BMC Bioinformatics

BioMed Central

Proceedings

Open Access

The Cluster Variation Method for Efficient Linkage Analysis on Extended Pedigrees

Cornelis A Albers, Martijn AR Leisink and Hilbert J Kappen*

Address: Department of Medical Physics and Biophysics, Radboud University, Nijmegen, The Netherlands

Email: Cornelis A Albers - k.albers@science.ru.nl; Martijn AR Leisink - m.leisink@science.ru.nl; Hilbert J Kappen* - b.kappen@science.ru.nl

* Corresponding author

from NIPS workshop on New Problems and Methods in Computational Biology
Whistler, Canada. 18 December 2004

Published: 20 March 2006

BMC Bioinformatics 2006, **7**(Suppl 1):S1 doi:10.1186/1471-2105-7-S1-S1

Abstract

Background: Computing exact multipoint LOD scores for extended pedigrees rapidly becomes infeasible as the number of markers and untyped individuals increase. When markers are excluded from the computation, significant power may be lost. Therefore accurate approximate methods which take into account all markers are desirable.

Methods: We present a novel method for efficient estimation of LOD scores on extended pedigrees. Our approach is based on the Cluster Variation Method, which deterministically estimates likelihoods by performing exact computations on tractable subsets of variables (clusters) of a Bayesian network. First a distribution over inheritances on the marker loci is approximated with the Cluster Variation Method. Then this distribution is used to estimate the LOD score for each location of the trait locus.

Results: First we demonstrate that significant power may be lost if markers are ignored in the multi-point analysis. On a set of pedigrees where exact computation is possible we compare the estimates of the LOD scores obtained with our method to the exact LOD scores. Secondly, we compare our method to a state of the art MCMC sampler. When both methods are given equal computation time, our method is more efficient. Finally, we show that CVM scales to large problem instances.

Conclusion: We conclude that the Cluster Variation Method is as accurate as MCMC and generally is more efficient. Our method is a promising alternative to approaches based on MCMC sampling.

Background

The goal of genetic linkage analysis is to link phenotype to genotype. Pedigrees are collected where a trait or disease is believed to have a genetic component. The individuals in the pedigree are genotyped for a number of markers on the chromosome. The markers are at known relative recombination frequencies, so that from the genotypes a distribution over inheritances can be inferred. Linkage of the trait to a specific location in the marker map then is quantified by the extent to which the distribution over inheritances as inferred from the markers can explain the observed phenotypes in the pedigree.

Parametric linkage analysis

In this article we compute linkage likelihoods with the parametric LOD score (log odds ratio) proposed by Morton [1]. The LOD score is the log ratio of the likelihoods of the hypothesis that the disease locus is linked to the marker loci at a specific location and the hypothesis that it is unlinked to the marker loci. The LOD score requires

specification of the disease frequency and penetrance values and therefore falls into the category of parametric scoring functions.

Exact computations

Several methods for exact computations are in use.

Lander et al. [2] introduced a Hidden Markov Model (HMM) where the meiosis indicators are the unobserved variables. This method is linear in the number of loci, but exponential in $2n - f$, where n is the number of non-founders and f the number of founders. Kruglyak et al. [3] optimized the method in the program Genehunter.

Elston et al. [4] developed an algorithm that is efficient on pedigrees that have little inbreeding. This method is linear in the number of individuals (in case there is no inbreeding) but scales with the number of possible multi-locus genotypes. The method was made computationally efficient in the package Vitesse [5].

Both of these methods exploit particular independence properties of the statistical model. Within the framework of Bayesian networks, this approach has been generalized in the junction tree algorithm [6,7]. In the computer program Superlink [8] this approach is implemented for the application of linkage analysis and is the first program to make use of Bayesian networks for computing exact linkage likelihoods.

Although exact algorithms have been substantially improved over the years, the fact remains that they require an exponential number of operations and have limited applicability.

The Cluster Variation Method

The Cluster Variation Method originated with the work of Bethe [9] and was extended to non-pair wise marginals by Kikuchi [10] to compute properties of magnetic materials, such as Ising models. In later years, the method has been extended and reformulated [11,12]. Recently the method has been introduced into the machine learning community [13,14]as a method for approximate inference in Bayesian networks and undirected graphical models.

The Cluster Variation Method approximates an intractable probability distribution in terms of marginal probability distributions on clusters of variables. These clusters of variables are chosen such that exact computations are feasible on each cluster. We make explicit use of the formulation of linkage analysis in terms of a Bayesian network to choose which variables will be contained in the clusters. In contrast with MCMC the approximation is deterministic and yields estimates of the pedigree likelihood.

CVM and linkage analysis

As large complex pedigrees with individuals genotyped at a large number of locations become increasingly available, along comes the need for methods of estimating likelihoods on pedigrees where exact computations are not possible.

In this article we describe in detail how the Cluster Variation Method can be applied to the problem of genetic linkage analysis on pedigrees without inbreeding. We discuss extension of our approach to inbred pedigrees.

Results

We compare the estimates of the LOD score obtained with our method to exact scores as computed with Vitesse [5]. We also compare our method to Markov Chain Monte Carlo (MCMC) simulations. For this we have used version 2.5 of the Morgan sampler [15]. This MCMC sampler is optimized for pedigrees for which exact single locus computations are possible. To our knowledge this is the most advanced sampler for the pedigrees we consider.

We consider CVM converged if the marker marginals change by no more than 10^{-3}. We use the following settings for the Morgan sampler: the number of prior samples and burn-in samples are set to respectively 50 % and 10 % of the number of samples used for the actual estimates.

We performed all experiments on a Pentium-IV 2.8 GHz with 1 GB of physical memory running Linux.

Simulations

We start by motivating the use of approximate methods with an example. On the pedigree shown in figure 1, we have simulated a dominant disease with penetrance values $\mathbf{f}$ = (0.02, 0.98, 0.98) and trait allele frequencies $\mathbf{t}$ = (0.98, 0.02), at 0 cM. This pedigree can be handled by Genehunter [3]. We have simulated 25 pedigrees, where half of the individuals has genotypic and phenotypic data. 14 bi-allelic markers were simulated with marker allele frequencies $\mathbf{m}$ = (0.4, 0.6) for all 14 markers. The marker spacings are 1 cM.

In figure 2 we now compare exact LOD scores computed with all of the 14 available markers to exact LOD scores computed with only a subset of the markers. The figure shows that significant power is lost when markers are excluded from the multi-point analysis. The solid line represents exact LOD scores computed with all available markers. The LOD score peaks at 0 cM, where the disease was indeed simulated. The dotted line represents the LOD score based on 5 markers: for each location of the trait locus, a LOD score is computed by doing a multi-point calculation with the 2 markers to the left of the trait locus,

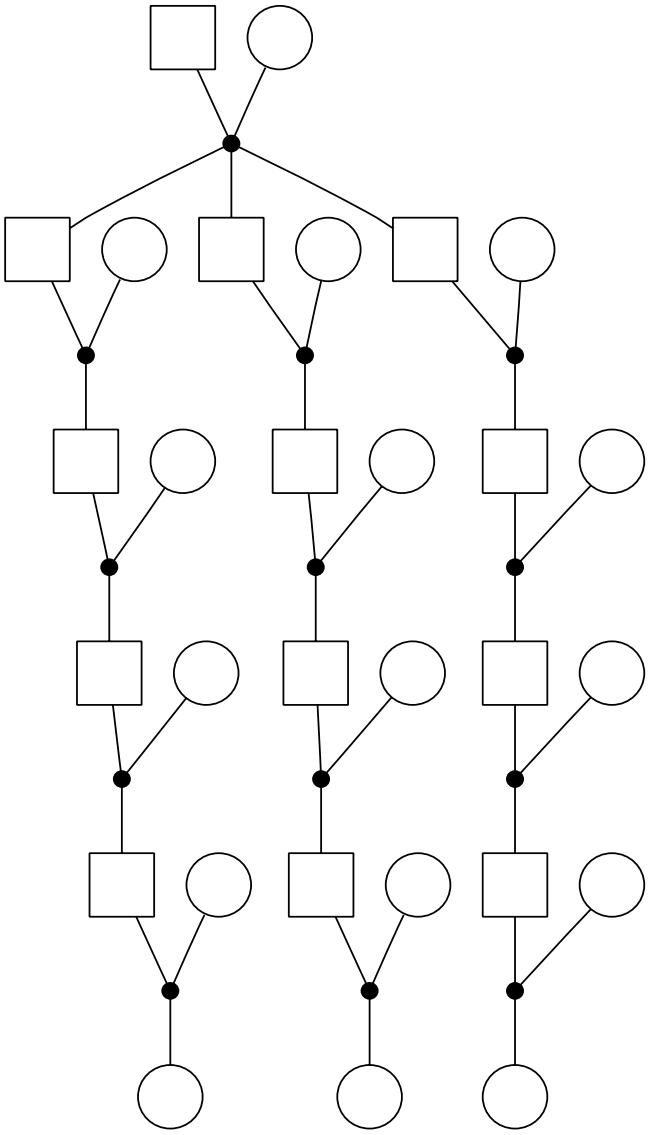

Figure 1
Pedigree 1. Pedigree used for the results of figure 2.

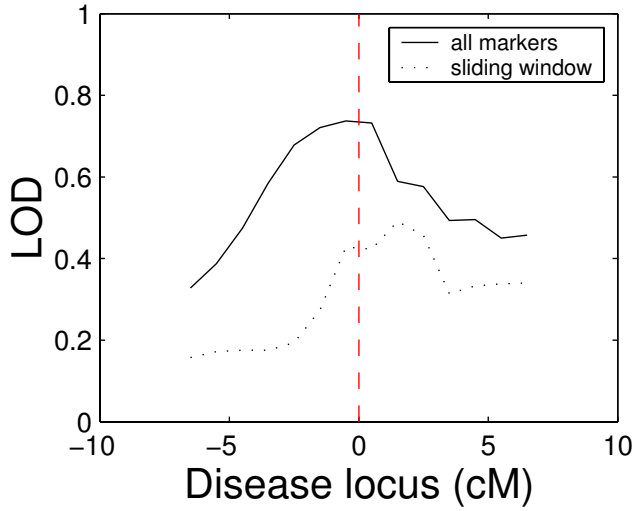

Figure 2
Power of analysis. Power decreases when markers are excluded from the multi-point analysis. On the pedigree of figure 1 a dominant disease is simulated at 0 cM. Solid line represents exact LOD scores based on all 14 markers; dotted line represents exact LOD scores based on 4 markers surrounding the trait locus.

2 markers to the right of the trait locus, and the marker at which the trait locus is located. If the trait locus is located on the first marker in the marker map, no markers to the left of this marker are available so that the first 5 markers are used to calculate the LOD score for this location of the trait locus. This approach can be characterized as a sliding window approach.

In this example the pedigree was small so that exact scores can be computed with Genehunter for a virtually unlimited number of markers. However, as the size of the pedigree increases, the number of markers that can be analyzed simultaneaously drops rapidly. In that case significant power may be lost. Thus, an accurate approximate method that can take into account all markers is desirable.

We now compare the estimates of the CVM and MCMC to the exact scores. The results are obtained on the pedigree shown in figure 3. There are 48 individuals of which 10 are founders. The number of children per nuclear family increases from two in the second generation to five in the third generation. We simulate phenotypes and genotypes according to this pedigree. We consider a dominant disease with penetrance values $\mathbf{f} = (0.02, 0.90, 0.90)$. The disease allele frequency has been set to 2 %, so that $\mathbf{t} = (0.98, 0.02)$. We assume that for each individual in the pedigree the affection status is known. For a marker spacing of 5 cM, we simulated 25 pedigrees with 3 markers and at least 15 affected individuals per pedigree. The number of alleles is 5 per marker with equal frequencies. 70 % of the individuals in the last two generations is genotyped for all markers. The individuals in the first two generations are not genotyped.

In figure 4 we compare the quality of the approximation resulting from two cluster choices C_1 and C_2. These cluster choices are specified in figure 5. The error is defined as the absolute difference between the exact LOD score and the CVM estimate of the LOD score, averaged over all positions of the trait locus. We see that the error of the larger

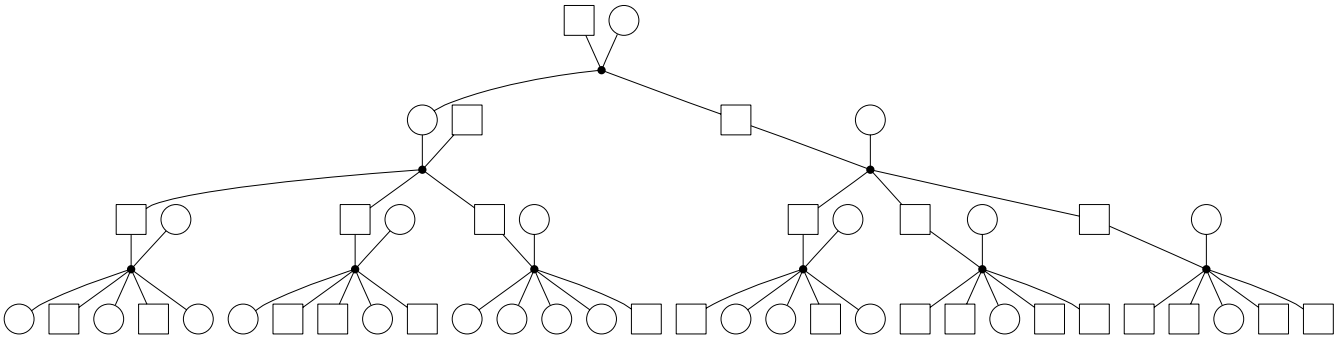

Figure 3
Pedigree 2. Pedigree used for the results of figures 5-8. This pedigree consists of 48 individuals, of which 10 are founders.

clusters of C_1 is small and an order of magnitude smaller than the error of the smaller clusters of C_2.

This result demonstrates two points. First, it is important to include the interactions between meiosis variables on adjacent loci into at least one cluster. Second, the accuracy of the approximation can be adapted by increasing the number of variables per cluster. Although it is not the case that an increased number of variables per cluster guarantees a higher accuracy of the approximation, in our experience it is generally possible to obtain more accurate estimates by increasing the number of loci and/or the number of generations covered by a cluster.

For a difficult problem in the dataset, i.e. one where MCMC and CVM error are relatively large, we compare our result to MCMC estimates obtained with Morgan. In figure 6 the decrease of the CVM and MCMC error as a function of computation time (i.e. number of samples) is shown. We see that a significant increase of the computation time does not significantly decrease the error and variance of the MCMC estimate. The error of the CVM estimates obtained with cluster choice C_1 is indicated by the dashed line. The CVM computation time is varied by adjusting the value of the convergence criterion. We conclude that our method achieves higher accuracy for a given amount of computation time.

For the other pedigrees in the data set we compare CVM to MCMC, where for each problem MCMC is alotted the computation time required by CVM with cluster choice C_1. The results are shown in figure 7. We see that the MCMC estimates are less accurate. The average CVM computation time is 700 seconds, although there is a considerable degree of variance in the order of 100 seconds. Memory requirements vary between 100 and 250 MB, depending on the informativeness of the markers. We did not find a correlation between CVM computation time and the absolute error with respect to the exact distribution. Also the outliers in the figure are not explained by large CVM computation times that consequently lead to an improvement of the MCMC estimate, as the MCMC computation time is fixed to the CVM computation time. Since both methods will theoretically converge to the exact solution in the limit of infinite time resources, this is the only fair comparison. Additional simulations (not shown) indicate, in agreement with the results reported in figure 6, that the MCMC estimates are sufficiently converged. The CVM estimates are reproducible; variance is in the order of the convergence criterion.

We now demonstrate that the method scales to larger problem instances. We therefore vary the number of

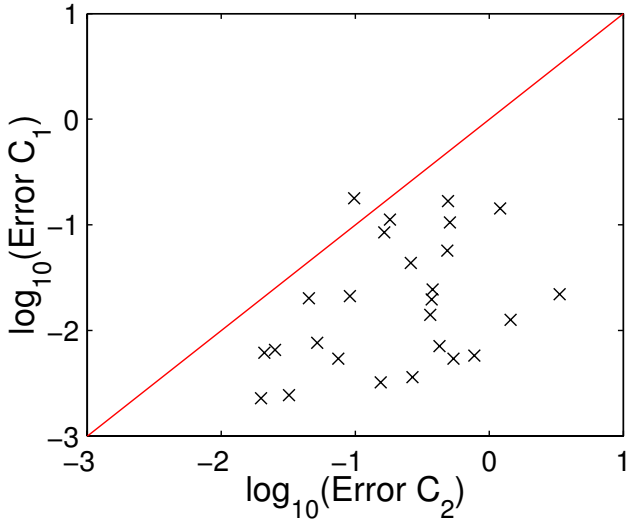

Figure 4
Comparison of cluster choices. Error of cluster choice C_1 versus error of cluster choice C_2 for a marker spacing of 5 cM. Error of cluster choice C_1 is an order of magnitude smaller than error of cluster choice C_2.

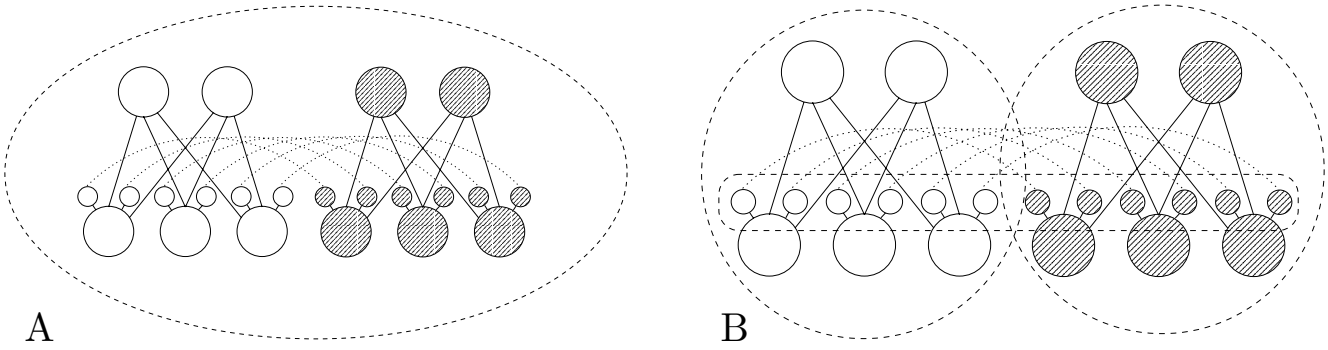

Figure 5
Cluster choices. Nodes on neighboring markers l and $l + 1$ (shaded) that form a cluster. Large node represents the genotype of an individual, a small node the meiosis indicator an individual. In the CVM approximation, only marginals of clusters on the marker loci are computed. A shows cluster choice C_1. Genotype nodes of parents and children and meiosis nodes of children in the nuclear family of adjacent marker loci form a single cluster. B shows cluster choice C_2. Genotype nodes of parents and children and meiosis nodes of children in a nuclear family on one marker locus form a cluster. The meiosis nodes of the children form a separate cluster.

markers, since Vitesse can handle very large pedigrees with no loops, but only a small number of markers. We have simulated a dominant disease on the pedigree of figure 3 and 32 bi-allelic markers with equal allele frequencies. We have simulated one pedigree where all individuals are genotyped for all 32 markers. From this instance we create 16 problems by selecting a subset of the markers of the original problem.

In figure 8 we show that CVM computation time scales approximately linearly with the number of markers, as do memory requirements (not shown). The varying informativeness of the markers explains the fluctuations. Vitesse cannot handle more than 10 markers, because memory requirements exceed the available 1 GB. Memory requirement of CVM for 32 markers is 150 MB. In this case, MCMC estimates take several hours to reach convergence. We conclude that our method scales to large problem instances.

Discussion

We compared our method to the MCMC implementation of the Morgan sampler, which is to our knowledge the most advanced program for this problem. There are packages that can handle more general pedigrees than Morgan, such as SIMWALK2 [16,17], but here we have investigated only pedigrees without inbreeding. Preliminary results indicate the CVM approximations based on the cluster choices presented in this article can give good results on inbred pedigrees. Extension to inbred pedigrees is possible and a direction for further research.

Theoretically, if the sampler is irreducible, MCMC estimates should ultimately converge to the exact score. However, in practice this may require extremely long computation times. The Cluster Variation Method does not guarantee that for a fixed choice of clusters the approximate marginals will converge to marginals of the exact distribution, but the same argument that holds for MCMC also holds for CVM: we can increase the cluster size and consequently computation time to improve the quality of the approximation.

In this article we have proposed cluster choices that generally give good results. Sometimes when many individuals are untyped the estimates can be inaccurate. Interestingly, on these problems the Morgan sampler also experienced severe difficulties.

In the approach we have taken, we can define a heuristic to detect errors in the approximation. Suppose we have three markers; then the LOD score for marker 2 can be computed either from the marginals defined on the nuclear families on the first and second marker, or from the marginals defined on the nuclear families on the second and third marker. If these LOD scores differ significantly, one should be very careful in interpreting the estimate and the number of variables per cluster must be increased. We cannot guarantee that if the LOD scores *are* consistent, the approximation is accurate. An obvious and useful extension would be an automatic procedure that gives the optimal set of clusters. However such a procedure is far from trivial, and the guideline to choose the clusters as large as available memory permits seems to work well in practice.

In the current implementation we have applied a number of preprocessing techniques to improve the efficiency. We expect that even better efficiency can be obtained by applying more preprocessing techniques such as genotype

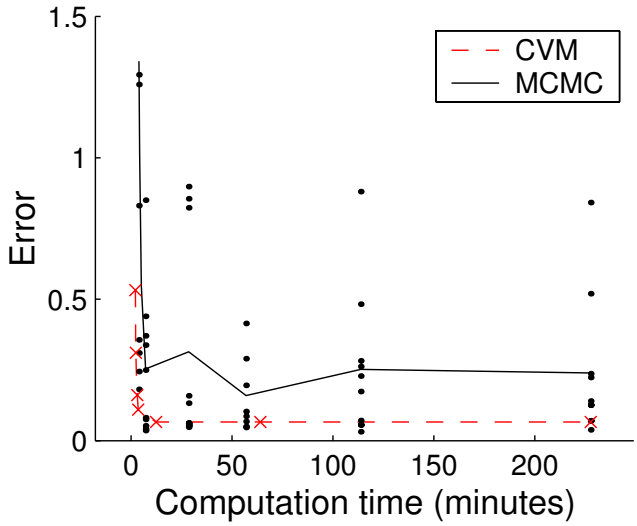

Figure 6
Error as function of computation time. MCMC and CVM error as a function of computation time. Error is defined as the absolute difference between the exact LOD score and the estimated LOD score, averaged over all positions of the trait locus. MCMC error of 10 independent runs do not converge to a better estimate than the CVM estimate with C_1. CVM computation time is varied by adjusting the convergence criterion. Thus, CVM achieves higher accuracy for a given amount of computation time. CVM estimates are reproducible.

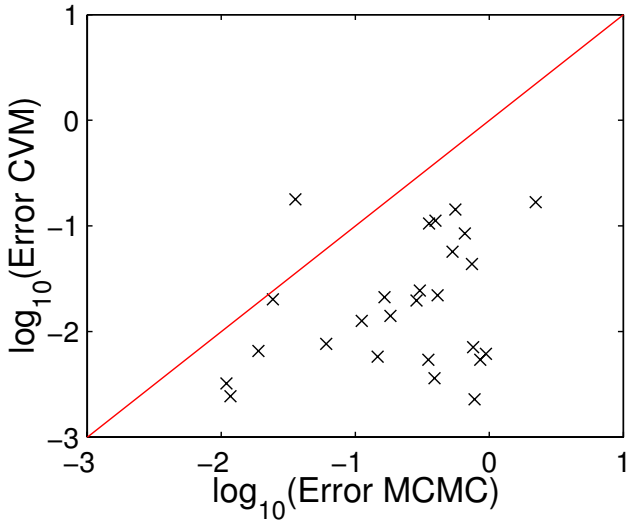

Figure 7
Comparison of CVM and MCMC error. Error of CVM estimate obtained with cluster choice C_1 versus error of MCMC for a marker spacing of 5 cM compared to exact results obtained with Vitesse. MCMC is alotted the same computation time as CVM. CVM yields more accurate estimates than MCMC.

elimination [18,19], and techniques specific to Bayesian networks such as value abstraction [20] and evidence based compiling [21]. Also, preliminary simulations indicate that smaller clusters can give equally accurate estimates with reduced memory requirements.

Other applications fit naturally into the framework presented here. Since the Cluster Variation Method is able to estimate pedigree likelihoods directly, the method presented here can be used directly to estimate recombination frequencies and marker ordering errors with a maximum likelihood approach. Maximum likelihood haplotyping on general pedigrees also is very promising in this framework.

Conclusion

In this article we have demonstrated the feasibility of a new approach to compute linkage likelihoods for linkage problems that are beyond the reach of exact computations. Previous methods that are suited to deal with these intractable problems relied on sampled estimates. We have shown that a deterministic approach based on the Cluster Variation Method is able to obtain accurate estimates of LOD scores and generally is more efficient than MCMC methods.

Methods

A Bayesian network formulation

We briefly describe the Bayesian network that will enable us to compute likelihoods. Any probability distribution can be represented with a Bayesian network [22]. Therefore the use of Bayesian networks is merely a matter of convenient representation of a probability distribution, and is irrelevant to the issue of Bayesian versus frequentist statistics. Bayesian networks in the context of genetics have first been applied by Jensen et al. [23]. Their approach was extended by Thomas et al. [24]. The use of Bayesian networks for exact computations has been proposed by Fishelson et al. [8]. An extensive discussion of Bayesian networks in the context of genetics is given by Sheehan et al. [25]. These articles have demonstrated the power of Bayesian networks for linkage analysis.

The transmission of alleles from parents to children is clearly a directed process. A Bayesian network represents a probability distribution in terms of a directed graph, i.e. a graph where the links between the variables are directed. A Bayesian network is therefore particularly suited to model the probability distribution associated with the problem of multi-point linkage analysis. By specifying conditional probability tables for each variable, the formalism of Bayesian networks guarantees that the corresponding probability distribution is consistent and normalized [22].

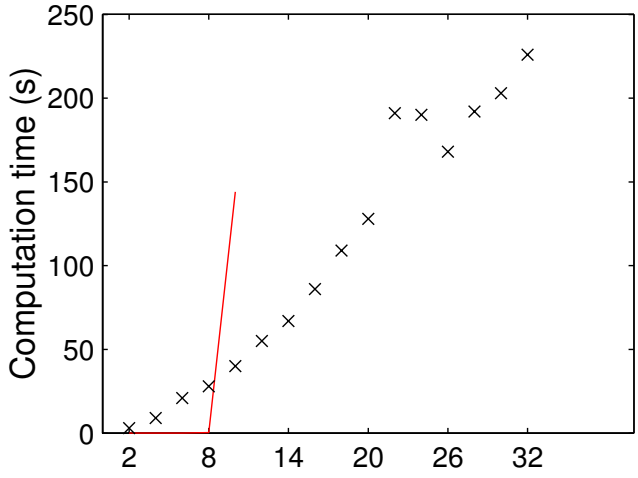

Figure 8
Scaling of CVM computation time with number of markers. Scaling of CVM computation time with the number of marker loci is indicated by the crosses. Scaling of computation time is approximately linear. All individuals have marker information, the number of alleles is 2. The red curve represents computation time of Vitesse. Vitesse cannot handle more than 10 markers as memory requirements exceed system memory of 1 GB. For the problem of 32 markers, MCMC estimates take several hours to reach convergence.

While the Bayesian network is the most convenient for model specification, it is not possible to apply the Cluster Variation Method directly to a Bayesian network. In order to perform inference, the Bayesian network is converted to an undirected graphical model by the procedure of moralization [7]. Moralization removes the directions of the links and adds links between the parent variables of a variable, i.e. all variables with a link directed towards a given variable. The undirected graphical model represents exactly the same probability distribution as the directed graphical model; the addition of extra links ensures that the correlations encoded in the conditional probability tables will be correctly taken into account by the inference method. The formal procedure of converting the Bayesian network is standard practice and used by all of the above mentioned methods that make use of Bayesian networks.

The Cluster Variation Method requires specification of which variables are contained in each cluster, and this specification becomes very transparent when the dependencies between these variables are modeled with a Bayesian network. The Bayesian network consists of a number of marker loci, with known relative recombination frequencies θ, and a single trait locus linked to the markers at a given position λ_T. The purpose of linkage analysis is to determine the most likely position of the trait locus relative to the markers. To that end a Bayesian network is constructed for each possible location λ_T of the trait locus, so that the likelihood of the trait phenotypes and marker genotypes can be computed for that location of the trait locus. The ratio of this likelihood and the likelihood of the trait locus unlinked to the markers then gives the LOD score for location λ_T.

Single locus model

First we define the Bayesian network for a single locus. The inheritance model is shown in figure 9A. Each variable is represented graphically by a node. A conditional probability table for a variable is defined by the variable itself and all variables (or, equivalently, nodes) that have a link which points to that variable. In the figure, the variables are the genotypes and the meiosis indicators. Each individual, denoted by the subscript i, possesses two genes $G_i^{l,p}$ and $G_i^{l,m}$ that correspond to the paternally and maternally inherited allele, indicated by the superscript p and m respectively. The meiosis indicators $v_i^{l,p}$ and $v_i^{l,m}$ indicate whether the paternal or the maternal allele of respectively the father and the mother is inherited. The nodes $G_i^{l,p}$ and $G_i^{l,m}$ take the values 1,..., $|m_l|$, with $|m_l|$ the number of marker alleles for marker locus l. We will use the shorthand notation $\mathbf{G}_i^l = (G_i^{l,p}, G_i^{l,m})$. The father and mother of individual i are denoted by $f(i)$ and $m(i)$ respectively, and in the following we will also use $\pi(i) = (f(i), m(i))$ to denote both parents.

Figure 9A is a graphical representation of the following conditional probability tables in the Bayesian network:

$$P\left(\mathbf{G}_i^l \mid \mathbf{v}_i^l, \mathbf{G}_{\pi(i)}^l\right) = P\left(G_i^{l,p} \mid v_i^{l,p}, G_{f(i)}^{l,p}, G_{f(i)}^{l,m}\right) \times P\left(G_i^{l,m} \mid v_i^{l,m}, G_{m(i)}^{l,p}, G_{m(i)}^{l,m}\right). \quad (1)$$

We use boldface to indicate vectors over the missing subscripts and superscripts. For each individual i that is not a founder we have two conditional probability tables as in equation 1. If individual i is a founder, we have a prior distribution on the genotypes instead: $P(\mathbf{G}_i^l \mid \mathbf{m}_l)$, where $\mathbf{m}_l$ represents the marker allele frequencies for marker l; on the trait locus, we have $P(\mathbf{G}_i^{l_T} \mid \mathbf{t})$, where $\mathbf{t}$ represents the trait allele frequencies.

We note that the genotypes of all non-founders are completely determined by the genotypes of the founders and the meiosis indicators $\mathbf{v}$. The meiosis indicators com-

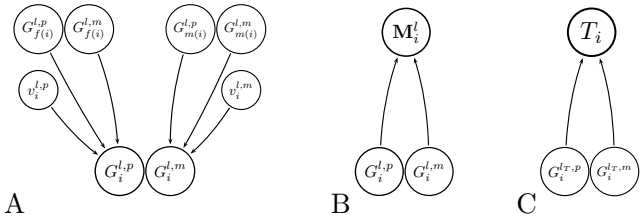

Figure 9

Single locus Bayesian network. A shows transmission model. The paternal allele $G_i^{l,p}$ of individual *i* on locus *l* is determined by the state of its paternal meiosis indicator $v_i^{l,p}$ and the genotype of its father denoted by $\mathbf{G}_{f(i)}^{l}$. The maternal allele is analogous. B shows marker observation model and C shows the trait observation model.

pletely specify the flow of the alleles. Thus, the genotypes of non-founders are not strictly necessary for computing the appropriate likelihood; however, they do simplify the structure of the Bayesian network such that the Cluster Variation Method can be applied.

The graphical representation for the marker and trait observations is shown in figure 9B and 9C respectively. If an individual *i* has marker data for marker locus *l*, then we have two marker alleles ($m_i^{l,1}$, $m_i^{l,2}$). However, it is not known which allele corresponds to the paternal allele and which one corresponds to the maternal allele. The phase ambiguity is reflected in the marker observation model. For example, consider $\mathbf{G}_i^l = (1, 2)$, then the only non-zero probabilities are

$$P\left(\mathbf{M}_i^l = (1,2) \middle| \mathbf{G}_i^l = (1,2)\right) = \frac{1}{2} \text{ and}$$

$$P\left(\mathbf{M}_i^l = (2,1) \middle| \mathbf{G}_i^l = (1,2)\right) = \frac{1}{2}.$$

Here we will only consider binary traits, although multi-valued or real-valued traits in principle are possible. Therefore, we need only two possible alleles on the trait locus, one which is assumed to cause the phenotype and one which is unrelated to the phenotype. The dependence of the trait T_i on the genotype $\mathbf{G}_i^{l_T}$ is specified with the penetrance values $\mathbf{f} = (f_0, f_1, f_2)$. The probabilities f_0, f_1 and f_2 are the probabilities $f_n = P(T_i = \text{affected} | \#g = n)$, where g is the number of trait alleles and n = 0, 1, 2. The trait model introduces the conditional probability table

$$P\left(T_i \mid \mathbf{G}_i^{l=l_T}, \mathbf{f}\right). \qquad (2)$$

Multi-locus model

The full Bayesian network of the multi-locus model consists of the single locus models for all markers and the trait locus. The recombinations between loci are modelled by adding links between the meiosis indicators of adjacent loci of the same individual, as illustrated in figure 10. In the absence of data, the meioses of any two individuals are independent. However, the meioses of a single individual are not independent. They depend on each other through the relation

$$\begin{aligned} P\left(\mathbf{v}_i^{l+1} \mid \mathbf{v}_i^l, \theta_{l+1,l}\right) &= P\left(v_i^{l+1,p} \mid v_i^{l,p}, \theta_{l+1,l}\right) \\ &\times P\left(v_i^{l+1,m} \mid v_i^{l,m}, \theta_{l+1,l}\right). \end{aligned}$$

These conditional probability tables are parameterized by the recombination frequency $\theta_{l+1,l}$ between the adjacent loci[1]. The first locus does not have a left neighbor, so we use a flat prior on its meiosis indicators. In the genetic linkage analysis of a pedigree, it is assumed that the recombination ratios between the markers are known. That is, for any two adjacent markers *l*, *l'*, the recombination frequency $\theta_{l,l'}$ is specified. The recombination frequency between the markers and the trait locus is fixed for a given position λ_T of the trait locus and can be determined from the marker map.

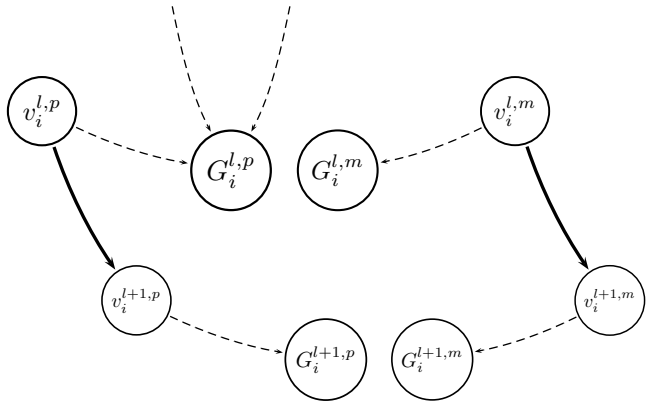

Figure 10

Coupling of the loci. Adjacent loci are coupled only through the meiosis indicators $v_i^{l,p}$ and $v_i^{l,m}$ of the individuals, as indicated by the solid lines. Dashed arrows represent links between nodes on the same locus.

Collecting all conditional probability tables, the definition of the full probability distribution is given by:

$$\begin{aligned} P(\mathbf{T},\mathbf{M},\mathbf{v},\mathbf{G}\mid \mathbf{m},\mathbf{t},\mathbf{f},\theta,\lambda_T) &= \\ \prod_{i\in \mathrm{F,NF}} P\left(T_i \mid \mathbf{G}_i^{l_T(\lambda_T)},\mathbf{f}\right) \prod_{l\neq l_T(\lambda_T)} P\left(\mathbf{M}_i^l \mid \mathbf{G}_i^l\right) &\times \\ \prod_{i\in \mathrm{NF}} \prod_l P\left(\mathbf{v}_i^l \mid \mathbf{v}_i^{l-1},\theta_{l,l-1}(\lambda_T)\right) P\left(\mathbf{G}_i^l \mid \mathbf{v}_i^l,\mathbf{G}_{\pi(i)}^l\right) &\times \\ \prod_{i\in \mathrm{F}} P\left(\mathbf{G}_i^{l_T(\lambda_T)} \mid \mathbf{t}\right) \prod_{l\neq l_T(\lambda_T)} P(\mathbf{G}_i^l \mid \mathbf{m}^l) & \end{aligned} \quad (3)$$

Here the founder and non-founder individuals are denoted by F and NF, respectively. The index of the trait locus depends on the position of the trait locus, so that $l_T = l_T(\lambda_T)$. The recombination frequency $\theta_{l,l'}$ depends on λ_T if either $l = l_T$ or $l' = l_T$, so that $\theta_{l,l'} = \theta_{l,l'}(\lambda_T)$. Otherwise, $\theta_{l,l'}$ is independent of λ_T.

This distribution is normalized to one, by construction. Computing marginal distributions is generally intractable because the structure of the corresponding Bayesian network can be too complex due to loops. Loops are caused by inbreeding in the pedigree and through the coupling of the meiosis indicators between different loci.

Calculating LOD scores

The LOD score of parametric linkage analysis is defined as the log ratio of the likelihood that the trait locus is linked to the marker loci at location λ_T and the likelihood that the trait locus is unlinked, denoted by $\lambda_T = \infty$:

$$\mathrm{LOD}(\lambda_T \mid \mathbf{f},\mathbf{t},\mathbf{m},\theta) = \log_{10}\left[\frac{P(\mathbf{T},\mathbf{M}\mid \mathbf{f},\mathbf{t},\mathbf{m},\theta,\lambda_T)}{P(\mathbf{T},\mathbf{M}\mid \mathbf{f},\mathbf{t},\mathbf{m},\theta,\lambda_T=\infty)}\right].$$

The denominator can be rewritten as

$$P(\mathbf{T},\mathbf{M}\mid \mathbf{f},\mathbf{t},\mathbf{m},\theta,\lambda_T=\infty) = P(\mathbf{T}\mid \mathbf{f},\mathbf{t})\,P(\mathbf{M}\mid \mathbf{m},\theta),$$

giving

$$\mathrm{LOD}(\lambda_T \mid \mathbf{f},\mathbf{t},\mathbf{m},\theta) = \log 10\left[\frac{P(\mathbf{T}\mid \mathbf{M},\mathbf{f},\mathbf{t},\mathbf{m},\theta,\lambda_T)}{P(\mathbf{T}\mid \mathbf{f},\mathbf{t})}\right]. \quad (4)$$

The denominator has to be computed only once and acts as a normalization constant. We will use the Cluster Variation Method to approximate both likelihoods independently. The likelihood in the numerator has to be estimated for each position of the trait locus.

The Cluster Variation Method

In this section we describe how the Cluster Variation Method can be used to obtain approximations of marginal distributions of the exact distribution. In order to apply the Cluster Variation Method more conveniently, we will make a slight change in notation.

The probability distribution of a Bayesian network is of the general form

$$P(\mathbf{x}) = \prod_i P(x_i \mid \mathbf{x}_{\pi(i)}),$$

Where $\pi(i)$ are the nodes with a link directed towards node i, x_i is the value assumed by node i and $\mathbf{x}$ is a vector of values assumed by all nodes in the Bayesian network. If there are no nodes that have a link pointing to node i, we have

$$P(x_i|\mathbf{x}_{\pi(i)}) = P(x_i).$$

We consider evidence to be the observation that a node is clamped to a state, e.g. an individual is affected or has marker genotype (1, 2). Suppose we have evidence that node n is clamped to state x_n^e, denoted by $e = \{x_n = x_n^e\}$, then

$$\begin{aligned} P(e = \{x_n = x_n^e\}) &= \sum_{\mathbf{x}} P(x_i = x_n^e \mid \mathbf{x}_{\pi(i)} \\ &\times \prod_{i\neq n} P(x_i \mid \mathbf{x}_{\pi(i)}) \end{aligned}$$

For genetic linkage analysis the evidence is on marker genotypes **M** and trait phenotypes **T**, and we wish to compute the likelihood of these observations given model parameters.

We now define

$$\psi_i(x_i,\mathbf{x}_{\pi(i)}) = \begin{cases} P(x_i \mid \mathbf{x}_{\pi(i)}): & \text{no evidence} \\ P(x_i \mid \mathbf{x}_{\pi(i)})\delta(x_i - x_i^e): & e_i = \{x_i = x_i^e\}, \end{cases}$$

and $\mathbf{e} = \{e_1, \ldots, e_n\}$. Here $\delta(\cdot)$ is the delta function, which serves to clamp a node to its observed value. Using these definitions, we can rewrite the likelihood of the evidence **e** as

$$P(\mathrm{e}) = \sum_{\mathbf{x}} \prod_i \psi_i(x_i,\mathbf{x}_{\pi(i)}).$$

so that the probability distribution over nodes without evidence $\mathbf{x}_{i\backslash e}$ conditional on nodes with evidence $\mathbf{x}_e$ is given by

$$P(\mathbf{x}_{i\backslash e} \mid \mathbf{x}_e) = \frac{1}{P(\mathbf{e})} \prod_i \psi_i(x_i, \mathbf{x}_{\pi(i)}). \qquad (5)$$

We now have reformulated the probability distribution of the Bayesian network in terms of so called *potential funtions* ψ which do not reflect any longer how the links between the nodes were originally directed. Also, the potential functions $\psi_i(x_i, \mathbf{x}_{\pi(i)})$ contain both node i and the parents of node i. As a result, in the undirected graph associated with these potential functions all parents of node i are connected to each other. This formal procedure is called moralization [7] and is essential to the application of all inference methods to Bayesian networks. Equation 5 specifies the same distribution as equation 3, but will be more convenient to apply the Cluster Variation Method to.

Obtaining the exact distribution from a variational principle

The exact distribution P can be derived from a variational principle:

$$P = \arg\min_P \mathrm{KL}(P \,||\, \Psi) = \sum_{\mathbf{x}} P(\mathbf{x}_{i\backslash e} \mid \mathbf{x}_e) \log \frac{P\left(\mathbf{x}_{i\backslash e} \mid \mathbf{x}_e\right)}{\Psi(\mathbf{x})} \qquad (6)$$

subject to the constraint that P is normalized to one, where

$$\Psi(\mathbf{x}) = \prod_i \psi_i(x_i, \mathbf{x}_{\pi(i)}), Z = \sum_{\mathbf{x}} \Psi(\mathbf{x}),$$

and KL is the Kullback-Leibler divergence. The solution is readily given by $P = \frac{1}{Z}\Psi$. However, the sum in equation 6 is over an exponential number of states, and is generally intractable.

At this point one can make various choices in making an approximation to the optimization problem defined in equation 6. The CVM approximation fits into this framework as follows. The approach is to replace P with the product

$$Q = \prod_\alpha Q_\alpha(\mathbf{x}_\alpha) \prod_\beta Q_\beta^{a_\beta}(\mathbf{x}_\beta). \qquad (7)$$

This expression deserves some explanation and motivation. The labels $\alpha \in B$ run over the set of *basic* clusters B. A cluster α consists of a set of nodes $\mathbf{x}_\alpha = \{ x_{\alpha_1}, x_{\alpha_2}, ..., x_{\alpha_n} \}$ which can in principle be chosen freely. However, the idea is to choose them such that the corresponding marginal distributions $Q_\alpha(\mathbf{x}_\alpha)$ are tractable for exact computation. Essential to the CVM approximation is that the clusters α are defined on overlapping subsets of nodes: a single node, or even a subset of nodes, can occur in several of the basic clusters $\alpha \in B$. Although the product[2] of the cluster marginals may not be a good approximation of the full distribution, the approximation is designed such that the cluster marginals $Q_\alpha(\mathbf{x}_\alpha)$ are accurate approximations of the exact marginals $P(\mathbf{x}_\alpha)$.

From the set of the basic clusters B follows the definition of the set of clusters M. The set M contains all clusters that can be constructed by taking intersections of basic clusters $\alpha \in B$, intersections of intersections of basic clusters $\alpha \in B$, and so forth. Defining U as $B \cup M$, the coefficients α_β are defined by

$$a_\beta = 1 - \sum_{\gamma \supset \beta \in U} a_\gamma, \qquad (8)$$

where $\alpha_\gamma = 1, \forall_\gamma \in B$. These coefficients are known as the *Moebius* numbers or over counting numbers.

How can the form of the distribution of equation 7 and the coefficients of equation 8 be justified? If the Bayesian network has no loops[3] then the exact distributions is of the form 7 with $\alpha_i = \{i, \pi(i)\}, B = \{\alpha_i\}$. If the Bayesian network does have loops, this is not true. However, due to the evidence, many variables become effectively independent and a choice of the basic clusters B exists such that the approximate marginal distributions $Q_\alpha(\mathbf{x}_\alpha)$ are very close to the exact marginals $P(\mathbf{x}_\alpha)$.

Figure 11 shows an example Bayesian network and a choice of clusters indicated by dotted lines, specifying a particular CVM approximation. We have the variables corresponding to the paternal (p) and maternal (m) gene of the founder individuals 1 and 2 and the child 3, for both locus 1 and 2; the index i of the individual is subscripted, the index l of locus is superscripted: $G_i^{l,p}$ and $G_i^{l,m}$. Then we have the paternal and maternal meiosis indicators of individual 3, also for both loci: $v_i^{l,p}$ and $v_i^{l,m}$ respectively. In this example, we have chosen the following clusters $\alpha \in B$ that will determine the approximation:

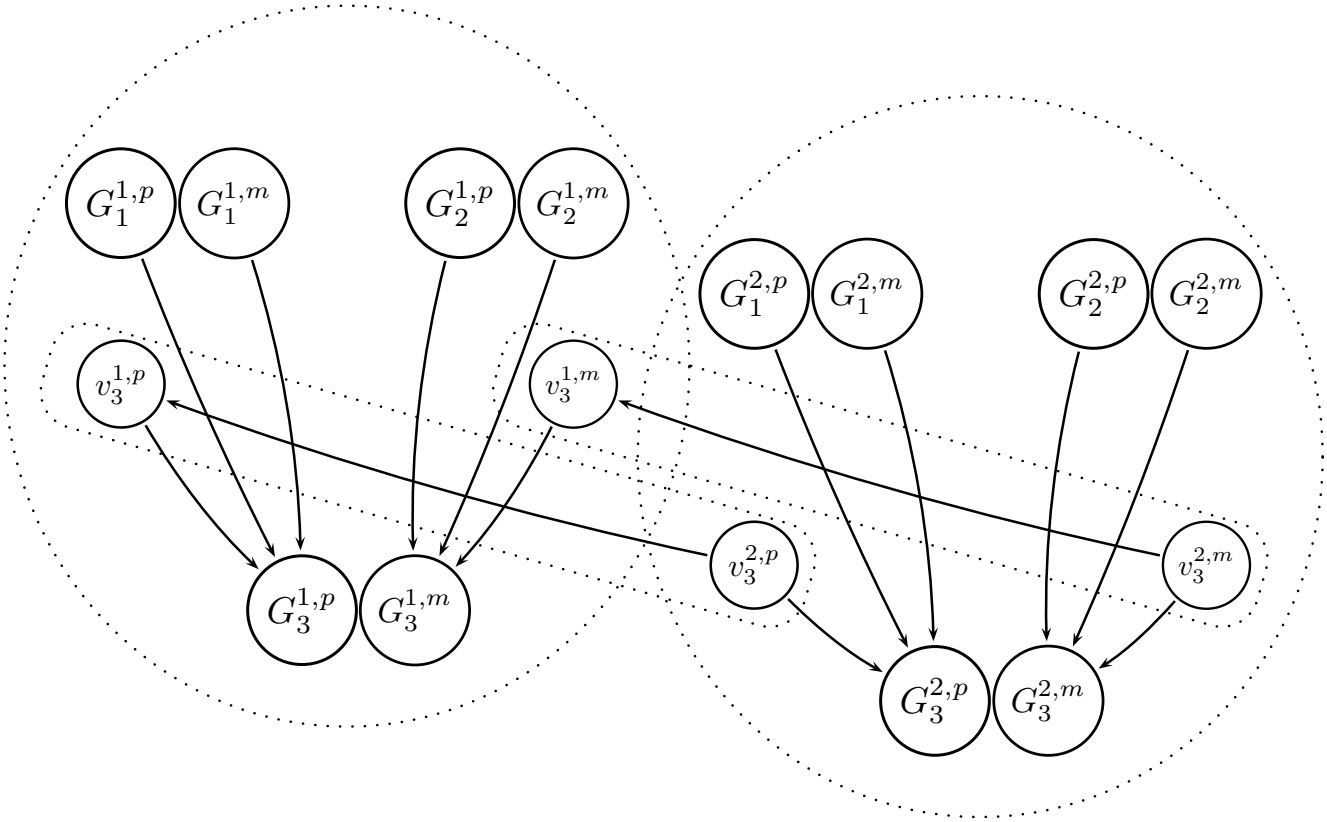

Figure 11
Example Bayesian network. Clusters are indicated by the dotted lines.

$$\alpha_1 = \left\{G_3^{1,p}, G_3^{1,m}, G_1^{1,p}, G_1^{1,m}, G_2^{1,p}, G_2^{1,m}, v_3^{1,p}, v_3^{1,m}\right\},$$
$$\alpha_2 = \left\{G_3^{2,p}, G_3^{2,m}, G_1^{2,p}, G_1^{2,m}, G_2^{2,p}, G_2^{2,m}, v_3^{2,p}, v_3^{2,m}\right\},$$
$$\alpha_3 = \left\{v_3^{1,p}, v_3^{2,p}\right\},$$
$$\alpha_4 = \left\{v_3^{1,m}, v_3^{2,m}\right\}.$$

Clusters α_1 and α_2 contain all variables of locus 1 and 2 respectively; clusters α_3 and α_4 contain the paternal and maternal meiosis indicators respectively that link the two loci. These clusters have the following intersections $\beta \in M$:

$$\beta_1 = \left\{v_3^{1,p}\right\}, \beta_2 = \left\{v_3^{2,p}\right\},$$
$$\beta_3 = \left\{v_3^{1,m}\right\}, \beta_4 = \left\{v_3^{2,m}\right\}.$$

In this example there are no intersections of intersections of the basic clusters $\alpha \in B$. This choice of the clusters leads to the following expression for equation 7:

$$\frac{Q_{\alpha_1}\left(\mathbf{G}_1^1, \mathbf{G}_2^1, \mathbf{G}_3^1, \mathbf{v}_3^1\right) Q_{\alpha_2}\left(\mathbf{G}_1^2, \mathbf{G}_2^2, \mathbf{G}_3^2, \mathbf{v}_3^2\right)}{Q_{\beta_1}\left(v_3^{1,p}\right) Q_{\beta_2}\left(v_3^{2,p}\right)} \times \frac{Q_{\alpha_3}\left(v_3^{1,p}, v_3^{2,p}\right) Q_{\alpha_4}\left(v_3^{1,m}, v_3^{2,m}\right)}{Q_{\beta_3}\left(v_3^{1,m}\right) Q_{\beta_1}\left(v_3^{2,m}\right)}.$$

Approximate free energies

We now discuss the optimization problem of equation 6, which is to be redefined in terms of the cluster marginals. Inserting the CVM approximation corresponding to equation 7 into expression 6, we obtain

$$\left\{Q_\gamma\right\} \approx \arg\min_Q \mathrm{KL}(Q \,||\, \Psi) = \arg\min_{Q_\gamma} F_{\mathrm{CVM}}(Q) \equiv \arg\min_{Q_\gamma} \sum_{\gamma \in U} a_\gamma F_\gamma(Q_\gamma), \quad (9)$$

where the minimization is subject to normalization constraints

$$\sum_{\mathbf{x}_\gamma} Q_\gamma(\mathbf{x}_\gamma) = 1,$$

and consistency constraints

$$\sum_{\mathbf{x}_\gamma \setminus \mathbf{x}_{\gamma'}} Q_\gamma(\mathbf{x}_\gamma) = Q_{\gamma'}(\mathbf{x}_{\gamma'}) \quad \forall \gamma \supset \gamma'.$$

Again, a_γ are the Moebius numbers. The consistency constraints ensure that if two clusters have a non-empty intersection, the marginal distributions on the nodes in the overlap are consistent.

In equation 9 we have introduced the *free energy* $F_\gamma(Q_\gamma)$ of cluster $\gamma \in U$:

$$F_\gamma(Q_\gamma) = \sum_{\mathbf{x}_\gamma} Q_\gamma(\mathbf{x}_\gamma) \log \frac{Q_\gamma(\mathbf{x}_\gamma)}{\Psi_\gamma(\mathbf{x}_\gamma)},$$

where

$$\Psi_\gamma(\mathbf{x}_\gamma) = \prod_{\{i,\pi(i)\} \subset \gamma} \psi_i(\mathbf{x}_i, \mathbf{x}_{\pi(i)}).$$

Ψ_γ contains all conditional probability tables that are defined on subsets of nodes in cluster γ. The optimization problem in equation 9 is now in terms of the distributions $Q_\gamma(\mathbf{x}_\gamma)$, which are tractable by choice. The intractable optimization problem of equation 6 has been turned into a tractable optimization problem, by substituting the exact distribution P the product Q defined in equation 7. In the next section we will discuss how the optimization problem of equation 9 can be solved efficiently.

Returning to the example of figure 11, we can now write down the corresponding free energy which is to be minimized with respect to the cluster marginals $Q_\alpha(\mathbf{x}_\alpha)$ and $Q_\beta(\mathbf{x}_\beta)$:

$$\begin{aligned} F_{\text{cvm}}(Q_\alpha, Q_\beta) \;=\; & +1 \cdot F_{\alpha_1}(Q_{\alpha_1}) + 1 \cdot F_{\alpha_2}(Q_{\alpha_2}) \\ & +1 \cdot F_{\alpha_3}(Q_{\alpha_3}) + 1 \cdot F_{\alpha_4}(Q_{\alpha_4}) \\ & -1 \cdot F_{\beta_1}(Q_{\beta_1}) - 1 \cdot F_{\beta_2}(Q_{\beta_2}) \\ & -1 \cdot F_{\beta_3}(Q_{\beta_3}) - 1 \cdot F_{\beta_4}(Q_{\beta_4}). \end{aligned}$$

Here the Moebius numbers are in boldface. The minimization is subject to normalization and consistency constraints. For example, the consistency constraint between clusters α_1 and β_1:

$$\sum_{\mathbf{G}_{3'}^1 \mathbf{G}_{1'}^1 \mathbf{G}_{2'}^1 \boldsymbol{\nu}_3^{1,m}} Q_{\alpha_1}\left(\mathbf{G}_{3'}^1 \mathbf{G}_{1'}^1 \mathbf{G}_{2'}^1 \boldsymbol{\nu}_3^{1,m}, \boldsymbol{\nu}_3^{1,p}\right) = Q_{\beta_1}\left(\boldsymbol{\nu}_3^{1,p}\right)$$

In the example, we have put all meiosis nodes into different clusters. In practice this gives inaccurate approximations; it turns out to be neccessary to join all paternal and maternal meiosis indicators in one cluster, because of the strong correlations between these variables. The reason is that if the phase in one of the genotypes of the parents is reversed, for a given state of the meiosis indicators different alleles are transmitted to the children.

Minimizing the CVM free energy

Minimizing the CVM free energy is difficult, since the functional $F_{\text{CVM}}(Q)$ is high-dimensional and generally non-convex. Yedidia et al. [13] derived an inference algorithm based on the Cluster Variation Method, called Generalized Belief Propagation (GBP). This fixed point iteration algorithm is not guaranteed to converge, because of the non-convexity of the CVM free energy. Convergent algorithms were proposed by Rangarajan et al. [26] and Teh et al. [27] and more recently by Heskes et al. [28]. These so-called double loop algorithms minimize $F_{\text{CVM}}(Q_\gamma)$ by iteratively improving a *convex* upper bound on the non-convex functional $F_{\text{CVM}}(Q_\gamma)$ that can be minimized by fixed point iteration. The double loop algorithm always converges to a (local) minimum of the free energy.

We use the double loop approach described in [28]. Although single loop algorithms [13] in some cases may converge, often they require damping of the fixed point equations and it can be difficult to find a good trade-off between efficiency and robustness of the algorithm. Double loop algorithms can be slower when single loop algorithms converge, but the setting of the parameters of the double loop algorithm is less critical and convergence is guaranteed in theory.

We will give an outline of the algorithm; for full details we refer to [28]. The starting point is the issue of the non-convexity of $F_{\text{CVM}}(Q)$. The free energy of each cluster, $F_\gamma(Q_\gamma(\mathbf{x}_\gamma))$, is convex in terms of the approximate marginals $Q_\gamma(\mathbf{x}_\gamma)$. This can be seen by writing it out:

$$F_\gamma\left(Q_\gamma(\mathbf{x}_\gamma)\right) = Q_\gamma(\mathbf{x}_\gamma) \log \Psi_\gamma(\mathbf{x}_\gamma) + Q_\gamma(\mathbf{x}_\gamma) \log Q_\gamma(\mathbf{x}_\gamma) \equiv -E(Q_\gamma) - S(Q_\gamma).$$

Here we have introduced the energy $E(Q_\gamma)$ and entropy $S(Q_\gamma)$. These names stem from statistical physics, where the Cluster Variation Method is used to compute properties of certain metals that can be described as systems of interacting magnetic spins. The energy term is linear in the marginal distribution Q_γ. By differentiation it can be seen that the minus entropy has a positive second derivative, and therefore $-S(Q_\gamma)$ is convex.

We now take a look at the CVM free energy again and identify the convex and concave terms:

$$F_{\mathrm{CVM}}(Q) = \sum_{\alpha} F_{\alpha}(Q_{\alpha}) + \sum_{\beta \in M^{+}} a_{\beta} F_{\beta}(Q_{\beta}) - \sum_{\beta \in M^{-}} |a_{\beta}| F_{\beta}(Q_{\beta}). \quad (10)$$

Here M^{+} is the set of clusters $\beta \in M : a_{\beta} > 0$ and $M^{-} \in M : a_{\beta} < 0$. Since all free energies are convex, clusters $\beta \in M^{-}$ with negative Moebius numbers have concave contributions to the total free energy $F_{\mathrm{CVM}}(Q)$, which therefore becomes non-convex.

The double loop algorithm is based on the following idea. Since the fixed point iterations as employed in GBP converge if the free energy is convex, a convergent algorithm can be constructed by iteratively minimizing and improving convex upper bounds to the CVM free energy. Let's denote the convex upper bound by $F_{\mathrm{conv}}(Q, Q')$. Define $\mathcal{Q}$ as the collection of marginal distributions (i.e. cluster marginals) that are normalized and satisfy all constistency constraints between overlapping marginal distributions. Following [28], if the upper bound is at least twice differentiable and satisfies the following properties:

1. $F_{\mathrm{conv}}(Q, Q') \geq F_{\mathrm{CVM}}(Q)\ \forall\ Q, Q' \in \mathcal{Q}$

2. $F_{\mathrm{conv}}(Q, Q) = F_{\mathrm{CVM}}(Q)\ \forall\ Q \in \mathcal{Q}$

3. $F_{\mathrm{conv}}(Q, Q')$ is convex in $Q \in \mathcal{Q}$, $\forall\ Q' \in \mathcal{Q}$,

then the algorithm

$$Q_{n+1} = \mathrm{argmin}_{Q \in \mathcal{Q}} F_{\mathrm{conv}}(Q, Q_n),$$

with Q_n the approximate marginals at iteration n, is guaranteed to converge to a local minimum of the CVM free energy $F_{\mathrm{CVM}}(Q)$ under the appropriate constraints. The free energy decreases with each iteration, since

$$F_{\mathrm{CVM}}(Q_{n+1}) \leq F_{\mathrm{conv}}(Q_{n+1}, Q_n) \leq F_{\mathrm{conv}}(Q_n, Q_n) = F_{\mathrm{CVM}}(Q_n),$$

where the first inequality follows from condition 1 (upper bound) and the second from the definition of the algorithm. Condition 2 (touching) in combination with differentiability ensures that the algorithm is only stationary in points where the gradient of F_{CVM} is zero. By construction, $Q_n \in \mathcal{Q}$ for all n.

A convex upper bound can be obtained easily by bounding the concave contributions to the free energy. Since the energy term in the free energy of each cluster is already convex, only the concave entropy terms of clusters with negative Moebius number need to be bounded. A convex upper bound on a concave entropy term can be achieved by linearizing it:

$$S_{\beta}(Q_{\beta}) = -\sum_{\mathbf{x}_{\beta}} Q_{\beta}(\mathbf{x}_{\beta}) \log Q_{\beta}(\mathbf{x}_{\beta}) \leq -\sum_{\mathbf{x}_{\beta}} Q_{\beta}(\mathbf{x}_{\beta}) \log Q'_{\beta}(\mathbf{x}_{\beta}) \equiv S_{\beta}(Q_{\beta}, Q'_{\beta}),$$

which directly follows from $\mathrm{KL}(Q_{\beta}, Q'_{\beta}) \geq 0$. Putting this into expression 10, we obtain for the convex upper bound:

$$F_{\mathrm{conv}}(Q, Q') = \sum_{\alpha \in B} E_{\alpha}(Q_{\alpha}) - \sum_{\alpha} S_{\alpha}(Q_{\alpha}) + \sum_{\beta \in M^{+}} a_{\beta} E_{\beta}(Q_{\beta}) + \sum_{\beta \in M^{-}} a_{\beta} E_{\beta}(Q_{\beta}) - \sum_{\beta \in M^{+}} a_{\beta} S_{\beta}(Q_{\beta}) - \sum_{\beta \in M^{-}} a_{\beta} S_{\beta}(Q_{\beta}, Q'_{\beta}). \quad (11)$$

We now see that the both the energy $E_{\beta}(Q_{\beta})$ and the bounded entropy $S_{\beta}(Q_{\beta}, Q'_{\beta})$ are linear in the cluster marginal $Q_{\beta}(\mathbf{x}_{\beta})$. We can therefore simplify expression 11 by redefining the energies of the basic clusters $\alpha \in B$:

$$-\tilde{E}_{\alpha}(Q_{\alpha}) \equiv -\sum_{\mathbf{x}_{\alpha}} Q_{\alpha}(\mathbf{x}_{\alpha}) \log \Psi_{\alpha}(\mathbf{x}_{\alpha}) - \sum_{\beta \in M \subset \alpha} a_{\beta} \sum_{\mathbf{x}_{\beta}} Q_{\beta}(\mathbf{x}_{\beta}) \log \Psi_{\beta}(\mathbf{x}_{\beta}) + \sum_{\beta \in M^{-} \subset \alpha} a_{\beta} \sum_{\mathbf{x}_{\beta}} Q_{\beta}(\mathbf{x}_{\beta}) \log Q'_{\beta}(\mathbf{x}_{\beta})$$

The convex upper bound becomes

$$F_{\mathrm{conv}}(Q, Q') = \sum_{\alpha \in B} \tilde{E}_{\alpha}(Q_{\alpha}) - \sum_{\alpha} S_{\alpha}(Q_{\alpha}) - \sum_{\beta \in M^{+}} a_{\beta} S_{\beta}(Q_{\beta}). \quad (12)$$

This upper bound can be minimized using the single loop algorithm described in [13].

Thus, the double loop algorithm consists of an outer loop and an inner loop:

Outer loop : compute convex upper bound 12 with $Q' = Q_n$;

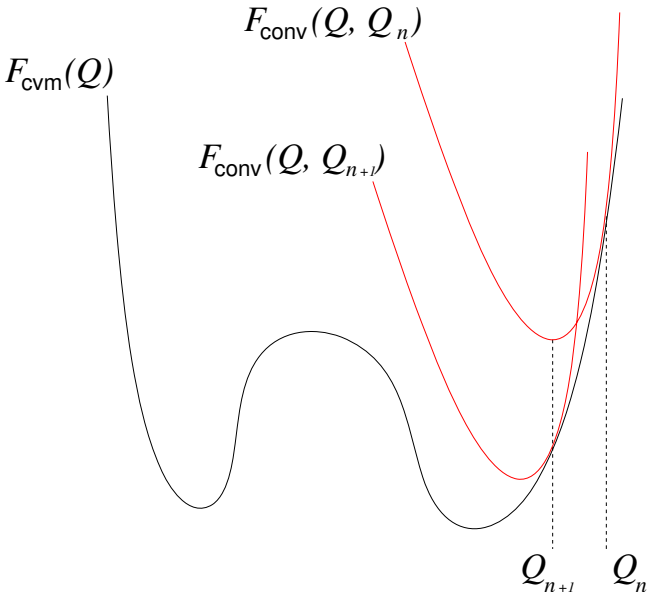

Figure 12
Illustration of the double loop algorithm. At iteration n + 1, in the outer loop the convex upper bound $F_{conv}(Q, Q_{n+1})$ to $F_{CVM}(Q)$ is computed, touching $F_{CVM}(Q)$ at Q_n. The unique minimum of the convex upper bound is reached using a single loop fixed point iteration scheme in the inner loop and is attained at Q_{n+1}. At this point an outer loop and an inner loop of the double loop algorithm have been completed, and a new upper bound to $F_{CVM}(Q)$ at Q_1 is computed in iteration n + 2.

Inner loop : minimize $F_{conv}(Q, Q')$, using single loop fixed point interations, yielding Q_{n+1}.

The procedure is illustrated in figure 12.

We have described the double loop algorithm with the case where all subclusters with negative Moebius numbers are bounded. However it is possible to bound convex entropy contributions of sub-clusters with positive Moebius numbers as well. This tends to sharpen the bound because then the bounding of the convex entropy terms counters the effect of bounding concave entropy terms. An advantage of this bound is that the inner loop iteration scheme becomes much simpler. This is the bound that we have used for the simulations in this article. Even tighter bounds can be obtained by not bounding *all* concave entropy terms such that F_{conv} is convex on the constraint subset. We refer to [28] for more details on the specific conditions.

Applying the Cluster Variation Method to linkage analysis
In this section we describe how we apply the Cluster Variation Method to estimate LOD scores. We outline the algorithm for the case that the pedigrees is not inbred, which is the case for which we have performed simulations.

From the definition of the Bayesian network in equation 3 it follows that the exact likelihood of the phenotypes **T** conditional on the marker genotypes **M** can always be rewritten as:

$$P(\mathbf{T} \mid \mathbf{M}, \mathbf{f}, \mathbf{t}, \mathbf{m}, \theta, \lambda_T) = \sum_{\mathbf{v}_T, \mathbf{v}_M, \mathbf{G}_T, \mathbf{G}_M} P(\mathbf{T}, \mathbf{G}_T \mid \mathbf{v}_T, \mathbf{f}, \mathbf{t}) \times P(\mathbf{v}_T \mid \mathbf{v}_M, \theta, \lambda_T) P(\mathbf{v}_M, \mathbf{G}_M \mid \mathbf{M}, \mathbf{m}, \theta) \quad (13)$$

The first factor on the right hand side concerns the likelihood of the trait data given an inheritance vector $\mathbf{v}_T$ on the trait locus. The second factor is a distribution over trait locus inheritance vectors, conditional on marker loci inheritance vectors, $\mathbf{v}_M$. The last term is the distribution over marker loci inheritance vectors conditional on the marker data **M**, the marker allele frequencies and the recombination frequencies θ specified by the marker map. Essentially this decomposition is possible because the marker and trait loci are connected only through the meiosis indicators; the model can be viewed as a Hidden Markov Model where the meiosis indicators **v** are the hidden variables.

Outline The decomposition of the probability distribution $P(\mathbf{T}|\mathbf{M}, \mathbf{f}, \mathbf{t}, \mathbf{m}, \theta, \lambda_T)$ of equation 13 is central to our approach. We now give an outline of the algorithm and then discuss each step of the algorithm in more detail.

1. The first step of the algorithm is to make the following approximation with the Cluster Variation Method:

$$P(\mathbf{v}_M, \mathbf{G}_M \mid \mathbf{M}, \mathbf{m}, \theta) \approx \prod_{\gamma \in U} Q_\gamma^{a_\gamma}(\mathbf{x}_\gamma), \quad (14)$$

where $Q_\gamma(\mathbf{x}_\gamma)$ are the approximate marginal distributions over clusters $\gamma \in U = B \cup M$, and α_γ the corresponding Moebius numbers. The marginals Q_γ are obtained with the double loop algorithm described in the previous section. This step is performed only once; the trait locus has no part in this approximation.

2. In the second step, the likelihood of trait data is computed for each location of the trait locus, using the approximate distribution over inheritances on the marker loci:

for each position of the trait locus λ_T:

$$P(\mathbf{T} \mid \mathbf{M}, \mathbf{f}, \mathbf{t}, \mathbf{m}, \theta, \lambda_T) \approx \sum_{\mathbf{v}_T, \mathbf{v}_M, \mathbf{G}_T, \mathbf{G}_M} P(\mathbf{T}, \mathbf{G}_T \mid \mathbf{v}_T, \mathbf{f}, \mathbf{t}) \times P(\mathbf{v}_T \mid \mathbf{v}_M, \theta, \lambda_T) \prod_{\gamma} Q_\gamma^{a_\gamma}(\mathbf{x}_\gamma), \quad (15)$$

where we have substituted equation 14 into equation 13. The CVM approximation expressed by the product in equation 14 has the consequence that if the pedigree is not inbred, the calculations involved in step 15 can be performed efficiently.

3. Finally, the LOD scores for each location λ_T are given by equation 4:

$$\mathrm{LOD}(\lambda_T \mid \mathbf{f}, \mathbf{t}, \mathbf{m}, \theta) = \log_{10}\left[\frac{P(\mathbf{T} \mid \mathbf{M}, \mathbf{f}, \mathbf{t}, \mathbf{m}, \theta, \lambda_T)}{P(\mathbf{T} \mid \mathbf{f}, \mathbf{t})}\right].$$

Step 1 Simulations indicate that approximating the full distribution $P(\mathbf{T}, \mathbf{M} \mid \mathbf{f}, \mathbf{t}, \mathbf{m}, \theta, \lambda_T)$ with the Cluster Variation Method can give bad results when the inheritance implied by the trait data **T** is very different from the inheritance implied by the marker data **M** for a given location of the trait locus. Therefore we choose to approximate the distribution over marker loci inheritance vectors $P(\mathbf{v}_M, \mathbf{G}_M \mid \mathbf{M}, \mathbf{m}, \theta)$ independently of the trait data **T**.

In terms of the conditional probability tables of the Bayesian network, we have

$$P(\mathbf{v}_M, \mathbf{G}_M, \mathbf{M} \mid \theta, \mathbf{m}) = \prod_{i \in \mathrm{F,NF}} \prod_{l} P(\mathbf{M}_i^l \mid \mathbf{G}_i^l) \times \prod_{i \in \mathrm{NF}} \prod_{l} P(\mathbf{v}_i^l \mid \mathbf{v}_i^{l-1}, \theta_{l,l-1}) P(\mathbf{G}_i^l \mid \mathbf{v}_i^l, \mathbf{G}_{\pi(i)}^l) \times \prod_{i \in \mathrm{F}} \prod_{l} P(\mathbf{G}_i^l \mid \mathbf{m}^l). \quad (16)$$

Here the subscript l runs only over marker loci. This is the multi-locus Bayesian network described previously, but without the trait locus. We make the CVM approximation

$$P(\mathbf{v}_M, \mathbf{G}_M \mid \mathbf{M}, \mathbf{m}, \theta) \approx \prod_{\gamma \in U} Q_\gamma^{a_\gamma}(\mathbf{x}_\gamma),$$

Now consider for the remainder of this section the case where the set of basic clusters B consists of the clusters

$$\alpha_{i,l} = \left\{\mathbf{G}_{\mathbf{c}_i}^l, \mathbf{v}_{\mathbf{c}_i}^l, \mathbf{G}_{\pi_i}^l, \mathbf{G}_{\mathbf{c}_i}^{l+1}, \mathbf{v}_{\mathbf{c}_i}^{l+1}, \mathbf{G}_{\pi_i}^{l+1}\right\}, \quad (17)$$

where $\mathbf{c}_i$ are the children in nuclear family i in the pedigree, π_i are the parents in nuclear family i, **G** and **v** are the corresponding genotype and meiosis nodes and $(l, l+1)$ represent two adjacent marker loci in the marker map. The subscript i runs over all nuclear families in the pedigree, and the subscript $l = (1, \ldots, L_M - 1)$, with L_M the number of marker loci. This is exactly cluster choice C_1 of figure 5A.

Given the conditional probability tables that define the Bayesian network in equation 16, the approximate marginals can be obtained by minimizing the CVM free energy $F_{\mathrm{CVM}}(Q)$ of equation 9 corresponding to the clusters defined in expression 17. The minimization is done using the double loop algorithm.

Step 2 In this step the likelihood of trait data is computed using the approximate distribution over inheritance vectors $\mathbf{v}_M$ on the marker loci. This computation entails the summation

$$P(\mathbf{T} \mid \mathbf{M}, \mathbf{f}, \mathbf{t}, \mathbf{m}, \theta, \lambda_T) \approx \sum_{\mathbf{v}_T, \mathbf{v}_M, \mathbf{G}_T, \mathbf{G}_M} P(\mathbf{T}, \mathbf{G}_T \mid \mathbf{v}_T, \mathbf{f}, \mathbf{t}) P(\mathbf{v}_T \mid \mathbf{v}_M, \theta, \lambda_T) \prod_{\gamma} Q_\gamma^{a_\gamma}(\mathbf{x}_\gamma). \quad (18)$$

Because of factorization assumed by the CVM approximation with the clusters defined in expression 17, this computation can be done efficiently. Suppose we would like to calculate the likelihood of the trait data **T** for the case where the trait locus is located between the fourth and fifth marker:

$$\ldots M_3 \rightarrow M_4 \rightarrow T \leftarrow M_5 \leftarrow M_6 \ldots . \quad (19)$$

Given the marginals $Q_{\gamma \in U^{l_4, l_5}}$, we can derive a distribution over inheritance vectors $\mathbf{v}_T$ on the trait locus. This is possible because the meiosis events on the trait locus are not directly observed, but only indirectly through the observed trait data **T**. We define $\tilde{Q}_\alpha$ through the relation

$$Q_\alpha(\mathbf{v}^l, \mathbf{G}^l, \mathbf{v}^{l+1}, \mathbf{G}^{l+1}) \equiv \tilde{Q}_\alpha(\mathbf{v}^l, \mathbf{G}^l, \mathbf{v}^{l+1}, \mathbf{G}^{l+1}) \prod_{i \in \alpha} P(\mathbf{v}_i^l \mid \mathbf{v}_i^{l+1}, \theta).$$

Here Q_α is the approximate marginal computed with CVM only on the adjacent marker loci $(l, l+1)$ and $P(\mathbf{v}_i^l \mid \mathbf{v}_i^{l+1}, \theta)$ is the conditional probability table that defines the coupling between the meiosis indicators in the

Bayesian network. As both of these terms are known, together they define $\tilde{Q}_\alpha$. We can now define a distribution over trait locus inheritance vectors as follows:

$$Q'_\alpha\left(\mathbf{v}^l,\mathbf{G}^l,\mathbf{v}^T,\mathbf{v}^{l+1},\mathbf{G}^{l+1}\middle|\lambda_T\right)\equiv$$
$$\tilde{Q}_\alpha\left(\mathbf{v}^l,\mathbf{G}^l,\mathbf{v}^{l+1},\mathbf{G}^{l+1}\right)\times$$
$$\prod_i P\left(\mathbf{v}_i^l\middle|\mathbf{v}_i^T,\theta,\lambda_T\right)P\left(\mathbf{v}_i^T\middle|\mathbf{v}_i^{l+1},\theta,\lambda_T\right).$$

Summing over all states of the trait locus meiosis indicators $\{\mathbf{v}_i^T\}$ yields again Q_α:

$$\sum_{\mathbf{v}^T}Q'_\alpha\left(\mathbf{v}^l,\mathbf{G}^l,\mathbf{v}^T,\mathbf{v}^{l+1},\mathbf{G}^{l+1}\middle|\lambda_T\right)=$$
$$Q_\alpha\left(\mathbf{v}^l,\mathbf{G}^l,\mathbf{v}^{l+1},\mathbf{G}^{l+1}\right). \quad (20)$$

As a result we have an effective distribution over trait locus inheritance vectors defined by the following product. For the example where the trait locus is located between marker 4 and marker 5, it is given by:

$$P\left(\mathbf{v}_T,\mathbf{v}_M^{l_4,l_5},\mathbf{G}_M^{l_4,l_5}\middle|\mathbf{M},\mathbf{m},\theta,\lambda_T\right)\approx$$
$$\prod_{\alpha\in B^{l_4,l_5}}Q'_\alpha\left(\mathbf{x}_\alpha\right)\prod_{\beta\in M^{l_4,l_5}}Q_\beta(\mathbf{x}_\beta).$$

Note that the marginal distributions of the intersections are the unprimed Q_β, since we have equality 20 and the fact that the trait meiosis nodes are not contained in any intersection of the basic clusters $\alpha \in B$. If the pedigree is not inbred, this product defines a proper probability distribution.

The summation in equation 18 now becomes

$$P\left(\mathbf{T}\middle|\mathbf{M},\mathbf{f},\mathbf{t},\mathbf{m},\theta,\lambda_T\right)\approx$$
$$\sum_{\mathbf{v}_T,\mathbf{v}_M^{l(\lambda_T)},\mathbf{G}_T,\mathbf{G}_M^{l(\lambda_T)}}P\left(\mathbf{T},\mathbf{G}_T\middle|\mathbf{v}_T,\mathbf{f},\mathbf{t}\right)\times$$
$$\prod_{\alpha\in B^{l(\lambda_T)}}Q'_\alpha\left(\mathbf{x}_\alpha\right)\prod_{\beta\in M^{l(\lambda_T)}}Q_\beta\left(\mathbf{x}_\beta\right), \quad (21)$$

where we have defined $l(\lambda_T)$ as the pair of markers flanking the trait locus to the left and right. We now observe that if the pedigree is not inbred, the summation involved in equation 21 can be performed efficiently with the junction tree algorithm. If the pedigree is too inbred, then this last step can be done using an additional CVM approximation.

Step 3 The last step is straightforward once the likelihood of the trait has been computed for every location of the trait locus in the second step.

A heuristic for detecting inaccurate approximations

The approach we have outlined here allows for a heuristic that indicates whether the approximation of the trait data likelihood conditional on the approximate marginals over marker loci is not accurate. We compute the likelihood of the trait data for a given location of the trait locus from the marginals on the marker loci flanking the trait locus. However, if the trait locus is located at the exact position of a marker, there are two possibilities for the marker loci $l(\lambda_T)$. Suppose the trait locus is at marker l_3, then one could take either the marginals defined on marker loci (l_2, l_3) or the marginals defined on marker loci (l_3, l_4). The likelihood of the trait data must be the same for either choice. However this only holds if the approximation is valid. Therefore, in the case that the trait locus is located at a marker, we compute the LOD score for both options to detect a possible inconsistency, indicating an inaccurate approximation. Conversely, we cannot guarantee that if there is no inconcistency, the approximation *is* accurate.

Preprocessing

Currently we apply three preprocessing steps.

1. The phase of the genotypes of the founders can be clamped on one marker locus as this does not change the likelihood.

2. In this step, genotypic configurations (assignments of alleles to the genotypes of the individuals) that are not consistent with the observed marker alleles **M**, are removed from the cluster potentials ψ_α, $\alpha \in B$. First we run the double loop algorithm on each marker locus seperately. Then some states $\mathbf{x}_\alpha$ in the cluster marginals $Q_\alpha(\mathbf{x}_\alpha)$ will be assigned zero probability, because the corresponding genotypic configuration is not consistent with the marker genotypes observed for that locus.

As an example consider a bi-allelic marker. If both parents have genotype (1, 1), then the children cannot have the genotype (1, 2). Consequently, any state in a cluster marginal which corresponds to a child having genotype (1, 2) will have zero probability and does not contribute to the likelihood. These states can therefore be removed from the potentials.

3. Nodes that are not in any intersection of the basic clusters B can be integrated out from the potentials ψ_α before running the double loop algorithm. If individuals are not genotyped this can give substantial reductions in the

number of states per cluster marginal that have to be stored in memory.

Authors' contributions

CAA developed the method, performed the simulations and prepared the manuscript. MARL and HJK co-developed the method and supervised the project.

Notes

[1]It is also possible to choose the parameterization $P(v_i^{l,m} \mid v_i^{l+1,m}, \theta_{l,l+1})$, which corresponds to reversing the direction of the links. This choice is equivalent.

[2]We explicitly do not say that Q is also a probability distribution, because normalization of this product is not guaranteed. This is the price that is paid for having a tractable optimization problem of the form of equation 9. The reason that normalization of the product cannot be guaranteed is that the clusters $\alpha \in B$ are defined on subsets of variables that are not disjoint. Thus computing the normalization constant of the product of marginals is as complex as computing the exact likelihood $P(\mathbf{e})$. It is possible to define a factorization in terms of disjoint subsets of variables. This factorization can be guaranteed to be normalized, since normalization of the marginal distributions on disjoint subsets of variables ensures normalization of the product of the marginal distributions. Such an approximation does not fit into the framework of the Cluster Variation Method. Exactly because the subsets of variables are disjoint and correlations between variables in disjoint subsets are neglected, the approximation tends to be less powerful.

[3]By following the links (ignoring direction) there is only one path from one node to another.

Acknowledgements

We would like to thank Han Brunner and Tom Heskes for useful discussions.

References

1. Morton NE: **Sequential tests for the detection of linkage.** *Am J Hum Genet* 1955, **7:**277-318.
2. Lander ES, Green P: **Construction of multilocus genetic linkage maps in humans.** *Proc Natl Acad Sci U S A* 1987, **84:**2363-2367.
3. Kruglyak L, Daly MJ, Reeve-Daly MP, Lander ES: **Parametric and non-parametric linkage analysis: A unified multipoints approach.** *Am J Hum Genet* 1996, **58:**1347-1363.
4. Elston RC, Stewart J: **A general model for the analysis of pedigree data.** *Hum Hered* 1971, **21:**523-542.
5. O'Connell JR: **Rapid multipoint linkage analysis via inheritance vectors in the Elston-Stewart algorithm.** *Hum Hered* 2001, **51:**226-240.
6. Lauritzen SL, Spiegelhalter D: **Local computations with probabilities on graphical structures and their application to expert systems.** *J Royal Statistical Society B* 1988, **50:**154-227.
7. Jensen : *An introduction to Bayesian networks* Edited by: *Lewin LA. UCL Press*; 1996.
8. Fishelson J, Geiger D: **Exact genetic linkage computations for general pedigrees.** *Bioinformatics* 2002, **18:**S189-S198.
9. Bethe HA: *Proc Roy Soc London* 1935, **150:**552-575.
10. Kikuchi R: **A Theory of cooperative phenomena.** *Physical Review* 1951, **81:**988.
11. An G: **A note on the Cluster Variation Method.** *Journal of Statistical Physics* 1988, **52:**727-734.
12. Morita T: **Cluster Variation Method and Moebius inversion formula.** *Journal of Statistical Physics* 1990, **59:**819-825.
13. Yedidia JS, Freeman WT, Weiss Y: **Generalized Belief Propagation.** *Advances in Neural Information Processing Systems 13* 2001:689-695.
14. Yedidia JS, Freeman WT, Weiss Y: **Constructing Free-Energy approximations and Generalized Belief Propagation algorithms.** *IEEE Transactions on Information Theory* 2005, **51:**2282-2312.
15. Thompson EA, George AW: **Discovering disease genes: multipoint linkage analyses via a new markov chain Monte Carlo approach.** *Statistical Science* 2003, **18:**515-535.
16. Lange K, Sobel E: **Descent graphs in pedigree analysis: application to haplotyping, location scores, and marker-sharing statistics.** *Am J Hum Genet* 1996, **58:**1323-1337.
17. Sobel E, Sengul H, Weeks DE: **Multipoint estimation of identity-by-descent probabilities at arbitrary positions among marker loci on general pedigrees.** *Hum Hered* 2001, **52:**121-131.
18. Lange K, Goradia TM: **An algorithm for automatic genotype elimination.** *Am J Hum Genet* 1987, **40:**250-256.
19. Du FX, Hoeschele I: **A note on algorithms for genotype and allele elimination in complex pedigrees with incomplete genotype data.** *Genetics* 2000, **156:**2051-2062.
20. Friedman N, Geiger D, Lotner N: **Likelihood computation with value abstraction.** *Proceedings of Uncertainty in AI* 2000:192-200.
21. Chavira M, Allen D, Darwiche A: **Exploiting evidence in probabilistic inference.** *Proceedings of Uncertainty in AI* 2005:112-119.
22. Pearl J: *Probabilistic reasoning in intelligent systems: networks of plausible inference Morgan Kaufmann Publishers Inc*; 1988.
23. Jensen C, Kong A: **Blocking-Gibbs sampling for linkage analysis in large pedigrees with many loops.** *Am J Hum Genet* 1999, **65:**885-902.
24. Thomas A, Abkevich V, Bansai A: **Multilocus linkage analysis by blocked Gibbs sampling.** *Statistics and Computing* 2000, **10:**259-269.
25. Sheehan NA, Lauritzen SL: **Graphical models for genetic analyses.** *Statistical Science* 2003, **4:**489-514.
26. Rangarajan A, Yuille AL: **The convex-concave principle.** *Advances in Neural Information Processing Systems 14* 2002:1033-1040.
27. Welling M, Teh Y: **The Unified Propagation and Scaling algorithm.** *Advances in Neural Information Processing Systems 14* 2002:953-960.
28. Heskes T, Albers CA, Kappen HJ: **Approximate inference and constrained optimization.** *Proceedings of Uncertainty in AI* 2003:313-320.