C}$ and collect terms appropriately into revised potentials ϕ^{**} and obtain

$$f(x_{V\setminus E} \,|\, x_E^*) \propto \prod_{A\in\mathcal{C}^*} \phi_A^{**}(x_A)\ .$$

Thus, the conditional distribution factorises over the subgraph of the moral graph *induced* by the unobserved nodes $V\setminus E$. This is the graph obtained from the moral graph by removing observed nodes and all links associated with these nodes.

Because observations tend to 'break' cycles in the graph, the (optimal) triangulation of the subgraph will typically have much smaller cliques than the corresponding subgraph of the triangulated graph. Thus, with good and fast triangulation algorithms available, and with many observations and forcings, such delayed triangulation will be preferable.

In the example above where Y_{15} is observed to be homozygous so that the alleles of individual 15 can be considered observed, we can remove the associated nodes and links from the moral graph in Figure 9 thus rendering the remaining part of the graph triangulated so that no additional links are needed and all cliques have at most three nodes with associated clique size a^3.

Allele recoding If a genetic system has several alleles, but only a subset of the possible allelic types are represented in the set of observations, it may be advantageous to combine all unobserved allelic types into a single type, labelled *other*, say. This reduces the individual state space of any multiallelic system without loss of information in most genetic models. However, this allelic recoding is specific to genetic applications and does not seem to have any counterpart within general graphical models.

4 Some specific applications

In this section we will show how some specific problems in pedigree analysis can be formulated using the Bayesian network representations of Section 2.4.

In the first of these, we will consider the problem of detecting a rare recessive disease by linkage to a marker locus where both loci are assumed to be discrete. A graphical modelling approach to this particular problem has been taken by several authors (Kong 1991, Jensen and Kong 1999, Thomas, Gutin, Abkevich and Bansal 2000). Our aim here is to emphasise that the *entire* model can be specified by the graph without the need for complicated equations and derivation of the relevant joint and full conditional distributions. The second example is taken from Sheehan et al. (2002) and shows how to incorporate a continuous quantitative trait into this setting.

4.1 Simple linkage analysis

Consider the scenario where we have a disease segregating through a population with two discernible phenotypes, affected and normal. Typically, we will have some observed phenotypes for the disease and some individuals will be typed (i.e. will be of known genotype) at the marker locus. We are willing to assume that there is a single locus for the disease with two possible alleles, D and d. Let d be the 'disease' allele in that homozygous dd individuals are more likely to have the disease than other types. The model we assume for the disease penetrance is that of complete penetrance as used in Kong (1991) whereby dd individuals are affected with probability 1 and are never normal while both other genotypes are normal with probability 1 and affected with probability 0. For now, we will assume that allele frequencies for both loci are known and segregation is Mendelian so the only unknown quantities are the unobserved genotypes and phenotypes and r, the recombination fraction between the two loci. The question of interest is whether the locus for the disease is close to the known marker locus. In particular, an estimate of r is required. Founder genotype frequencies are taken to be in Hardy-Weinberg proportions, as before. Furthermore, we will assume that the founder population is in *linkage equilibrium* i.e. the frequency of any haplotype is the product of the relevant allele frequencies.

In a linkage analysis, this question is addressed by trying to assess whether or not the patterns of segregation at the disease locus are comparable with those at the marker. A tendency for the paternal gene at the disease locus to be inherited with the paternal gene at the marker, for instance, would be interpreted as evidence that the loci are not segregating independently and the strength of this dependence is related to the proximity of the loci on the chromosome, or the *tightness* of the linkage between them. Formally, we calculate the likelihood $L(r)$ for the disease and marker observations as a function of the recombination fraction over a grid of values of r. Note that the peeling algorithm is fully sufficient in this case as only the likelihood is required. When there is a large amount of missing information, and particularly on a large looped pedigree, the necessary summation over all configurations of unobserved variables consistent with the data can be intractable and approximate methods must be used (Heath 2003).

To test the hypothesis of 'no linkage' the ratio $L(r)/L(\frac{1}{2})$ is calculated and maximised over $r \in [0, \frac{1}{2}]$. Recall that $r = \frac{1}{2}$ means that there is no linkage and the loci are segregating independently whereas $r \approx 0$ indicates tight linkage. Traditionally, it is the standardised log likelihood ratio

$$\log_{10}\left[\frac{L(r)}{L(\frac{1}{2})}\right]$$

which is maximised and this is known as the *LOD score* (Morton 1955, Ott 1999). The maximising value $\hat{r}$ will provide us with an estimate of the recombination fraction and hence, the location of the disease locus.

As an alternative one may take a Bayesian approach which in addition incorporates uncertainty on gene frequencies. Then *random propagation* becomes an important part of a block updating MCMC algorithm as in the more complex problem of locating a major gene for a quantitative trait discussed below in Section 4.2, and also the problem associated with genetic modelling of the fur colour of foxes discussed in Section 4.3.1.

Following Sheehan et al. (2002), we now show how to construct a graphical model for the two-locus linkage problem by focusing on a nuclear family comprising a father 1, mother 2 and their offspring 3. This construction is of course replicated for all parent-child triplets in the pedigree but it is far too complicated visually to consider more than a few individuals at a time. As we need haplotype information for linkage in order to detect recombinations, it is most convenient to use the segregation network for this representation. Beginning with the marker locus—the 'α-locus'— for each parent $i = 1, 2$, we create the two nodes, $i^{1\alpha}$ and $i^{0\alpha}$, for the paternal and maternal genes of the individual with values assigned at random according to the marker allele frequencies. Note that this random assignment immediately deals with the fact that phase is unknown in the parents and we have to sum over all possibilities. We can assume that

$$P(S^{\alpha}_{1,3} = 1) = P(S^{\alpha}_{1,3} = 0) = 1/2 \tag{19}$$

by Mendelian inheritance at this first locus in the absence of information at the linked locus. As in Section 2.4, the node for the paternally inherited allele of the offspring, $3^{1\alpha}$, is a (graph) child of both alleles in the father, $1^{1\alpha}$ and $1^{0\alpha}$, and of $S^{\alpha}_{1,3}$. Similarly, $S^{\alpha}_{2,3}$ describing the segregation from 2 to 3 has a Bernoulli distribution with probability $\frac{1}{2}$ and the maternally inherited gene in 3 is a child of this node and of both genes in the mother. Next, we add genotype nodes to the graph for each individual. As the genotype is fully determined by the allelic types of the individual's genes at the locus, it is a (graph) child of both gene nodes. Figure 17 shows the corresponding network for one locus in our nuclear family.

Now consider the disease locus where the frequency of the disease allele d, is f_d. If we label this locus as δ, we now extend our graph by adding two nodes for each of the parents exactly as before with values determined by a Bernoulli distribution with parameter f_d. These are labelled $i^{1\delta}$ and $i^{0\delta}$ for $i = 1, 2$ in Figure 18. The unobserved genotype, G^{δ}_i, is represented as a child node of the corresponding gene nodes. It is also a (graph) parent of the observable phenotype, Y_i^δ with link specified by the penetrance function. For the offspring, 3, we have gene nodes and a segregation indicator exactly as for the marker locus with the difference now being that we must take account of linkage between the loci. In particular, the value of the segregation indicators $S_{1,3}^\delta$ and $S_{1,3}^\alpha$ are dependent via the recombination fraction, r. This dependence can be modelled with an undirected link between the corresponding nodes. Formally this would lead to a chain graph representation (Lauritzen 1996) rather than a DAG. However, for the sake of exposition, we here use the equivalent non-symmetric description through the conditional distribution of $S_{1,3}^\delta$ given $S_{1,3}^\alpha$, specifically:

$$S_{1,3}^\delta \sim \begin{cases} Ber(1-r) & \text{if } S_{1,3}^\alpha = 1 \\ Ber(r) & \text{if } S_{1,3}^\alpha = 0, \end{cases} \tag{20}$$

and similarly for $S_{2,3}^\delta$. To complete the graph in Figure 18 we now add nodes G_3^δ and Y_3^δ for the offspring's unobserved genotype and phenotype with links defined exactly as above.

Note that this is a full specification of the model similar to that described in Kong (1991). Further derivation of the relevant joint and marginal distributions are not necessary as these are a direct result of the factorisation (1) in Section 2.3. We note that in principle, this model could have been represented in terms of either an allele or a genotype network. Compared with these we have complicated the problem visually by adding extra variables but, as pointed out by Kong (1991), such visual complications can often reduce computational complexity since each variable now has fewer possible values. The graph that has most appeal to the human eye is not necessarily the best graph on which to perform the required calculations. Rather, the most disaggregated graph possible tends to exploit factorisations better, leading to more efficient computation (Section 3).

4.2 Detection of a quantitative trait locus

The discussion so far has focused on genetic traits with a phenotype which is fully determined by the genotype at a single locus, or *Mendelian* traits. Sheehan et al. (2002) extend the linkage scenario described above to the problem of detecting a *quantitative trait locus* (QTL) from possibly incomplete marker data and begin with the trivial example involving two flanking loci. Recall (Section 2.2) that the phenotype of a quantitative trait is held to derive from the segregation of many genes at many loci and may also have a non-genetic component. In principle, if individuals are scored for their genotypes at a marker locus and phenotypes for the quantitative trait, differences in mean records for the trait among different classes of marker genotype would provide evidence for a QTL close to the marker.

In this application, two markers are considered with known map positions (and hence known recombination fraction between them) and it is hypothesised that there is a diallelic QTL somewhere between the two. The trait of interest is any trait measured on a continuum having an associated polygenic effect unlinked to the QTL such as milk yield in dairy cattle, for example. Marker data are available on a *half-sib* design comprising several families each with a single sire and up to 100 offspring. Trait data are only available on the offspring. By contrast with the two-locus linkage example above, no information is given on the mothers (dams) of these offspring and hence the maternal segregations are all ignored. This is common in animal breeding applications where data are habitually collected on designs which can be handled by simple least squares or likelihood methods.

The phenotype record on offspring j of sire i is a realisation of Y_{ij}. The effect of the unobserved genotype at the QTL is q_{ij} where q_{ij} can have three possible values μ_1, μ_2, μ_3 corresponding to each of the three genotypes. A normal linear mixed model for the data is

$$Y_{ij} = Z_i + q_{ij} + E_{ij},$$

where Z_i represents the average additive genetic effect of the i^{th} sire on the phenotypes of his offspring and which cannot be explained by the QTL. Let σ_a^2 be the total additive genetic variance unexplained by the QTL and σ_e^2 be the environmental variance. We have that $Z_i \sim N(0, \sigma_z^2)\ \forall\ i$ where σ_z^2 is the sire variance component (Falconer and Mackay 1996) and $\sigma_z^2 = \frac{1}{4}\sigma_a^2$ since half the genes of an offspring are shared with its sire. The remaining unexplained variation is picked up by the residual term, $E_{ij} \sim N(0, \frac{3}{4}\sigma_a^2 + \sigma_e^2)$. Estimates of the recombination fractions between the QTL and each of the markers are required and estimates of the QTL effects, μ_1, μ_2, μ_3, are also of interest. Assuming no genetic interference (Section 2.2) only one of the unknown recombination fractions, or equivalently the QTL map location λ_Q, is necessary.

Figure 19 shows the graphical model for this trivial QTL mapping problem for one sire and two offspring. The marker loci are labelled α and β while the QTL locus is now δ. The model is essentially an extension of the two-locus linkage problem in Figure 18 to a three-locus problem. Gene nodes are added for the third locus in an analogous fashion with the segregation indicator for inheritance from the sire linked to the previous value via the recombination fraction between the second and third loci, just as described in (20) above. This assumes that there is no genetic interference and that recombinations in adjacent intervals are independent.

The main difference is that the offsprings' maternal genes are assumed to be randomly drawn from the population as there is no phase information on the dams. Non-relatedness of the dams is a fundamental assumption of the half-sib design. Sire and offspring have marker genotype nodes while only offspring have trait genotype and phenotype nodes. Finally, covariance between the offspring is reflected in the genes they share with their sire and they are duly connected by the sire effect node. Note that this creates a cycle in the graph which becomes increasingly complex computationally when more offspring are added and for a typical half-sib design with a sire having up to 100 offspring, the relevant Bayesian network for this problem features many long cycles despite the simplicity of the pedigree structure and the model under consideration (see Section 3).

We conclude this section by noting that these models lend themselves readily to Bayesian analysis and interpretation (Spiegelhalter 1998) where all unknown quantities can be regarded as random variables. Thus, data, latent variables and model parameters can all be represented as nodes in the graph with associated distributions. Lund and Jensen (1999) use graphical models for a Bayesian formulation of a mixed inheritance model, for example, and Sheehan et al. (2002) extend the model in Figure 19 to a full Bayesian analysis for the QTL mapping problem. In all of these, *random propagation* is used as an essential part of the associated MCMC computational procedure.

4.3 Beyond pedigrees

The flexibility of a graphical modelling approach to applications in genetics is powerfully demonstrated by examples where the pedigree is not fixed and known, or when other circumstances should be integrated into the analysis. We briefly discuss some examples of this.

4.3.1 Fur colour of foxes

Hansen and Pedersen (1994) elegantly handle incomplete paternity information in a two-locus inheritance model for fur colour in foxes from pedigrees supplied by Scandinavian fur farms. A more standard analysis of the same data appears in Skjøth, Lohi and Thomas (1994). There are four fox pedigrees in this dataset, comprising 253 animals in total, all born between 1975 and 1982. Although there are some loops, the pedigrees are generally small enough to enable exact likelihood calculations for the simple models considered (Skjøth et al. 1994). However, there is uncertainty with some of the litter paternities. In many cases, a female is mated with two males in order to increase the chances of fertilisation. Depending on the time between these matings, it is not always possible to determine which male actually fathered the resulting pups. Indeed, with this breeding practice, two males could father pups in the same litter. However, for the pedigree, the farmer states the most likely candidate as the father (to all pups) and registers the second sire as an alternative whenever there is doubt.

The phenotypic record on each fox is a subjective classification of fur colour with animals classified by different observers and at different ages. Modes of inheritance of genes for fur colour have been studied for many animals, especially mice (Silvers 1979). Based on homology with mice and sheep, Adalsteinsson, Hersteinsson and Gunnarsson (1987) proposed a genetic model involving two diallelic loci, α and ε, possibly on the same chromosome. Labelling the alleles at the '$\alpha-$locus' as A and a and those at the '$\varepsilon-$locus' as E and e, the relationship between fur colour phenotype and corresponding two-locus genotype suggested by this model is given in Table 4, taken from Hansen and Pedersen (1994).

The animals are also classified on a scale from $1-10$, increasing numbers reflecting increasing amounts of black colour over red. This classification is again subjective and not always consistent with the one based on overall coat colour. Hansen and Pedersen (1994) reduce this to a scale of $1-8$. The structure of the penetrance matrix they use for the nine possible genotypes and these eight phenotypes is shown in Table 5.

The usual method for handling paternity uncertainty in a likelihood setting is to compare the likelihoods for all possible pedigrees (Thompson 1986). In this case, there is only one alternative father for each of a small number of litters but as each pup in the litter could have been fathered by either of the two candidates, a likelihood comparison for this particular problem would require the consideration of 2^{21} pedigrees. Skjøth et al. (1994) circumvent this problem by estimating paternal genotypes from the phenotypic information and choosing the most likely individual. Standard statistical genetics programs will not accept a pedigree where an individual can have more than one biological father. However, a graphical model does not distinguish between biological and graph parents and Hansen and Pedersen (1994) exploit this by defining a binary node, W_i, to indicate the paternity of i. Specifically,

$$W_i = \begin{cases} 1 & \text{if stated father is the true father} \\ 0 & \text{if alternative father is the true father} \end{cases}$$

and $W_i \sim Ber(p_w)$ where p_w is to be estimated.

We now build the graphical model for the fox data, following Hansen and Pedersen (1994), but simplifying the model slightly by assuming a known penetrance matrix and known allele frequencies at both loci. The recombination fraction, r, between the two loci is unknown in this example. We consider four animals: the mother, 1, putative father, 2, offspring, 3 and alternative father, 4. Using the same notation as before, we assign paternal and maternal alleles at both loci, $i^{1\alpha}, i^{0\alpha}, i^{1\varepsilon}, i^{0\varepsilon}$ from the relevant Bernoulli distributions to the founders, $i = 1, 2$ and 4. Segregation from mother to offspring can be quantified by indicators for each locus, $S^{\alpha}_{1,3}, S^{\varepsilon}_{1,3}$, and corresponding indicators for paternal inheritance will be denoted as $S^{\alpha}_{f_3,3}, S^{\varepsilon}_{f_3,3}$

where $f_3 = 2$ if $W_3 = 1$ and $f_3 = 4$, otherwise. In contrast with our representation of Section 4, where the segregation indicator at the first locus is given a Bernoulli(1/2) distribution to reflect Mendelian segregation and the second indicator depends on the value of this via the recombination fraction, r, Hansen and Pedersen (1994) consider segregation at both loci jointly. We will represent their joint segregation, or *phase* indicator for maternal inheritance by $S_{1,3}$ where

$$S_{1,3} = \begin{cases} (0,0) & \text{with probability} \quad (1-r)/2 \\ (0,1) & \text{with probability} \quad r/2 \\ (1,0) & \text{with probability} \quad r/2 \\ (1,1) & \text{with probability} \quad (1-r)/2 \end{cases}.$$

Paternal inheritance is represented analogously and designated by by $S_{f_3,3}$. Note that $S_{1,3} = (S^{\alpha}_{1,3}, S^{\varepsilon}_{1,3})$ in our notation and that this is the same model as described in (19) and (20) above. The latter parameterisation, involving more nodes with fewer states, is more flexible when considering more than two loci and is generally better for computational purposes. All four alleles of individual i are graph parents of the node G_i representing the two-locus genotype and Y_i denotes the fur colour phenotype. Figure 20 shows the corresponding graphical model for this problem.

Despite the fact that we have omitted nodes for the allele frequencies and parameters of the penetrance matrix, the graph in Figure 20 is more complicated than those shown earlier in this section in that it has many more loops. The advantage, however, is that questions about paternity, genetic inheritance and linkage can all be addressed from this one graph.

Hansen and Pedersen (1994) use a Bayesian approach with prior Dirichlet (or Beta) distributions on unknown parameters and an alternating blocking Gibbs sampler to carry out their analysis. In one step of the sampler, all unobserved nodes in the network are imputed by *random propagation* as described in Section 3.2 using fixed and current values of the parameters. In the other step, all parameters of the model are sampled using conjugate updating of Dirchlet-distributions, conditional on complete observed and imputed data.

4.3.2 Forensic genetics

The ease with which graphical models, or *probabilistic expert systems*, can be adapted to handle a wide range of routine problems in forensic inference is illustrated clearly by Dawid, Mortera, Pascali and van Boxel (2002). Here, the focus is on the general problem of inferring the identity of an individual based on the given evidence which may include DNA profile information. In principle, this can be done (Dawid and Mortera 1996) by calculating relative likelihoods for the various competing hypotheses. Such calculations can become computationally intensive however, when information is either imperfect or missing altogether (Dawid and Mortera 1998), or especially when the possibility of observing a mutation from one generation to the next is entertained (Dawid, Mortera and Pascali 2001) and efficient computational algorithms on good representations are required.

Paternity problems As an example, consider an inheritance claim case taken from Dawid et al. (2002) which can formally be expressed as a case of disputed paternity and is represented in pedigree form in Figure 21. A man whom we shall label as 9 (the disputed child) claims to be the son of the diseased individual 3 (the putative father) and hence entitled to part of his estate. We know that 3 has an (undisputed) child, 6, and we know that the two children in question have different mothers. There is no DNA information on the putative father since he is dead and buried, nor on either of the two mothers, but we have DNA profile samples from both children and the man's brother, 4. As is often the case with such applications, we focus attention on just two competing hypotheses: either the true father, 8, of the disputed child, 9, really is one and the same as the putative father, 3, *or* the true father can be considered as randomly drawn from the general population.

It is usually assumed that the markers used in forensic inference are unlinked, as they are either on different chromosomes or else so far apart on the same chromosome that linkage is negligible. Hence, we need only consider the model for calculating the likelihood of interest at any single marker and the overall likelihood will be the product taken across all markers. Figure 22 shows the representation used by Dawid et al. (2002) for a single forensic marker. They use the allele network rather than the segregation network, but as noted previously, the latter is superfluous in the absence of linkage. Our notation is as before, but we omit the marker labels for simplicity. Thus, i^0 and i^1 represent the random variables assigning maternal and paternal genes of individual i and G_i assigns the genotype of i at the marker. Note that untyped individuals who are not directly of interest (i.e. $1, 2, 5$ and 7), are only represented by the genes they contribute to the next generation which, in the absence of any information, are assumed to be randomly drawn from the population.

One of the interesting features of the graph in Figure 22 is the black node referred to as the 'query' or 'target' node by Dawid et al. (2002) and which is a (graph) parent of both genes in individual 8. This is a binary node taking the value 1 if the true father of the disputed child is the putative father i.e. if individual 8 is the same as 3 in Figure 21. In this case, the two genes in 8 are copies of the corresponding two in 3. Otherwise, the men are different individuals and the genes of 8 are drawn randomly from the population. The advantage of using the query node is that the quantity of interest to the courtcase—the likelihood ratio in favour of paternity—can be read off directly. Note that this node is essentially the same as the paternity indicator, W_i, of Hansen and Pedersen (1994) which determines which of the two possible alternatives fathered individual i (Figure 20). The emphasis in the fox analysis was, however, somewhat different.

Complex identification problems Related problems which extend the traditional paternity question to the consideration of more than two alternatives are those of establishing family relationships in immigration cases, and of identifying multiple remains from disasters like airplane crashes, wars and fires. In particular, it may be important to identify members of the same family as, for example, in the famous case of the executed Romanov family believed to have been found in Yekaterinburg (Gill, Ivanov, Kimpton, Piercy, Benson, Tully, Evett, Hagelberg and Sullivan 1994). Here, interest is centered on finding the most probable pedigree given the DNA evidence from the bodies buried in the grave as well as from a few known living individuals, in this case individuals known to have a specific genetic relationship to members of the Romanov family.

All possibilities ranging from totally unrelated individuals to all individuals belonging to the same connected pedigree, including all the extra people required to define the necessary relationships, must be entertained. Sometimes the set of possible pedigrees is too large for an exact approach. Egeland, Mostad, Mervåg and Stenersen (2000) consider this problem by selecting a subset of probable pedigrees. Certain structures are eliminated according to various criteria which may vary from case to case. For example, if sex and age data are available, some individuals will not be allowed to feature as parents in any pedigree structure and limits on incestuous relationships and numbers of marriages and offspring may also be applied. A prior probability distribution is then imposed on the selected set of pedigrees and the program FAMILIAS used to combine this with the relevant likelihood information to deliver posterior probabilities for the various structures considered. To date, this problem has not been tackled systematically with graphical models but, as noted by Dawid et al. (2002), it is an important application for consideration.

In addition to the problems mentioned, the area of forensic genetics yields a variety of problems where the flexibility and modularity of the graphical model approach can be fruitfully exploited. For example, in criminal cases it is not uncommon to observe DNA evidence which represents a *mixture* of phenotypes of an unknown number of individuals. Such cases can readily be accomodated within graphical models (Mortera, Dawid and Lauritzen 2003).

5 Perspective

We have hopefully demonstrated that graphical models can be used to formulate and analyse many problems in genetics, or problems having a strong genetic component, especially when potentially complex family relations must be taken into account. The advantages of phrasing these problems in the language of graphical models derive from the flexibility with which standard problems can be modified to accommodate special situations, whether they be observational schemes, types of problem under consideration, or other external circumstances that need to be incorporated into the basic genetic models. Moreover, this entirely general approach facilitates the accessibility of these problems in genetics together with their associated computational and statistical methods to a wider and less specialised community. In this perspective, analyses of complex genetic data can benefit enormously from current rapid developments in the area of general graphical models, thereby extending the domain in which these methods can be most usefully exploited.

Acknowledgments

The authors are indebted to Klitgaarden Refugium, a refuge for artists and scientists in Skagen, Denmark, where much of this paper was written. We also acknowledge research support from Leverhulme Research Interchange Grant F/07134/K, Wellcome Trust Biomedical Research Collaboration Grant 056266/Z/98/Z and the TVW Telethon Institute for Child Health Research, Perth, Western Australia.

References

Adalsteinsson, S., Hersteinsson, P. and Gunnarsson, E.: 1987, Fox colors in relation to colors in mice and sheep, *Journal of Heredity* **78**, 235–237.

Amestoy, P. R., Davis, T. A. and Duff, I. S.: 1996, An approximate minimum degree ordering algorithm, *SIAM Journal of Matrix Analysis and its Applications* **17**, 886–905.

Andersen, S. K., Olesen, K. G., Jensen, F. V. and Jensen, F.: 1989, HUGIN—a shell for building Bayesian belief universes for expert systems, *Proceedings of the 11th International Joint Conference on Artifical Intelligence*, Morgan Kaufmann, San Mateo, pp. 1080–1085.

Baum, L. E.: 1972, An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, *Inequalities* **3**, 1–8.

Berry, A., Bordat, J.-P. and Cogis, O.: 2000, Generating all the minimal separators of a graph, *International Journal of Foundations of Computer Science* **11**, 397–403.

Bouchitté, V. and Todinca, I.: 2001, Treewidth and minimum fill-in: Grouping the minimal separators, *SIAM Journal on Computing* **31**, 212–232.

Cannings, C., Thompson, E. A. and Skolnick, M. H.: 1978, Probability functions on complex pedigrees, *Advances in Applied Probability* **10**, 26–61.

Cottigham, R. W., Idury, R. M. and Schäffer, A. A.: 1993, Faster sequential genetic linkage computations, *American Journal of Human Genetics* **53**, 252–263.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. and Spiegelhalter, D. J.: 1999, *Probabilistic Networks and Expert Systems*, Statistics for Engineering and Information Science, Springer-Verlag, New York, Inc.

Dawid, A. P.: 1992, Applications of a general propagation algorithm for probabilistic expert systems, *Statistics and Computing* **2**, 25–36.

Dawid, A. P. and Mortera, J.: 1996, Coherent analysis of forensic identification evidence, *Journal of the Royal Statistical Society, Series B* **58**, 425–443.

Dawid, A. P. and Mortera, J.: 1998, Forensic identification with imperfect evidence, *Biometrika* **85**, 835–849.

Dawid, A. P., Mortera, J. and Pascali, V. L.: 2001, Non-fatherhood or mutation? A probabilistic approach to parental exclusion in paternity testing, *Forensic Science International* **124**, 55–61.

Dawid, A. P., Mortera, J., Pascali, V. L. and van Boxel, D.: 2002, Probabilistic expert systems for forensic infererence from genetic markers, *Scandinavian Journal of Statistics* **29**, 577–595.

Egeland, T., Mostad, P. F., Mervåg, B. and Stenersen, M.: 2000, Beyond traditional paternity and identification cases. Selecting the most probable pedigree, *Forensic Science International* **110**, 47–59.

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

Falconer, D. S. and Mackay, T. F. C.: 1996, *Introduction to Quantitative Genetics*, 4th edn, Longman Group Ltd.

Fernandez, S. A., Fernando, R. L., Guldbrandtsen, B., Totir, R. and Carriquiry, A. L.: 2001, Sampling genotypes in large pedigrees with loops, *Genetics Selection Evolution* **33**, 337–367.

Fishelson, M. and Geiger, D.: 2002, Exact genetic linkage computations for general pedigrees, *Bioinformatics* **18**, S189–S198.

George, A. and Liu, J. W. H.: 1989, The evolution of the minimum degree ordering algorithm, *SIAM Review* **31**, 1–19.

Gill, P. E., Ivanov, P. L., Kimpton, C., Piercy, R., Benson, N., Tully, G., Evett, I., Hagelberg, E. and Sullivan, K.: 1994, Identification of the remains of the Romanov family by DNA analysis, *Nature Genetics* **6**, 130–135.

Haldane, J. B. S.: 1919, The combination of linkage values and the calculation of distances between the loci of linked factors, *Journal of Genetics* **8**, 229–309.

Hansen, B. and Pedersen, C. B.: 1994, Analysing complex pedigrees using Gibbs sampling. A theoretical and empirical investigation, *Technical Report R-94-2032*, Institute for Electronic Systems, Aalborg University.

Heath, S. C.: 2003, Genetic linkage analysis using Markov chain Monte Carlo techniques, *in* P. J. Green, N. L. Hjort and S. Richardson (eds), *Highly Structured Stochastic Systems*, Oxford University Press, Oxford, UK, pp. 363–381.

Jensen, C. S.: 1997, *Blocking Gibbs Sampling for Inference in Large and Complex Bayesian Networks with Applications in Genetics*, PhD thesis, Aalborg University, Denmark.

Jensen, C. S., Kjærulff, U. and Kong, A.: 1995, Blocking Gibbs sampling in very large probabilistic expert systems, *International Journal of Human-Computer Studies* **42**, 647–666.

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

Jensen, F.: 2002, *HUGIN API Reference Manual Version 5.4*, HUGIN Expert Ltd., Aalborg, Denmark.

Jensen, F. V.: 1996, *An Introduction to Bayesian Networks*, University College London Press, London, United Kingdom.

Jensen, F. V., Lauritzen, S. L. and Olesen, K. G.: 1990, Bayesian updating in causal probabilistic networks by local computation, *Computational Statistics Quarterly* **4**, 269–282.

Kjærulff, U.: 1992, Optimal decomposition of probabilistic networks by simulated annealing, *Statistics and Computing* **2**, 7–17.

Kong, A.: 1991, Efficient methods for computing linkage likelihoods of recessive diseases in inbred pedigrees, *Genetic Epidemiology* **8**, 81–103.

Kruglyak, L., Daly, M. J., Reeve-Daly, M. P. and Lander, E. S.: 1996, Parametric and nonparametric linkage analysis: a unified multipoint approach, *American Journal of Human Genetics* **58**, 1347–1363.

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

Lander, E. S. and Schork, N. J.: 1994, Genetic dissection of complex traits, *Science* **265**, 2037–2048.

Lange, K. and Elston, R. C.: 1975, Extensions to Pedigree Analysis. I. Likelihood calculations for simple and complex pedigrees, *Human Heredity* **25**, 95–105.

Lauritzen, S. L.: 1996, *Graphical Models*, Clarendon Press, Oxford, United Kingdom.

Lauritzen, S. L.: 2001, Causal inference from graphical models, *in* O. E. Barndorff Nielsen, D. R. Cox and C. Klüppelberg (eds), *Complex Stochastic Systems*, Chapman and Hall/CRC Press, London/Boca Raton, pp. 63–107.

Lauritzen, S. L. and Jensen, F. V.: 1997, Local computation with valuations from a commutative semigroup, *Annals of Mathematics and Artificial Intelligence* **21**, 51–69.

Lauritzen, S. L. and Spiegelhalter, D. J.: 1988, Local computations with probabilities on graphical structures and their application to expert systems (with discussion), *Journal of the Royal Statistical Society, Series B* **50**, 157–224.

Lund, M. S. and Jensen, C. S.: 1999, Blocking Gibbs sampling in the mixed inheritance model using graph theory, *Genetics, Selection, Evolution* **31**, 3–24.

Mendel, G.: 1866, Experiments in plant hybridisation, *in* J. H. Bennett (ed.), *English Translation and Commentary by R A Fisher*, Oliver and Boyd, Edinburgh 1963.

Monaco, A. P., Bertelson, C. J., Middlesworth, W., Colletti, C. A., Aldridge, J., Fischbeck, K. H., Bartlett, R., Pericak-Vance, M. A., Roses, A. D. and Kunkel, L. M.: 1985, Detection of deletions spanning the duchenne muscular dystrophy locus using a tightly linked DNA segment, *Nature* **316**, 842–845.

Mortera, J., Dawid, A. P. and Lauritzen, S. L.: 2003, Probabilistic expert systems for DNA mixture profiling, *Theoretical Population Biology* **63**, 191–205.

Morton, N. E.: 1955, Sequential tests for the detection of linkage, *American Journal of Human Genetics* **7**, 277–318.

O'Connell, J. R.: 2001, Rapid multipoint linkage analysis via inheritance vectors in the Elston–Stewart algorithm, *Human Heredity* **51**, 226–240.

Ott, J.: 1999, *Analysis of Human Genetic Linkage*, third edn, The Johns Hopkins University Press, Baltimore.

Pearl, J.: 1986, Fusion, propagation and structuring in belief networks, *Artificial Intelligence* **29**, 241–288.

Pearl, J.: 1988, *Probabilistic Inference in Intelligent Systems*, Morgan Kaufmann, San Mateo, California.

Riordan, J. R., Rommens, J. M., Kerem, B. S., Alon, N., Rozmahel, R., Grzelczak, Z., Zielenski, J., Lok, S., Plavsic, N., Chou, J. L., Drumm, M. L., Iannuzzi, M. C., Collins, F. S. and Tsui, L. C.: 1989, Identification of the cystic fibrosis gene: cloning and characterization of complimentary DNA, *Science* **245**, 1066–1073.

Sham, P.: 1998, *Statistics in Human Genetics*, Applications of Statistics Series, Arnold, London.

Sheehan, N. A.: 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., Guldbrandtsen, B., Lund, M. S. and Sorensen, D. A.: 2002, Bayesian MCMC mapping of quantitative trait loci in a half-sib design: a graphical model perspective, *International Statistical Review* **70**, 241–267.

Shenoy, P. P. and Shafer, G.: 1990, Axioms for probability and belief–function propagation, *in* R. D. Shachter, T. S. Levitt, L. N. Kanal and J. F. Lemmer (eds), *Uncertainty in Artificial Intelligence 4*, North-Holland, Amsterdam, The Netherlands, pp. 169–198.

Shoiket, K. and Geiger, D.: 1997, A practical algorithm for finding optimal triangulations, *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, California, pp. 185–190.

Silvers, W. K.: 1979, *The Coat Colors of Mice*, Springer-Verlag, New York.

Skjøth, F., Lohi, O. and Thomas, A. W.: 1994, Genetic models for the inheritance of the silver mutation of foxes, *Genetical Research* **64**, 11–18.

Sobel, E. and Lange, K.: 1996, Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics, *American Journal of Human Genetics* **58**, 1323–1337.

Spiegelhalter, D. J.: 1990, Fast algorithms for probabilistic reasoning in influence diagrams, with applications in genetics and expert systems, *in* R. M. Oliver and J. Q. Smith (eds), *Influence Diagrams, Belief Nets and Decision Analysis*, John Wiley and Sons, Chichester, United Kingdom, pp. 361–384.

Spiegelhalter, D. J.: 1998, Bayesian graphical modelling: a case-study in monitoring health outcomes, *Applied Statistics* **47**, 115–133.

Thomas, A.: 1985, *Data structures, methods of approximation and optimal computation for pedigree analysis*, PhD thesis, Cambridge University.

Thomas, A., Gutin, A., Abkevich, V. and Bansal, A.: 2000, Multilocus linkage analysis by blocked Gibbs sampling, *Statistics and Computing* **10**, 259–269.

Thompson, E. A.: 1981, Pedigree analysis of Hodgkin's disease in a newfoundland genealogy, *Annals of Human Genetics* **45**, 279–292.

Thompson, E. A.: 1986, *Pedigree Analysis in Human Genetics*, The Johns Hopkins University Press, Baltimore.

Thompson, E. A.: 1994, Monte Carlo likelihood in genetic mapping, *Statistical Science* **9**(3), 355–366.

Thompson, E. A.: 2000, *Statistical Inference from Genetic Data on Pedigrees*, Vol. 6 of *NSF-CBMS regional Conference Series in Probability and Statistics*, Institute of Mathematical Statistics and the American Statistical Association.

Thompson, E. A.: 2001, Monte Carlo methods on genetic structures, *in* O. E. Barndorff-Nielsen, D. R. Cox and C. Klüppelberg (eds), *Complex Stochastic Systems*, Chapman & Hall/CRC Press, London/Boca Raton, chapter 4, pp. 176–218.

Thompson, E. A. and Heath, S. C.: 2000, Estimation of conditional multilocus gene identity among relatives, *in* F. Seiller-Moiseiwitsch (ed.), *Statistics in molecular biology and genetics*, IMS Lecture Notes. Institute of Mathematical Statistics. American Mathematical Society, pp. 95–113.

Thompson, E. A. and Wijsman, E. M.: 1990, The Gibbs sampler on extended pedigrees: Monte Carlo methods for the genetic analysis of complex traits, *Technical Report 193*, Department of Statistics, University of Washington, Seattle, Washington, USA.

Yannakakis, M.: 1981, Computing the minimum fill-in is NP-complete, *SIAM Journal on Algebraic and Discrete Methods* **2**, 77–79.

Figures and tables

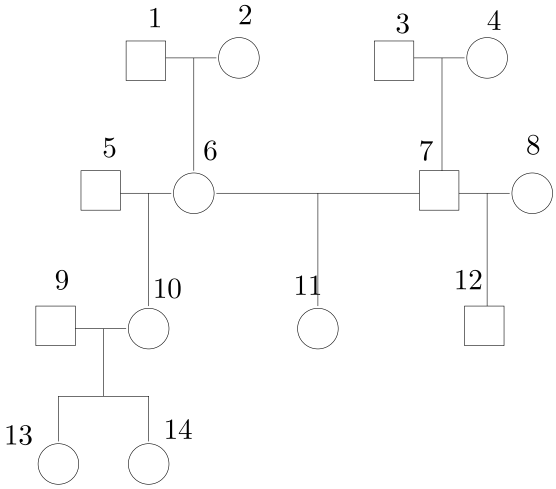

FIGURE 1: A standard graphical representation of a simple pedigree of 14 individuals. As is consistent with common usage, females are represented by circles and males by squares. Individuals $1, 2, 3$ and 4 are the baseline founders, while $5, 8$ and 9 are recent founders who have married in. Individuals $11, 12, 13$ and 14 are *finals* in that they have no marriages.

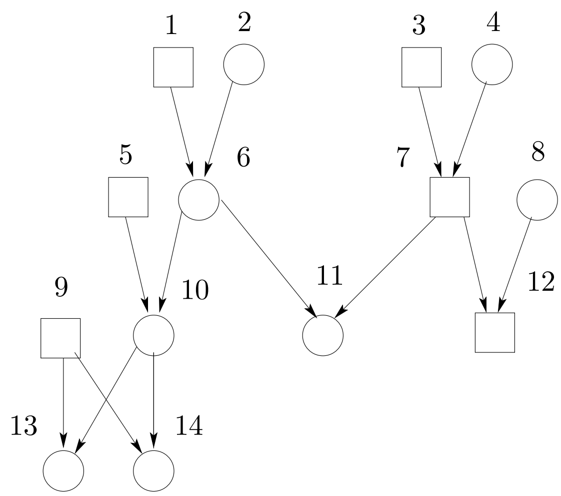

FIGURE 2: The pedigree of Figure 1 drawn as a relationship graph with nodes representing individuals and directed edges connecting individuals to their offspring.

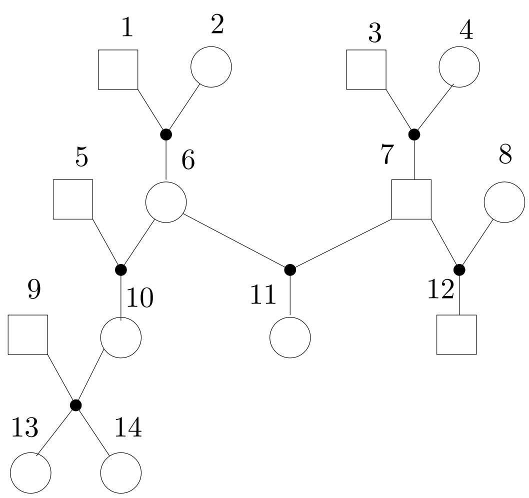

FIGURE 3: The pedigree in Figure 1 represented as a marriage node graph with edge directions omitted.

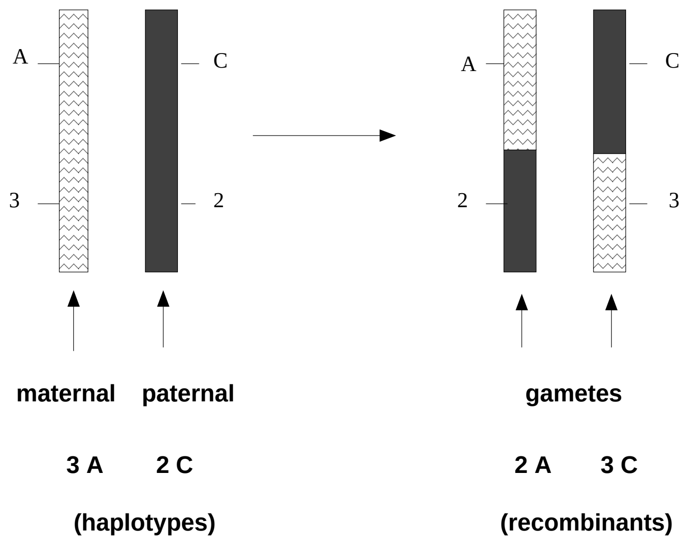

FIGURE 4: A schematic representation of meiosis adapted from Heath (2003) showing the chromosomes which form the gametes containing some maternal and some paternal DNA after crossing over has occurred.

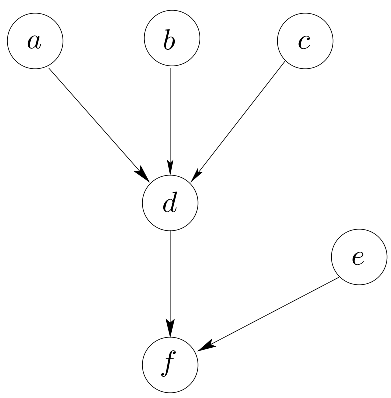

FIGURE 5: A simple Bayesian network with nodes a, b and c all parents of d while f is a child node of both d and e. Note that if nothing is known about d besides what can be inferred from its parents, then a, b and c are all independent. Conditional dependencies between a, b and c are imposed, however, if information on f influences the certainty of d.

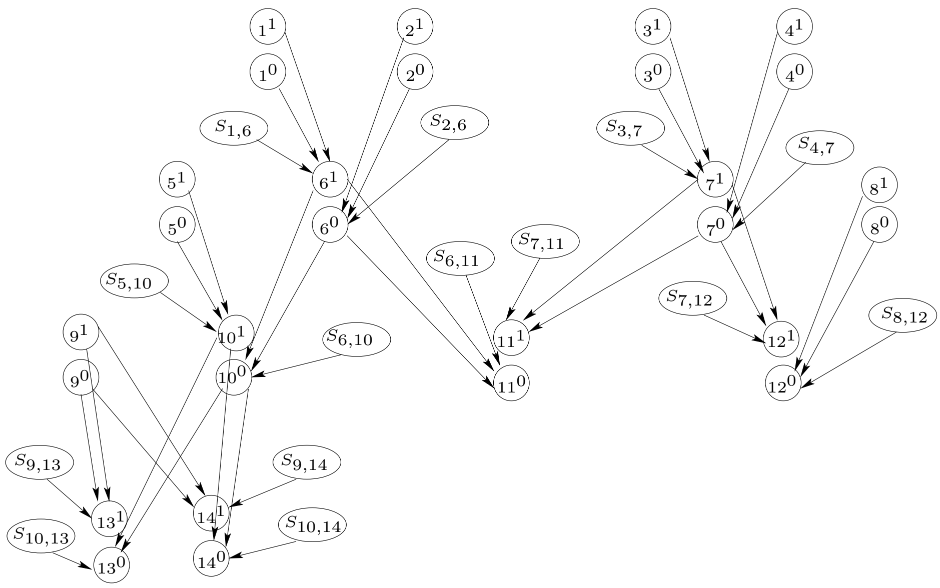

FIGURE 6: The segregation network for our simple pedigree of Figure 1.

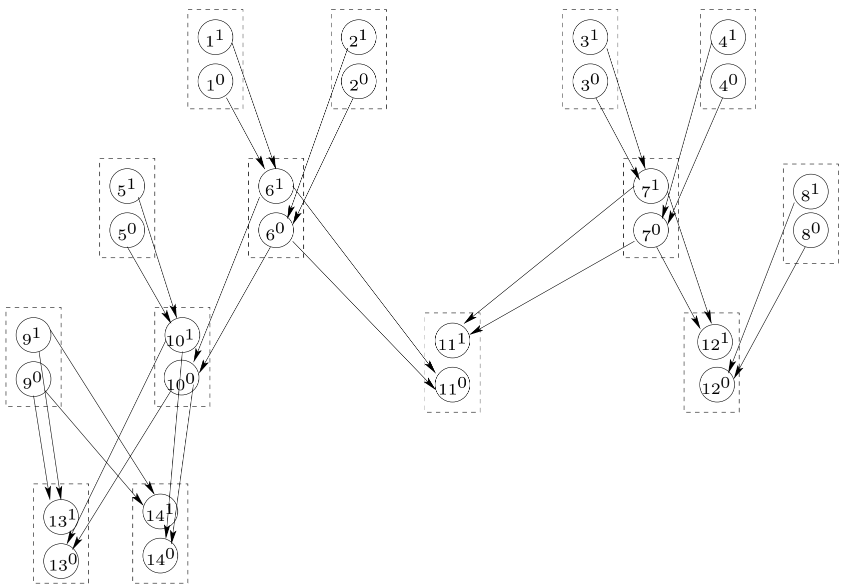

FIGURE 7: The allele network for our simple pedigree of Figure 1.

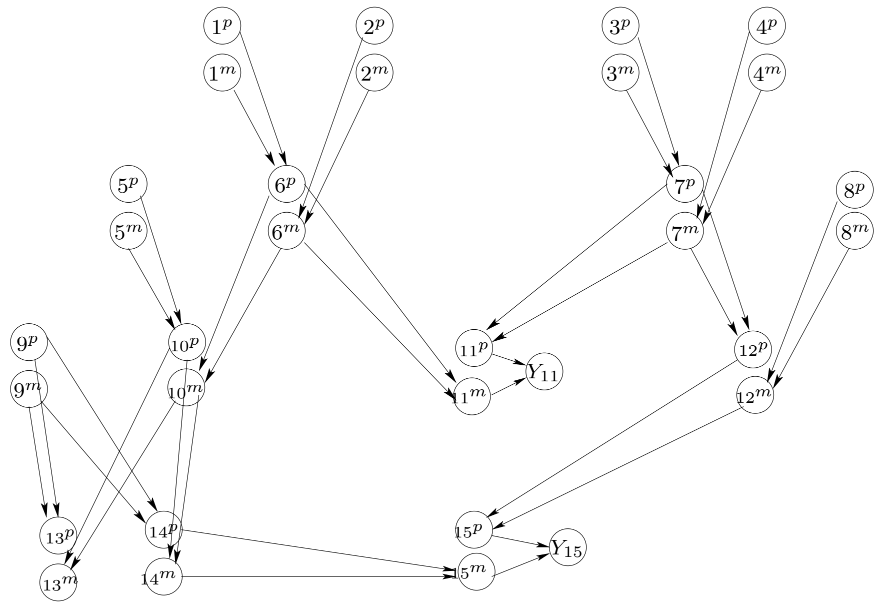

FIGURE 8: Bayesian allele network with phenotypic information on two individuals.

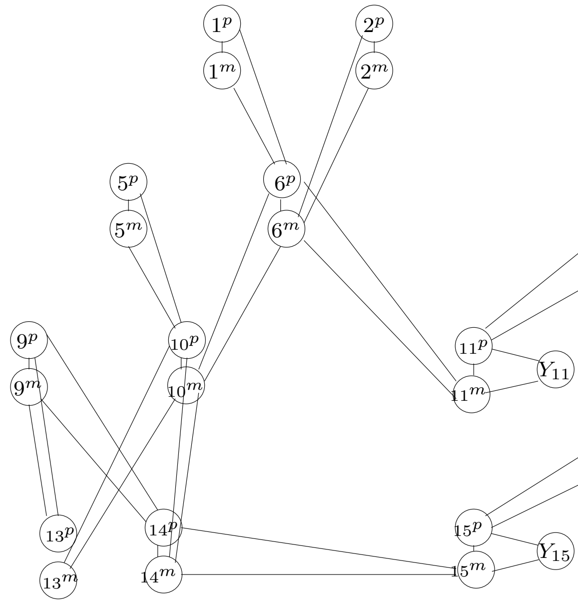

FIGURE 9: Moral graph for the Bayesian allele network of Figure 8 with phenotypic information on two individuals. The graph parents (allele pairs) have been married and directions dropped.

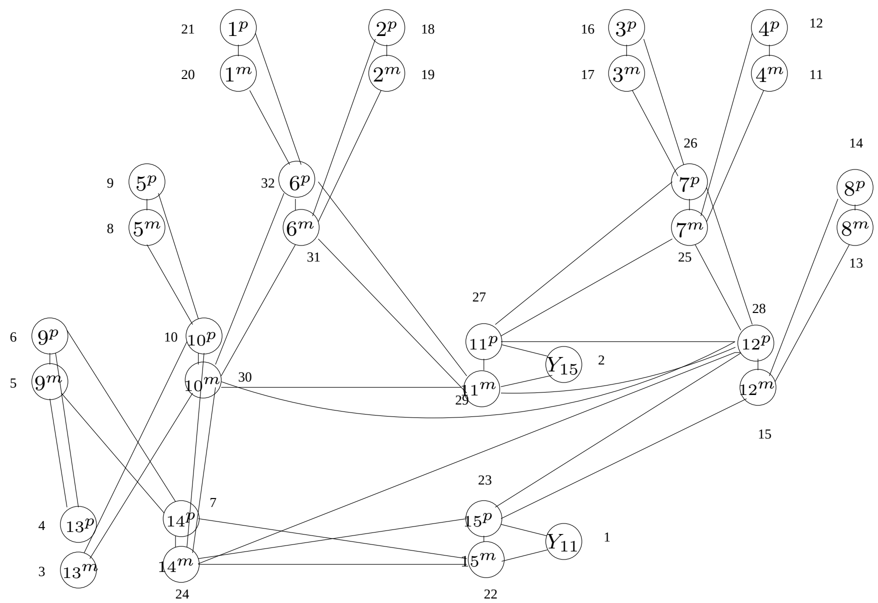

FIGURE 10: A triangulated graph for the Bayesian allele network of Figure 8 with phenotypic information represented for two individuals. The numbers $1, \ldots, 32$ indicate the corresponding node elimination ordering.

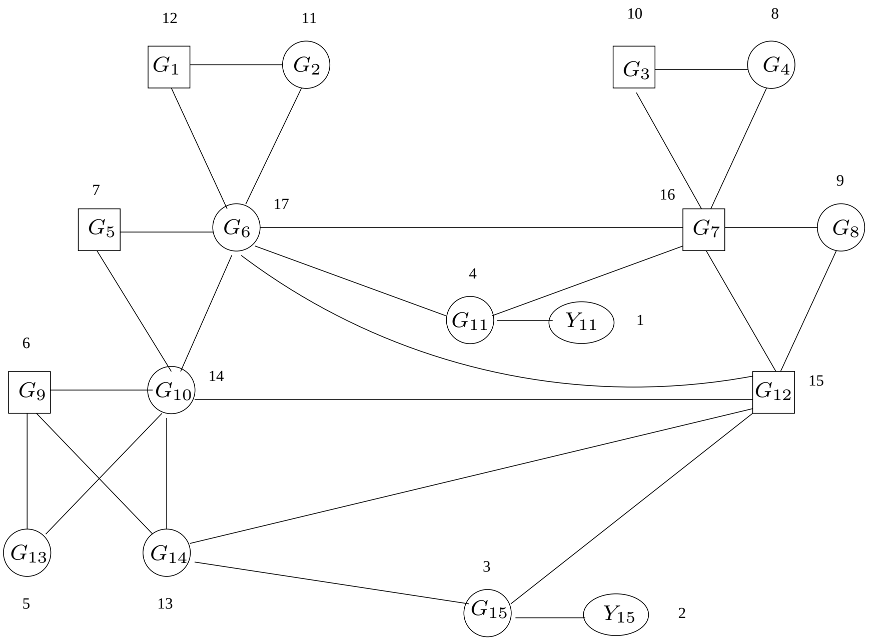

FIGURE 11: A triangulation and corresponding node elimination sequence for the genotype network associated with our example. Visually, the genotype network is the same as the relationship graph (i.e. Figure 2 with individuals 12 and 14 married with offspring 15) but the nodes represent the genotypes, $G_i, i = 1, \ldots 15$ of the individuals and two phenotype nodes, Y_{11} and Y_{15} have been added for the observations on 11 and 15.

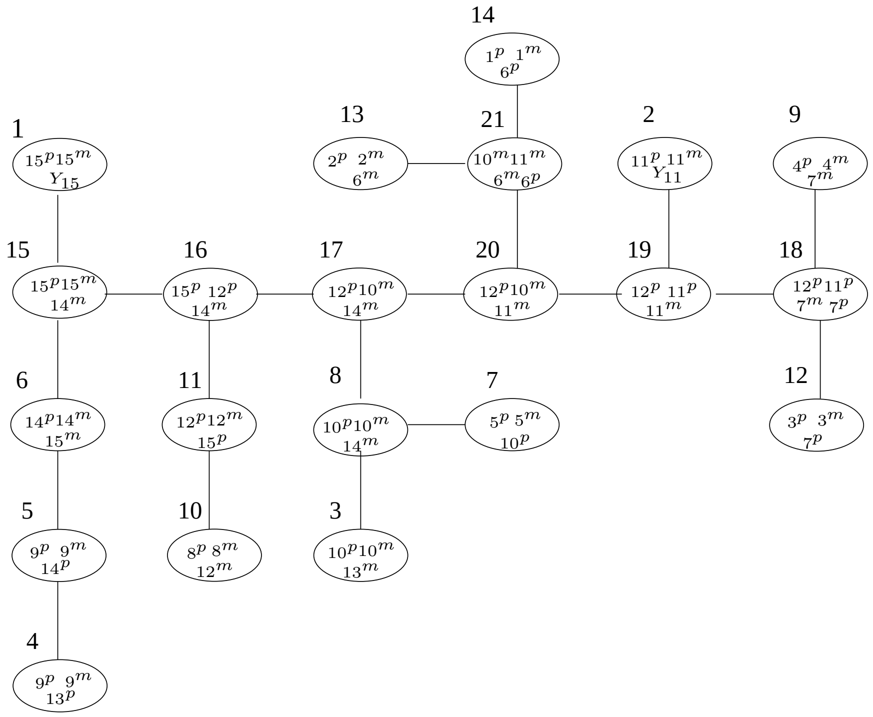

FIGURE 12: Junction tree for the Bayesian allele network showing the cliques as given in Table 2.

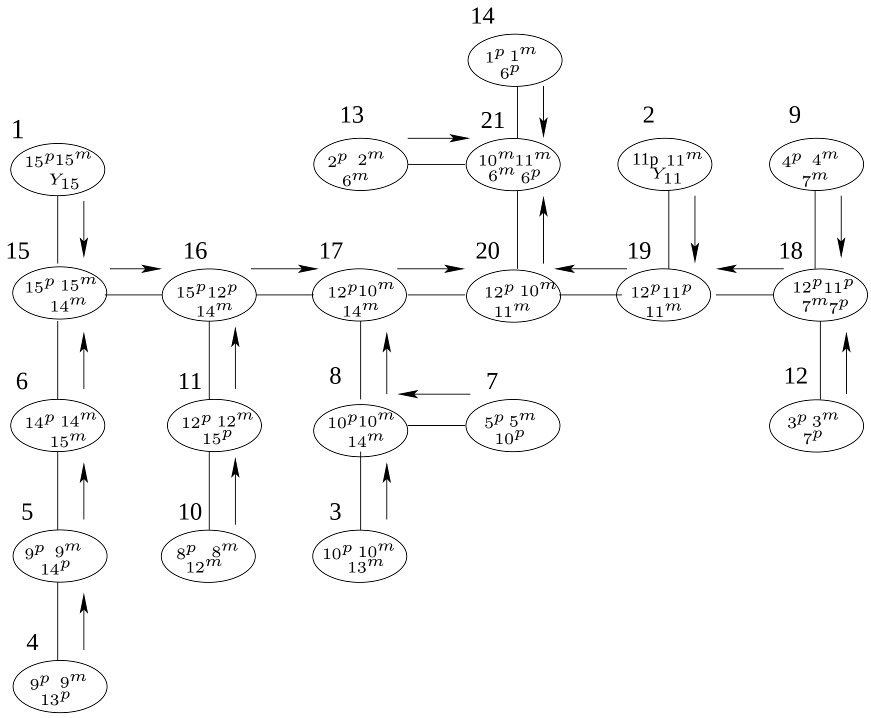

FIGURE 13: Peeling the junction tree towards a root. Messages are initially sent from leaves and propagate towards the root C_{21}. When all messages have been sent, the potential at C_{21} is proportional to the probability distribution of the variables in C_{21} and the normalisation constant is the joint likelihood of the observations. Messages are sent according to the numbering of the cliques, so C_1 sends first.

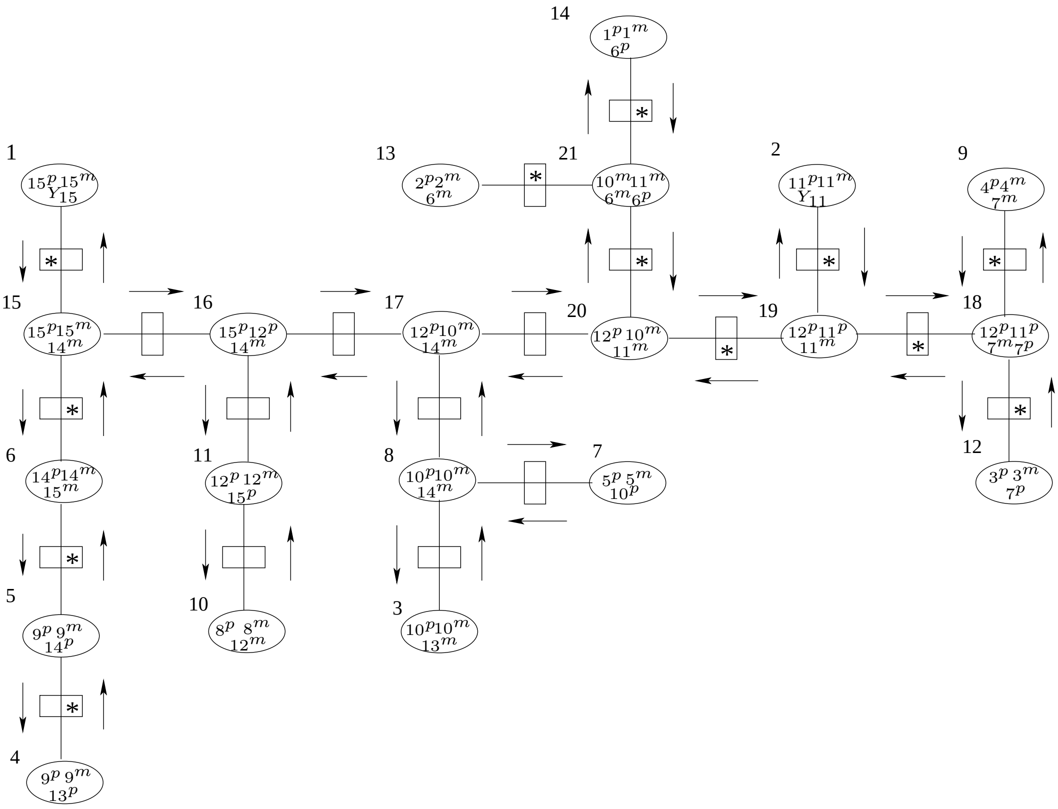

FIGURE 14: The Shafer-Shenoy procedure. Messages are stored in mailboxes along the branches of the junction tree. Every clique C can send a message to a neighbour D when all incoming mailboxes other than that from D are full. When all messages have been sent, a single local step at any C completes the procedure of peeling towards C as root. In the figure, mailboxes with $*$ are full, so at this point C_{20} is allowed to send to C_{17} but not to C_{19}.

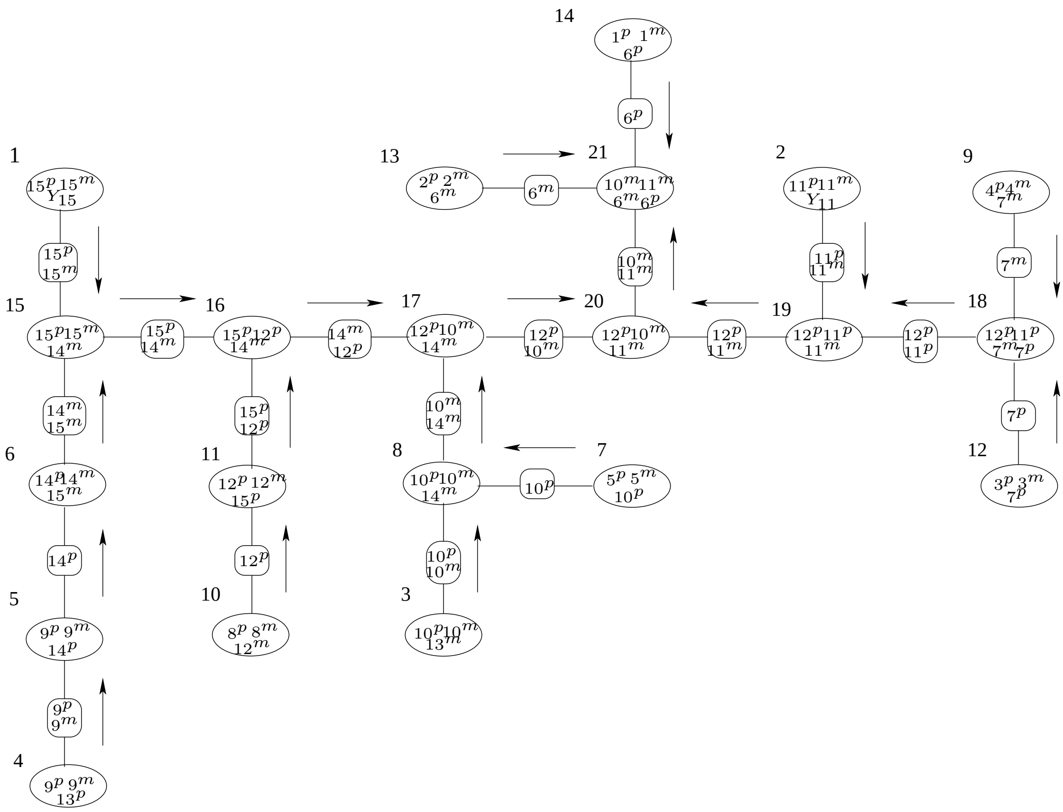

FIGURE 15: The first of the two computational phases in the HUGIN procedure. During COLLECTEVIDENCE messages are sent towards a root (C_{21}) as in the peeling procedure.

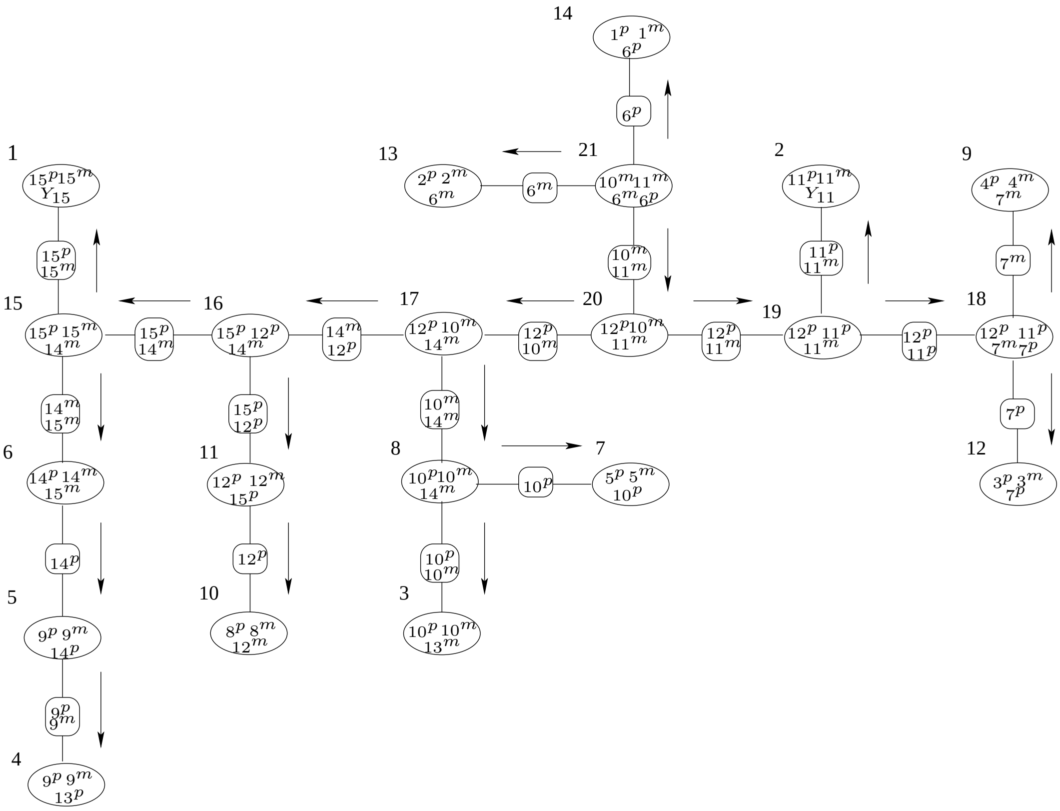

FIGURE 16: The second computational phase in the HUGIN procedure. After COLLECTEVIDENCE (Figure 15), DISTRIBUTEEVIDENCE sends messages from the root towards the leaves. After message passing, all cliques and separators are proportional to the relevant probability distribution, the normalising constant in all cases being equal to the likelihood, i.e. the probability of the observed data.

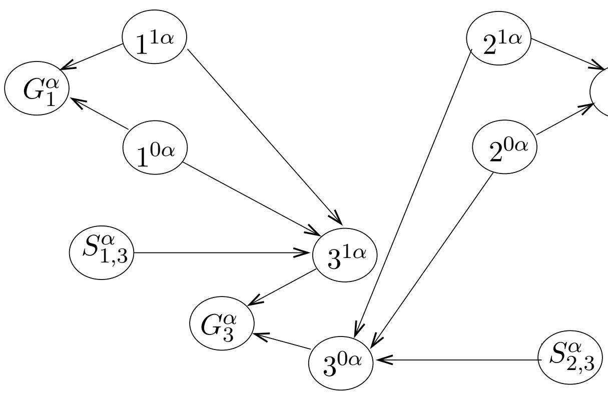

FIGURE 17: The segregation network for the $\alpha-$locus for a father, 1, mother 2 and child 3.

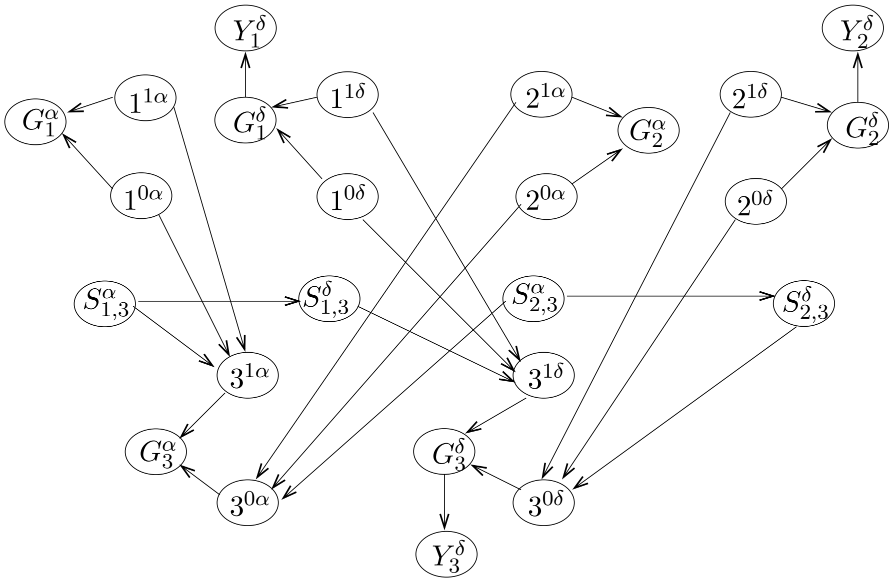

FIGURE 18: The segregation network for two linked loci on individuals 1 (father), 2 (mother) and 3 (offspring). Note that the information on linkage is contained in the directed edge between the segregation indicators $S_{1,3}^\alpha$ and $S_{1,3}^\delta$ and similarly between $S_{2,3}^\alpha$ and $S_{2,3}^\delta$.

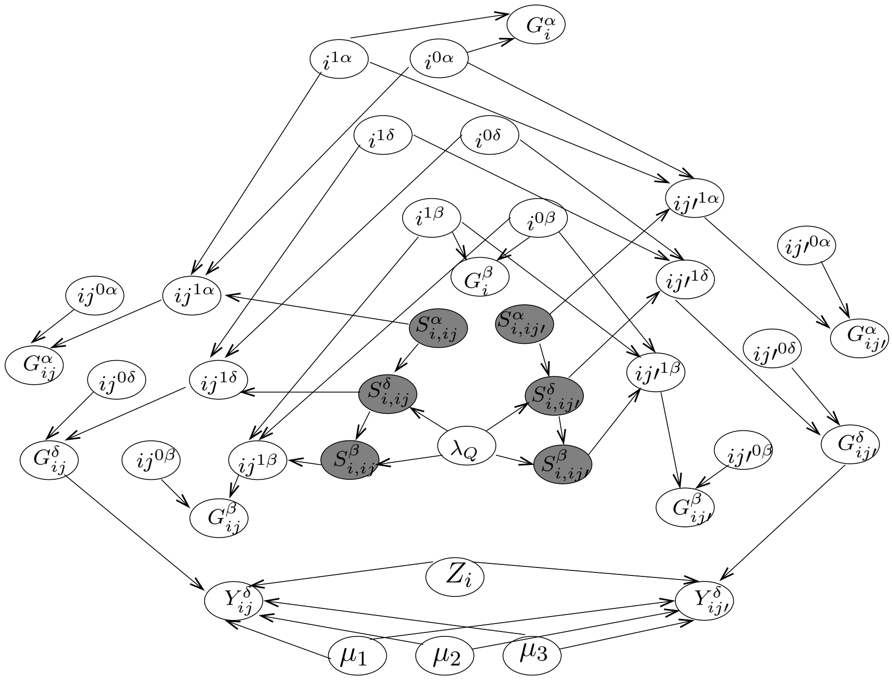

FIGURE 19: The graphical model for the QTL-mapping problem depicting a sire, i with two daughters, ij and $ij\prime$ adapted from Sheehan et al. (2002).

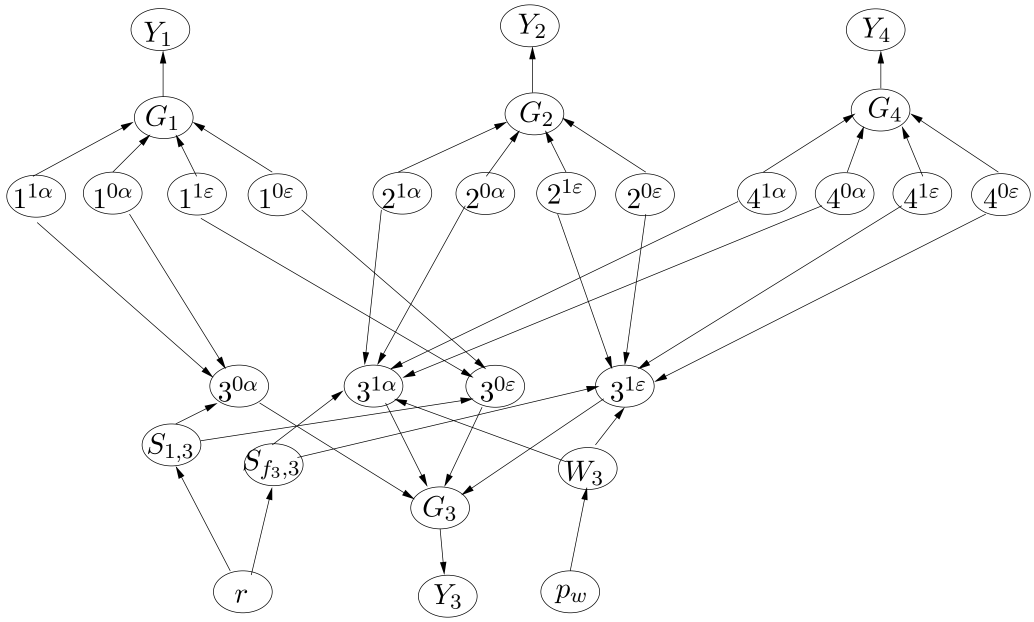

FIGURE 20: The graphical model for the fox data depicting a mother, 1, father 2, offspring 3 and alternative father 4 adapted from Hansen and Pedersen (1994).

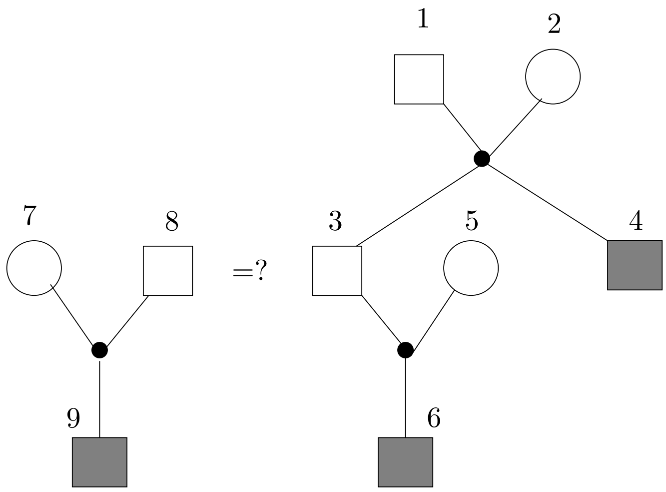

FIGURE 21: The simple paternity problem of Dawid et al. (2002) represented here by two marriage node graphs. . Individuals shaded in grey are those for whom DNA evidence is available. Note that individuals 1, 2, 5 and 7 are only drawn in to clarify relationships.

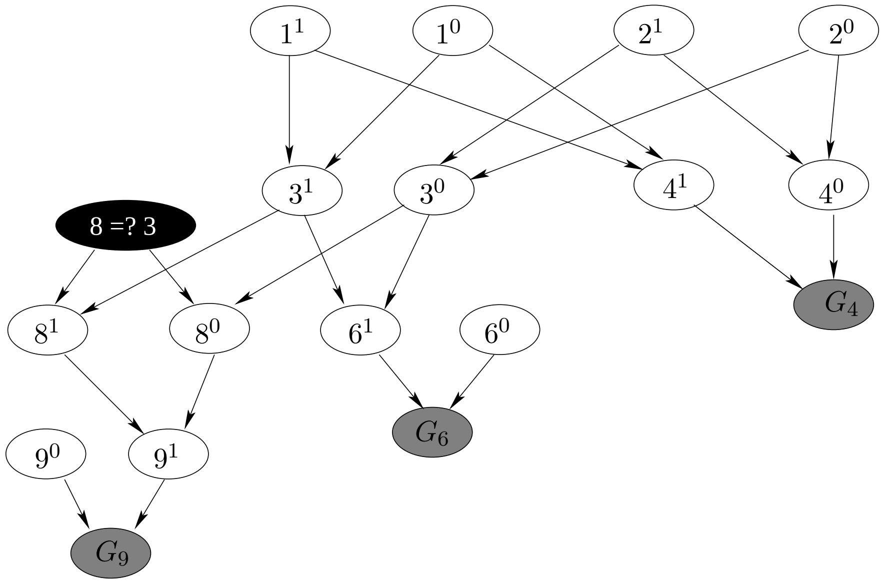

FIGURE 22: The graphical model representation used by Dawid et al. (2002) for the simple paternity problem of Figure 21. The three grey nodes represent the observed genotypes. The black node is the "query" node.

Table 1: The 6 genotypes for the human ABO system and the 4 corresponding observable characteristics or expressed phenotypes. The homozygous genotypes are AA, BB and OO, while the heterozygous types are AO, AB and BO. AA and AO have the same phenotype, so A dominates O or O is recessive to A. Similarly, B dominates O. The genotype AB has its own phenotype, however, and we conclude that A and B are *co-dominant*.

Genotype	**Phenotype**
AA	A
AO	
AB	AB
BB	B
BO	
OO	O

Table 2: Cliques, node assignments, and potentials for the junction tree of our allele network corresponding to the triangulation in Figure 10. We have identified each allele variable with its label.

Number	Elements	Assignments	Potential
1	$15^p, 15^m, Y_{15}$	Y_{15}	$f(Y_{15} \mid 15^p, 15^m)$
2	$11^p, 11^m, Y_{11}$	Y_{11}	$f(Y_{11} \mid 11^p, 11^m)$
3	$10^p, 10^m, 13^m$	13^m	$f(13^m \mid 15^p, 15^m)$
4	$9^p, 9^m, 13^p$	13^p	$f(13^p \mid 9^p, 9^m)$
5	$9^p, 9^m, 14^p$	$14^p, 9^m, 9^p$	$f(13^m \mid 9^p, 9^m)f(9^p)f(9^m)$
6	$14^p, 14^m, 15^m$	15^m	$f(15^m \mid 14^p, 14^m)$
7	$5^p, 5^m, 10^p$	$10^p, 5^m, 5^p$	$f(10^m \mid 5^p, 5^m)f(5^p)f(5^m)$
8	$10^p, 10^m, 14^m$	14^m	$f(14^m \mid 10^p, 10^m)$
9	$4^p, 4^m, 7^m$	$7^m, 4^m, 4^p$	$f(7^m \mid 4^p, 4^m)f(4^p)f(4^m)$
10	$8^p, 8^m, 12^m$	$12^m, 8^m, 8^p$	$f(12^m \mid 8^p, 8^m)f(8^p)f(8^m)$
11	$12^p, 12^m, 15^p$	15^p	$f(15^p \mid 12^p, 12^m)$
12	$3^p, 3^m, 7^p$	$7^p, 3^m, 3^p$	$f(7^p \mid 3^p, 3^m)f(3^p)f(3^m)$
13	$2^p, 2^m, 6^m$	$6^m, 2^m, 2^p$	$f(6^m \mid 2^p, 2^m)f(2^p)f(2^m)$
14	$1^p, 1^m, 6^p$	$6^p, 1^m, 1^p$	$f(6^p \mid 1^p, 1^m)f(1^p)f(1^m)$
15	$15^p, 15^m, 14^m$		1
16	$15^p, 12^p, 14^m$		1
17	$12^p, 10^m, 14^m$		1
18	$12^p, 11^p, 7^m, 7^p$	$11^p, 12^p$	$f(11^p \mid 7^p, 7^m)f(12^p \mid 7^p, 7^m)$
19	$12^p, 11^p, 11^m$		1
20	$12^p, 10^m, 11^m$		1
21	$10^m, 11^m, 6^m, 6^p$	$11^m, 10^m$	$f(11^m \mid 6^p, 6^m)f(10^m \mid 6^p, 6^m)$

Table 3: Comparison of total state space sizes of the allele and genotype network representations corresponding to the triangulations in Figures 10 and 11, respectively, for a single-locus trait with a alleles.

Number of alleles	Allele network size	Genotype network size
2	192	297
3	729	2376
4	1920	11000
5	4125	37125
10	48000	1830125

Table 4: Relationship between genotypes and fur colour phenotypes for the model proposed by Adalsteinsson et al. (1987).

Colour Phenotype	Two-locus Genotype	
Red Fox	AA	EE
Gold Fox	Aa	EE
Cross Foxes:		
Gold (Alaska) Cross Fox	AA	Ee
Silver (Blended) Cross Fox	Aa	Ee
Silver Foxes:		
Alaska Silver Fox	aa	EE
Canadian (Standard) Silver Fox	AA	ee
Sub-Canadian Silver Fox	Aa	ee
Sub-Alaska Silver Fox	aa	Ee
Double Black Fox	aa	ee

Table 5: Penetrance matrix adapted from Hansen and Pedersen (1994). Allowing for inconsistencies between the different classifications, entries marked with an asterisk have associated positive probability. All other entries have zero probability. The j^{th} element of the i^{th} row of this matrix is the conditional probability of observing phenotype j given genotype i.

Genotype		Phenotype							
AA	EE	*	*	*	-	-	-	-	-
Aa	EE	*	*	*	*	-	-	-	-
AA	Ee	-	-	-	*	*	*	-	-
Aa	Ee	-	-	-	-	*	*	*	-
AA	ee	-	-	-	-	-	-	*	*
aa	EE	-	-	-	-	-	-	*	*
Aa	ee	-	-	-	-	-	-	-	*
aa	Ee	-	-	-	-	-	-	-	*
aa	ee	-	-	-	-	-	-	-	*