Edwards et al. BMC Bioinformatics 2010, 11:18
http://www.biomedcentral.com/1471-2105/11/18

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

Selecting high-dimensional mixed graphical models using minimal AIC or BIC forests

David Edwards*, Gabriel CG de Abreu, Rodrigo Labouriau

Abstract

Background: Chow and Liu showed that the maximum likelihood tree for multivariate discrete distributions may be found using a maximum weight spanning tree algorithm, for example Kruskal's algorithm. The efficiency of the algorithm makes it tractable for high-dimensional problems.

Results: We extend Chow and Liu's approach in two ways: first, to find the forest optimizing a penalized likelihood criterion, for example AIC or BIC, and second, to handle data with both discrete and Gaussian variables. We apply the approach to three datasets: two from gene expression studies and the third from a genetics of gene expression study. The minimal BIC forest supplements a conventional analysis of differential expression by providing a tentative network for the differentially expressed genes. In the genetics of gene expression context the method identifies a network approximating the joint distribution of the DNA markers and the gene expression levels.

Conclusions: The approach is generally useful as a preliminary step towards understanding the overall dependence structure of high-dimensional discrete and/or continuous data. Trees and forests are unrealistically simple models for biological systems, but can provide useful insights. Uses include the following: identification of distinct connected components, which can be analysed separately (dimension reduction); identification of neighbourhoods for more detailed analyses; as initial models for search algorithms with a larger search space, for example decomposable models or Bayesian networks; and identification of interesting features, such as hub nodes.

Background

Recent years have seen intense interest in representing complex biological systems as networks, and a new research discipline, network biology, has arisen. In particular, Markov networks and Bayesian networks have been applied in many domains [1-3]. The former are based on undirected graphs, and the latter on DAGs (directed acyclic graphs). A key challenge in deriving such networks from the high-dimensional data typical of the genomics era is computational efficiency: model selection algorithms that perform well for small or moderate dimensions may be intractable for high dimensions. The approach of Chow and Liu [4], which predates much of the development of probabilistic graphical models, is particularly efficient, being quadratic in the number of variables.

The Chow-Liu algorithm

Suppose that we have a dataset with N observations of p discrete random variables $X = (X_v)_{v\epsilon\Delta}$. We call the possible values a discrete variable may take its *levels*, and label these 1,...$|X_v|$, so that $|X_v|$ is the number of levels of X_v. We write a generic observation (or *cell*) as $x = (x_1,..., x_p)$, and the set of possible cells as χ. We assume that the observations are independent and are interested in modelling the probabilities $p(x) = \Pr(X = x)$ for $x \epsilon \chi$.

Suppose also that the cell probabilities factorize according to a tree, that is, a connected acyclic graph, written $\mathcal{X} = (X, E)$ where X is the vertex set and E the set of edges. That is to say, the cell probabilities can be written $p(x) = \prod_{e\epsilon E} g_e(x)$ for functions $g_e(x)$ that only depend on the variables in e. So when $e = (X_u, X_v)$, $g_e(x)$ is a function of x_u and x_v only. Chow and Liu [4] showed that the cell probabilities take the form

* Correspondence: David.Edwards@agrsci.dk
Institute of Genetics and Biotechnology, Faculty of Agricultural Sciences, Aarhus University, Aarhus, Denmark

$$p(x) = \frac{\prod_{(u,v)\in E} \Pr(x_u, x_v)}{\prod_{v\in V} \Pr(x_v)^{d_v - 1}} \quad (1)$$

$$= \prod_{v\in V} \Pr(x_v) \prod_{(u,v)\in E} \frac{\Pr(x_u, x_v)}{\Pr(x_u)\Pr(x_v)} \quad (2)$$

where d_v is the degree of v, that is, the number of edges incident to v. Hence up to a constant the maximized log-likelihood is $\sum_{(u,\ v)\epsilon E} I_{u,\ v}$, where $I_{u,\ v}$ is given by

$$I_{u,v} = \sum_{x_u, x_v} n(x_u, x_v) \ln \frac{n(x_u, x_v)}{n(x_u) n(x_v)},$$

$n(x_u, x_v)$ being the number of observations with $X_u = x_u$ and $X_v = x_v$. The quantity $I_{u,\ v}$ is called the *mutual information*. It follows that if we use the $I_{u,\ v}$ as edge weights on the complete graph with vertex set X, and apply a maximum spanning tree algorithm, we obtain the maximum likelihood tree.

In statistical terms, $I_{u,\ v}$ is one half of the usual likelihood ratio test statistic for marginal independence of X_u and X_v, that is G^2 = -2 ln Q = $2I_{u,\ v}$, calculated using the table of counts $\{n(x_u, x_v)\}$ formed by cross-tabulating X_u and X_v. Under marginal independence G^2 has an asymptotic $\chi^2_{(k)}$ distribution, where k = ($|X_u|$ - 1)($|X_v|$ - 1). The degrees of freedom k is the number of additional free parameters required under the alternative hypothesis, compared with the null hypothesis.

A very similar exposition can be given for multivariate Gaussian data: here the sample mutual information is

$$I_{u,v} = -N \ln(1 - \hat{\rho}^2_{u,v}) / 2,$$

where $\hat{\rho}_{u,v}$ is the sample correlation between X_u and X_v. As before the likelihood ratio test statistic G^2 = -2 ln Q = $2I_{u,\ v}$. Under marginal independence G^2 has a $\chi^2_{(1)}$ distribution.

Algorithms to find the maximum weight spanning tree of a arbitrary undirected connected graph $\mathcal{W}$ with positive edge weights have been studied thoroughly. The following simple and efficient algorithm is due to Kruskal [5]. Starting with the null graph, repeat this step: among the edges not yet chosen, add the edge with the largest weight that does not form a cycle with the ones already chosen. When p - 1 edges have been added, the maximum weight spanning tree of $\mathcal{W}$ has been found. The algorithm can be implemented to run in $O(p^2 \ln p)$ time.

As mentioned above, $\mathcal{W}$ is here taken to be the complete graph on X with edge weights given by $\{I_{u,\ v}\}_{u,\ v\epsilon X}$. In practice the task of calculating these $p(p - 1)/2$ edge weights dominates the time usage, so the complexity of the Chow-Liu algorithm may be taken to be $O(p^2)$. Methods to improve computational efficiency have been described [6,7].

Chow and Liu's approach has been extended to more general classes of graphs than trees: to thin junction trees [8]; to polytrees [9]; to bounded tree-width networks [10], and to mixtures of trees [11]. The approach has also been extended to tree-based models for Gaussian processes [12] and discrete-valued time series [13]. The consistency of the algorithm has been shown [14].

Results and Discussion

Extension to minimal AIC/BIC forests

A disadvantage with selecting a tree based on maximum likelihood is that it will always include the maximum number of edges, irrespective of whether the data support this or not. It is desirable to take account of the number of model parameters in some fashion. In the machine learning literature it is customary to penalize the likelihood using the minimum description length principle [15], whereas in the statistical literature the use of information criteria is well-established, particularly AIC (the Akaike information criterion [16]) and BIC (the Bayesian information criterion [17]). The former is defined as -2 ln L + 2r, where L is the maximized likelihood under the model and r is the number of parameters in the model, and the latter as -2 ln L + ln(N)r. Discussions of the relative merits of these criteria are available [18] and need not be repeated here.

First, suppose that Kruskal's algorithm is applied using penalized mutual information quantities $I^{AIC}_{u,v} = I_{u,\ v} - k_{u,\ v}$ or $I^{BIC}_{u,v} = I_{u,\ v} - \ln(N)k_{u,\ v}/2$, where $k_{u,\ v}$ is the degrees of freedom associated with $I_{u,\ v}$, as described above. Then it is easily seen that the tree with the minimum AIC or BIC is obtained. Note that for Gaussian data this will be identical to the maximum likelihood tree, since all edges have the same degrees of freedom. For discrete data with varying numbers of levels, the maximum likelihood tree and the minimal AIC/BIC tree will generally differ.

Second, given a graph $\mathcal{W}$ = (V, E) with both positive and negative edge weights, consider the problem of finding the maximum weight forest, that is, the acyclic subgraph on vertex set V with maximum weight. Let $\mathcal{W}'$ be the graph derived from $\mathcal{W}$ by omitting all edges with negative weights. For any forest with vertex set V, removing all edges with negative weights would increase the total weight and not introduce any cycles. It follows that we can construct the maximum weight forest by finding the maximum weight spanning tree for each connected component of $\mathcal{W}'$. We can do this simply by

applying Kruskal's algorithm to $\mathcal{W}'$: it is not necessary to find the connected components explicitly.

So it is easy to find the minimal AIC or BIC forest by using penalized mutual information quantities as weights. This approach is attractive with high-dimensional data, since if the selected forest does consist of multiple connected components these may then be analyzed separately – allowing a dimension reduction. We show below that the connected components of the minimal AIC/BIC forest are also connected components of the minimal AIC/BIC decomposable model, providing further justification for this procedure.

That using penalized likelihood with the Chow-Liu algorithm leads to forests rather than trees appears to be known in the machine learning literature [19]; also, [20] finds the Bayesian MAP tree/forest in a similar way, but we have found no published references in the computational biology or statistics research literatures. We believe that it is a useful method that deserves to be far more widely known.

A numerical illustration

Here we compare application of the algorithms to some simulated data involving three discrete random variables, X_a, X_b and X_c with 2, 5, and 5 levels respectively, and whose joint distribution is given by

$$\Pr(x_a, x_b, x_c) = \Pr(x_a)\Pr(x_b \mid x_a)\Pr(x_c \mid x_a)$$

where $\Pr(x_a) = (0.5,\ 0.5)'$,

$$\Pr(x_b \mid x_a) = \begin{pmatrix} 0.8 & 0.025 & 0.025 & 0.05 & 0.1 \\ 0.1 & 0.05 & 0.025 & 0.025 & 0.8 \end{pmatrix}$$ and

either (i) $$\Pr(x_c \mid x_a) = \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.3 & 0.1 \end{pmatrix}$$ or

(ii) $$\Pr(x_c \mid x_a) = \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{pmatrix}.$$

Note that X_a and X_b are strongly associated but there is weak or no association between X_a and X_c.

Figure 1 shows the corresponding independence graphs: in case (i), $\mathcal{G}_1$, and in case (ii), $\mathcal{G}_2$. A random dataset with 500 observations was drawn from each of the joint distributions and the algorithms applied. This was repeated 1000 times. The results are shown in Table 1.

Table 1 Simulation Results

	Case (i)			Case (ii)		
Algorithm	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_3$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_3$
ML tree	826	0	174	5	0	995
min AIC forest	1000	0	0	94	897	9
min BIC forest	995	5	0	0	1000	0

In case (i), the ML tree algorithm incorrectly identifies $\mathcal{G}_3$ about 17% of time; otherwise it correctly identifies $\mathcal{G}_1$. Penalizing with AIC or BIC increases the success frequencies to almost 100%. In case (ii) the true model $\mathcal{G}_2$ is a forest rather than a tree, so the ML tree algorithm cannot select it. Note that it almost always selects $\mathcal{G}_3$: since $2I_{b,\,c} \sim \chi^2_{(16)}$ and $2I_{a,\,c} \sim \chi^2_{(4)}$, the former is almost always greater than the latter. Penalizing using AIC and BIC increases the success frequencies to 90% and 100%, respectively. For insight into the relative performance of AIC and BIC in this example, see [18].

Extension to mixed discrete and Gaussian data

The second extension we consider is to data with both discrete and Gaussian variables. Our approach uses the class of undirected mixed graphical models [21-23]. Consider a data set with N observations of p discrete random variables $X = (X_1,\ldots X_p)$, and q continuous random variables $Y = (Y_1,\ldots Y_q)$. The models are based on the conditional Gaussian distribution, that is to say, the conditional distribution of Y given $X = x$ is multivariate Gaussian with mean, and possibly also variance, depending on x. Models in which the variance depends on x are termed heterogenous, otherwise, they are called homogeneous.

Tree (or forest) dependence models can be defined as mixed graphical models whose independence graphs are trees (or forests). But since their likelihood functions do not in general factorize according to (2) the theory does not carry through directly. To obtain the analogous factorization, we restrict attention to those models that have explicit maximum likelihood estimates, the so-called strongly decomposable models [21,22,24]. These are easily characterized. A mixed graphical model is strongly decomposable if and only if it is triangulated (that is, contains no chordless cycles of length greater or equal to four) and contains no *forbidden paths* [22]. See Figure 2.

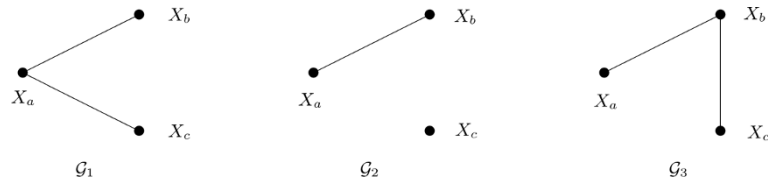

Figure 1 Graphs connected with the simulations. Data were simulated from $\mathcal{G}_1$, in case (i), and from $\mathcal{G}_2$, in case (ii). The third graph $\mathcal{G}_3$ is sometimes selected by the algorithms.

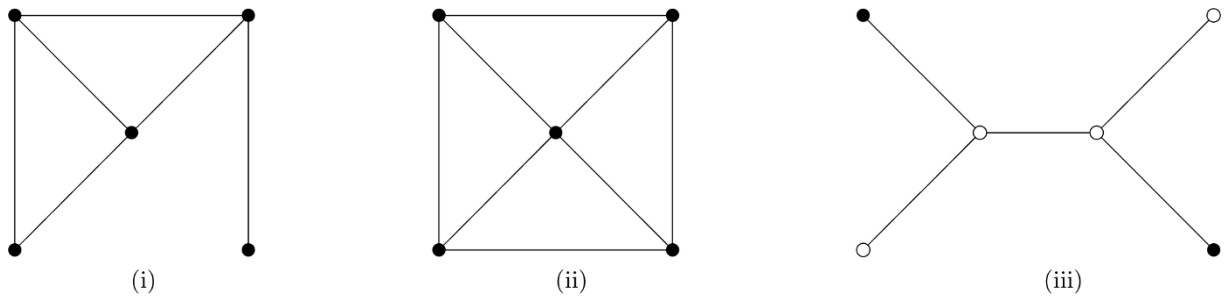

Figure 2 Three undirected graphs. Graph (i) is triangulated, that is, contains no chordless cycles of length four or greater. Graph (ii) is not triangulated, since it contains a chordless cycle of length four. Graph (iii) contains both discrete nodes (dots) and continuous nodes (circles). It is triangulated and contains a forbidden path.

A forbidden path is a path between two non-adjacent discrete vertices passing through continuous vertices. Since trees and forests are acyclic, they are triangulated, and since they contain at most one path between any two vertices, we can simplify the criterion as follows: A tree or forest dependence model is strongly decomposable if and only if it contains no path between discrete vertices passing through continuous vertices. We call such a tree (or forest) an SD-tree (or SD-forest). In an SD-tree the discrete vertices induce a connected subgraph.

To apply the algorithm we need to derive the mutual information between a discrete variable X_u and a continuous variable Y_v. The marginal model is a simple ANOVA model (section 4.1.7 of [21]). Let $s_0 = \Sigma_k(y^{(k)} - \bar{y})^2/N$, and write the sample cell counts, means and variances as $\{n_i, \bar{y}_i, s_i\}_{i=1...|X_u|}$. In the homogeneous case, the mutual information is $I_{u,v} = N \ln(s_0/s)/2$, where $s = \sum_{i=1,...|X_u|} n_i s_i / N$. There are $k = |X_u| - 1$ degrees of freedom. In the heterogeneous case, the mutual information is $I_{u,v} = N \ln(s_0)/2 - \sum_{i=1,...|X_s|} n_i \ln(s_i) / 2$, with $k = 2(|X_u| - 1)$ degrees of freedom. The expressions given here assume that all parameters are estimable: when this is not so, they need to be modified slightly, but we omit the details.

We also need to modify Kruskal's algorithm. As before an undirected graph $\mathcal{W}$ with positive weights is given. Starting with the null graph, we repeatedly add the edge with the largest weight that does not form a cycle or a forbidden path. It is shown below that this returns the maximum weight SD-forest.

About the forbidden path restriction

We describe here a perspective on the forbidden path restriction that gives useful insight. Graphical models encode sets of conditional independence relations, and if two graphical models encode the same set of conditional independence relations they are termed *Markov equivalent* [25,26]. For example, each graph in Figure 3 represents the conditional independence of X_a and X_c given X_b. Sample data from the joint distribution of X_a, X_b and X_c supply information on which conditional independence relations hold and which do not, but cannot distinguish between the four graphs. To do this would require intervention in the system, for example by perturbing X_a to see whether the distribution of X_b is altered. For this reason algorithms to identify Bayesian networks from sample data [27,28] can only do this up to Markov equivalence.

The DAGs that are Markov equivalent to a given tree comprise a Markov equivalence class. As illustrated in Figure 4, they are easily found. Labelling a node (X_r, say) as a root and orienting all edges away from the root, induces a single-parent DAG, that is, one in which all nodes have at most one parent. Any node can be chosen as root. Under such a DAG, the joint distribution factorizes into

$$p(x) = \Pr(x_r) \prod_{u \neq r} \Pr(x_u \mid pa(x_u)),$$

where $\mathrm{pa}(x_u)$ denotes the parents (here, parent) of x_u in the DAG. Models corresponding to the DAG are constructed by specifying a marginal distribution $\Pr(x_r)$ and a series of conditional models for $\Pr(x_u|\mathrm{pa}(x_u))$.

First consider the *pure* case, that is, when all variables are either discrete or continuous. In the discrete case, we can construct a model for the DAG by specifying a multinomial distribution for X_r and arrays of transition probabilities for the conditional models. In the continuous case, X_r is Gaussian and the conditional models are simple linear regressions. When X_u and X_v are both discrete or both continuous, the mutual information $I_{u,v}$ is symmetric, and is consistent with the conditional models for both $\Pr(x_v|x_u)$ and $\Pr(x_u|x_v)$. It follows that a DAG model in the Markov equivalence class is essentially a reparametrization of the tree model, and so has the same maximized likelihood and

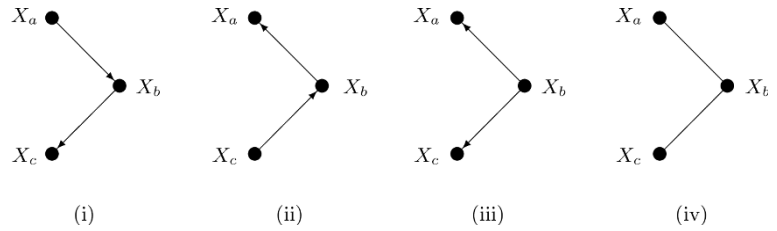

Figure 3 Markov equivalence. The first three graphs are DAGs, the fourth is undirected. All four graphs represent the same conditional independence relation: that X_a and X_c are conditionally independent given X_b. They are called *Markov equivalent*.

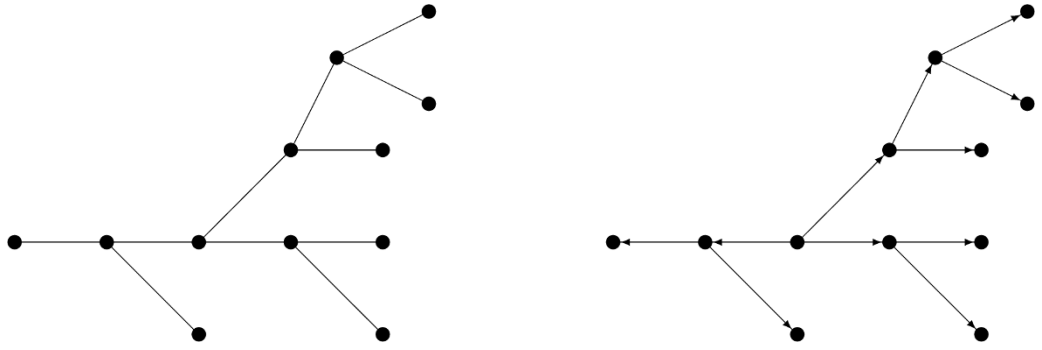

Figure 4 A tree and a rooted tree. Specifying a root generates a single-parent DAG.

penalized likelihood scores. So in the pure case the algorithm identifies a Markov equivalence class of DAGs, just like other Bayesian network selection algorithms. Note that the search space is restricted to single-parent DAGs.

In the *mixed* case, however, the mutual information between a discrete X_u and a continuous X_v is asymmetric, and corresponds to an ANOVA-type conditional model for $\Pr(x_v|x_u)$ but not for $\Pr(x_u|x_v)$. So a DAG model in the Markov equivalence class is a reparametrization of the tree model only if the DAG contains no edges pointing from continuous to discrete nodes. If the tree has a forbidden path, no such DAG will exist: see for example Figure 2(iii). If the tree has no forbidden paths, then a DAG generated in the above way will have this property if and only if its root is discrete. So in the mixed case the algorithm identifies a subset of a Markov equivalence class of DAGs, those generated using discrete roots. That only a subset is identified is due to a limitation of the model apparatus, not to any evidence in the data. The limitation is unproblematic provided that the discrete variables are prior to the continuous variables.

All this has two broad implications. The first is that, when interpreted causally, the tree and forest models allow at most one determinant of each variable. The second is that the approach implicitly assumes that discrete variables are prior to continuous ones.

A marginality property

In some cases the global optimality of the selected model holds under marginalization. The following result is shown below in the methods section. Suppose that $\mathcal{G}$ is the maximum likelihood tree (or minimal AIC or BIC forest) for a variable set V and let the connected components of $\mathcal{G}$ be C_1,... C_k, say. Then $\mathcal{G}_A$ (the marginal subgraph induced by $A \subseteq V$) is the maximum likelihood tree (respectively, minimal AIC or BIC forest) for the variable set A provided that $\mathcal{G}_{A\cap C_i}$ is connected, for each component C_i.

For example, consider a genetics of gene expression study involving a set of discrete DNA markers Δ and a set of continuous gene expression variables Γ. A central tenet is that DNA can affect gene expression but not vice versa. Suppose that the minimal AIC/BIC forest for $V = (\Delta, \Gamma)$ is $\mathcal{G}$. The forbidden path restriction implies that for each connected component C_i of $\mathcal{G}$, $\mathcal{G}_{\Delta\cap C_i}$ is connected. Hence $\mathcal{G}_\Delta$ is the minimal AIC/BIC forest for the discrete data alone. It follows that $\mathcal{G}$ can be regarded as a chain graph model [22] with two blocks, Δ and Γ, with Δ prior to Γ, consistent with the tenet.

Some applications of the algorithm

We show the results of applying the algorithm to three datasets.

Study of leucine-responsive protein (Lrp) in E. coli

The first dataset stems from a previously reported gene expression study [29]. The stated purpose of this was to

identify the network of genes that are differentially regulated by the global *E. coli* transcription factor, leucine-responsive regulatory protein (Lrp), during steady state growth in a glucose supplemented minimal salts medium. Lrp has been reported to affect the expression of approximately 236 genes [30]. Gene expression in two *E. coli* bacteria strains, labelled lrp+ and lrp-, were compared using eight Affymetrix ecoli chips. The lrp+ strain is the control or wild type, and the lrp- strain is the experimental type, with the Lrp gene knocked-out. Four chips were hybridized with RNA from the lrp+ strain, and four chips with RNA from the lrp- strain. The raw data were preprocessed using standard methods and the algorithm applied to the derived data. The dataset had $N = 8$ observations and 7313 variables, comprising 7312 continuous variables (the log-transformed gene expression values) and one discrete variable, strain.

Our implementation of the algorithm (see below) took about 2 minutes on a laptop running Windows XP to find the minimal BIC forest. This is too large to display here, so instead we examine an interesting subgraph.

Figure 5 shows the radius eight neighbourhood of strain, that is to say the subgraph of vertices whose path length from strain is less than or equal to 8. There are three variables adjacent to strain. The short arm links to the knockout gene itself via an intergenic region (IG) tRNA gene. This arm just reflects the marked downregulation of Lrp in the knockout strain. The other two arms suggest that Lrp targets just two genes, serA and gltD. It is instructive to compare Figure 4 with a conventional analysis of differential expression using the limma library [31]. If a false discovery rate of 0.2 is used, 40 genes are flagged as possibly differentially regulated. Although the two analysis approaches are very different – limma is based on gene-by-gene hypothesis testing, and is concerned with the operating characteristics of this, while the present approach is based on approximating the joint distribution of the entire variable set – the results are broadly consistent. Of the 40 genes identified by the limma analysis, 35 have a path length less or equal to 8 to strain in the minimum BIC forest, and so appear in Figure 5. The remaining 5 genes, however, are very distant from strain, with path lengths ranging from 59 to 81. This could suggest that their apparent regulation by Lrp is spurious.

The regulatory system of *E. coli* has been well-studied, and it is interesting to note that other studies confirm that serA and gltD are targets of Lrp [30,32]. Indeed, Lrp has many targets: 138 Lrp-binding sites have been identified [30], so it is certainly not true that Lrp only targets serA and gltD. We have not been able to find other reports that the five distant genes – ndk, pnt, ptsG, nupG and atpG – should be directly or indirectly regulated by Lrp.

The minimal BIC forest provides a provisional causal model for the effect of Lrp, and in this sense more directly addresses the stated goal of the study than a conventional analysis of differential expression. However, given the small number of observations in the study, it is clear that the network identification and any interpretations based on this are highly uncertain.

Gene expression profiling in breast cancer patients

The second dataset comes from another gene expression study [33], whose purpose was to compare the gene expression profiles in tumours taken from two groups of breast cancer patient, those with and those without a mutation in the p53 tumour suppression gene. A dataset containing a subset of the study data is supplied along with the R library gRbase. The dataset has $N = 250$ observations and 1001 variables, comprising 1000 continuous variables (the log-transformed gene expression values) and the class variable. There are 58 cases (with a p53 mutation) and 192 controls (without the mutation). The gene expression variables were filtered from a larger set, and all exhibit differential expression between the two groups. They have been standardized to zero mean and unit variance, but since the mixed graphical models used here are location and scale invariant, this does not affect the analysis.

The algorithm took about 18 seconds to find the minimal BIC forest. Figure 6 shows the radius seven neighbourhood of the class variable. The graph suggests that the effect of the p53 mutation on the gene expression profile is mediated by its effect on the expression of a gene with column number 108. This gene is CDC20, a gene involved in cell division. To examine this hypothesis more critically we could apply a richer class of models to this neighbourhood of genes, but that would take us outside the scope of this paper. Figure 6 also shows some apparent hub nodes, including 209 (GPR19), 329 (BUB1), 213 (CENPA), 554 (C10orf3) and 739 (CDCA5), that appear to play a key role in the system. See table 2 of [33] for further information on p53 - associated genes.

Genetics of gene expression using HapMap data

The third dataset comes from a large multinational project to study human genetic variation, the HapMap project http://www.hapmap.org/. The dataset concerns a sample of 90 Utah residents with northern and western European ancestry, the so-called CEU population, and contains information on genetic variants and gene expression values for this sample. The subjects are not unrelated (they comprise parent-sibling trios), but the analysis ignores this. The genetic variants are SNPs (single nucleotide polymorphisms). Datasets containing both genomic and gene expression data enable study of the the genetic basis for differences in gene expression. This dataset is supplied along with the R library GGtools.

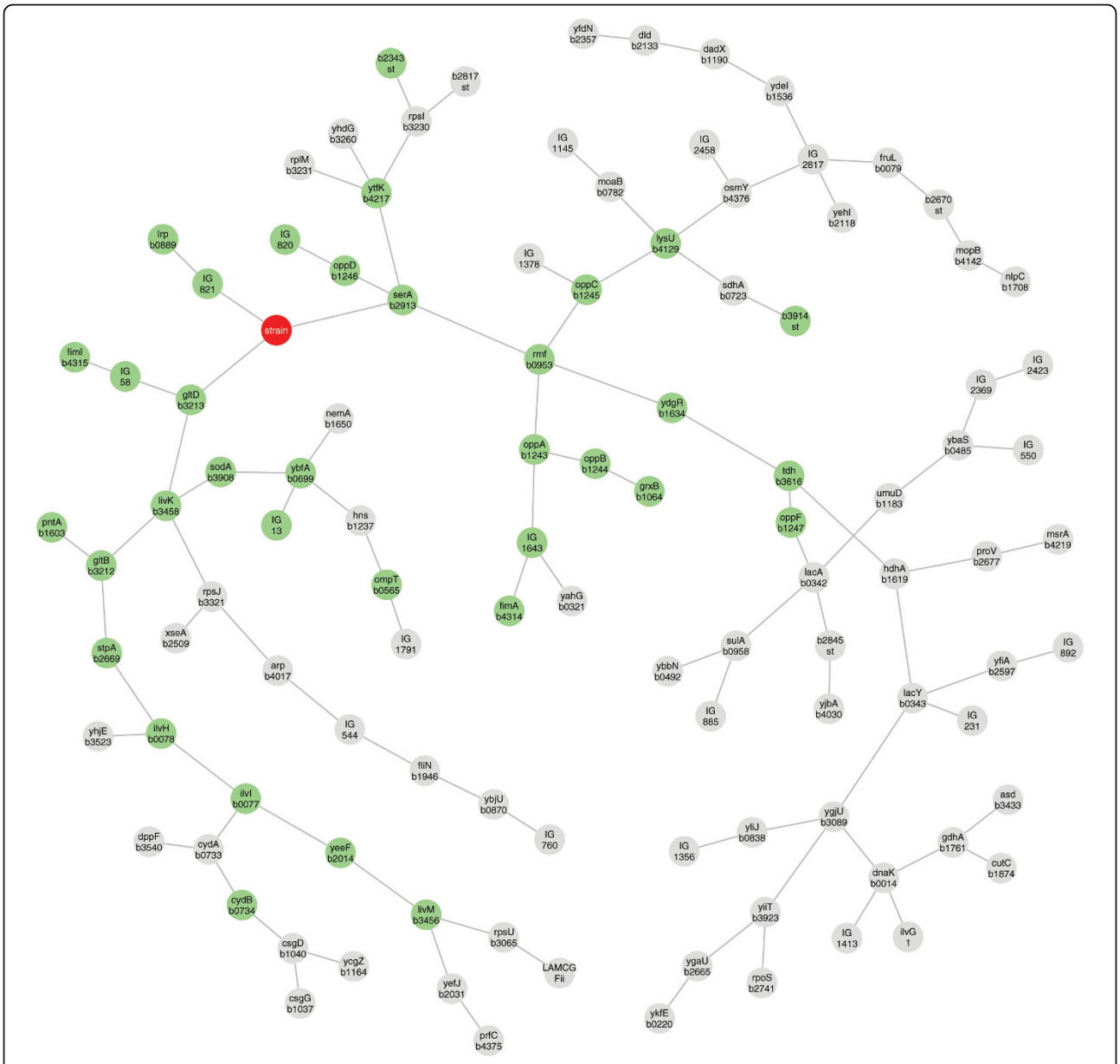

Figure 5 The radius eight neighbourhood of strain**in the minimal BIC forest for the E. coli data**. The class variable strain is shown as a red circle, and genes that are among the 40 top ranked in the limma analysis are shown as green circles.

For illustrative purposes, the first 300 polymorphic SNPs and 300 gene expression values are here used in the analysis. If non-polymorphic SNPs were included, they would appear as isolated vertices in the SD-forest, but it is more efficient to exclude them beforehand. As may be characteristic for SNP data, there are many ties in the mutual information quantities, so there may be multiple SD-forests with minimal BIC. The algorithm took about 2 seconds to find the one shown in Figure 7 below.

The main component of the SD-forest consists of a large connected block of SNPs, attached to most of the gene expression nodes via SNP number 87 at the bottom of the figure. There are also 30 or so gene expression nodes adjacent to the SNPs as singletons, and a component of nine gene expression variables connected to SNP number 54 in the centre of the graph. SNP number 130 is possibly a gene expression hotspot and there are several potential hub nodes among the gene expression values.

The SD-forest does not allow study of the joint effect of SNPs on gene expression values since, as we have seen, in trees and forests variables may have most one determinant. The minimal BIC forest obtained can be

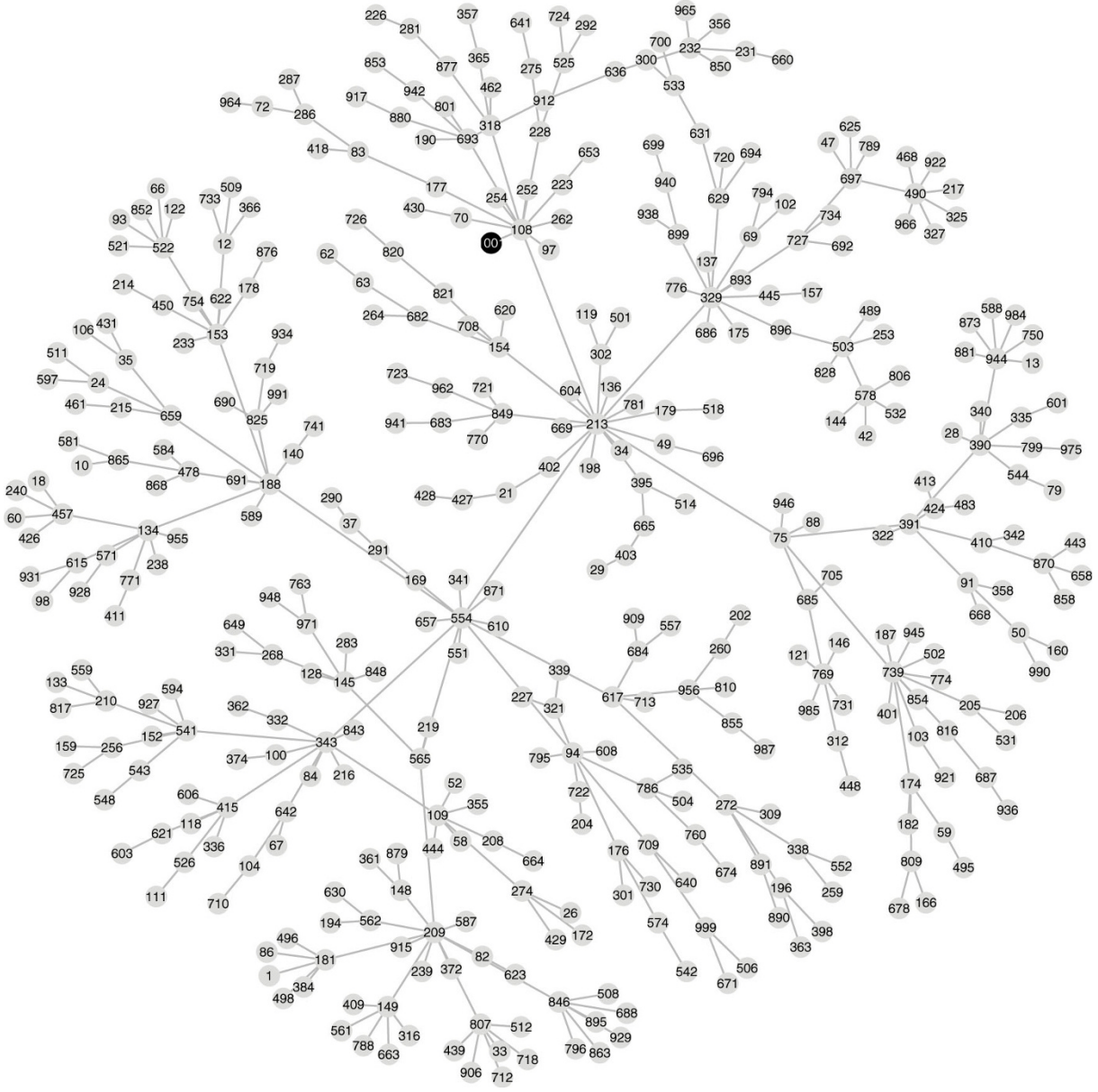

Figure 6 The radius seven neighbourhood of the class variable in the minimal BIC forest for the breast cancer data. The class variable is shown as a black circle.

regarded as a special case of a chain graph model with two blocks, with the SNP data in the first block and transcript abundance data in the second block, as mentioned above. This framework would be well-suited for further analysis of the data, allowing study of the joint action of SNPs on gene expression values.

Discussion

Deriving networks from high-dimensional data is a key challenge in many disciplines, and many different approaches have been proposed: for example, using approximation techniques [34] or low-order conditional independence tests [35,36]. One broad approach is to consider restricted classes of graphs, for example triangulated graphs [37], interval graphs [38] and others mentioned above, for which faster algorithms can be applied. The Chow-Liu algorithm falls into this class. Its utility is due to its remarkable computational efficiency, which reflects the simplicity of the graphs used. At the other end of the spectrum, it has been shown that selecting general Bayesian networks by maximizing a score function is NP-hard [39].

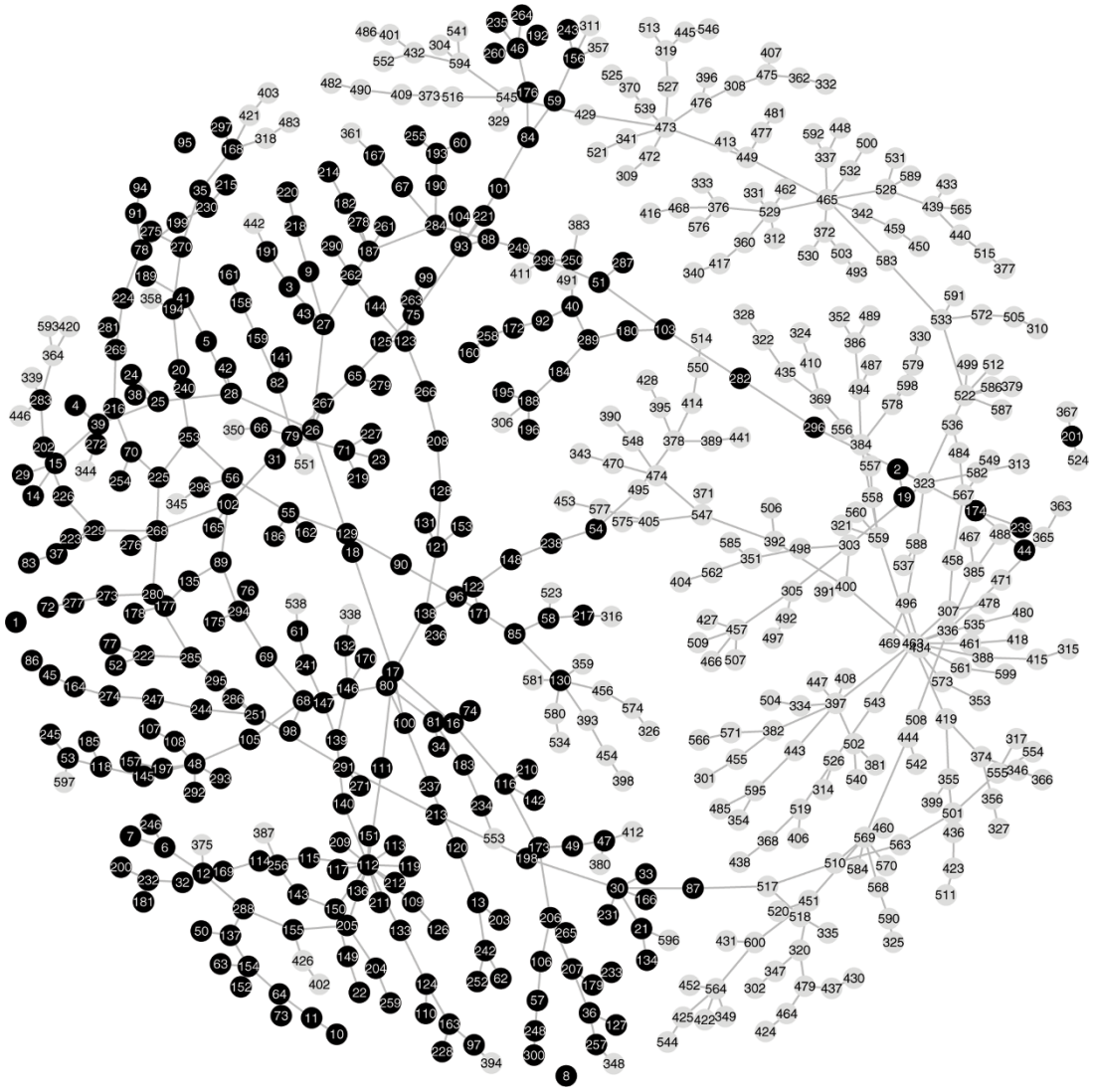

Figure 7 The minimal BIC forest for the HapMap data. There are five connected components: the main component has 594 nodes, there is one with three nodes and there are three isolated nodes.

In this paper we have described some simple extensions to Chow and Liu's method that enable forests with minimal AIC or BIC to be found, and allow datasets with both discrete and Gaussian variables to be handled. In the previous section we demonstrated that useful insights into various high-dimensional datasets may be obtained by this method.

Trees and forests are too simple to be realistic models of complex biological systems. Nevertheless we believe that they can give a preliminary understanding of the overall dependence structure, and can be put to a number of practical uses.

Firstly, we can use the selected model as a start model in a search algorithm based on richer, but more computationally demanding, model classes. Since trees are triangulated, the class of (strongly) decomposable models is a natural choice for high-dimensional data. As described above, trees and forests represent Markov equivalence classes of DAGs, so the minimal AIC/BIC forest can also be used as start model in Bayesian network search procedures.

Secondly, we can regard properties of the selected model as proxies for corresponding properties of the true, underlying network. Properties that can be used in this way include connectivity, path length and degree. Provided we can assume that the data are generated by a joint undirected model, we can model the connected components of the selected forest separately. This may allow substantial dimension reduction. It is natural to use the selected forest to identify neighborhoods of

interesting variables for more detailed analysis: in effect, this uses path length in the forest as a proxy for minimum path length in the unknown true network. Similarly, we can identify interesting features such as hub nodes – nodes of high degree – that may play a special role in the true network.

Recently there has been interest in *network motifs* – patterns of interconnections between small numbers of nodes that occur significantly more often than could be expected by chance [40]. For a review of motif discovery algorithms, see [41]. Many of these motifs, such as the feed-forward or bi-parallel motifs, will not appear in trees due to the single-parent restriction discussed above. For this reason trees and forests appear to be too restrictive for motif discovery.

As pointed out by a referee, there are some similarities between the Chow-Liu algorithm and the ARACNE algorithm [42]. Like the Chow-Liu algorithm, this algorithm initially computes the mutual information quantities $I_{u,\,v}$ for all node pairs (although ARACNE uses the Gaussian kernel method of [43]). It forms an initial graph $\mathcal{G}_0$ by including all edges for which the $I_{u,\,v}$ exceeds a given threshold. The data-processing inequality states that if X_u and X_w are conditionally independent given X_v, then $I_{u,\,w} < \min(I_{u,\,v}, I_{v,\,w})$. This is used to prune all complete triplets in $\mathcal{G}_0$, that is, all triplets X_u, X_v, X_w with all three edges present in $\mathcal{G}_0$, by removing the edge with the least mutual information. Since the condition given in the data-processing inequality is sufficient but not necessary, that the inequality holds does not imply that the condition is true, and the authors acknowledge that the process may incorrectly remove edges.

Nevertheless the heuristic is reported to perform well when the true graph is a tree or is tree-like [42].

Although mixed graphical models have been studied for some time [21-23], their adoption by the machine learning community seems to have been limited. As illustrated above, some natural application areas include comparative microarray studies, to model the effect of an intervention or class variable on gene expression, and genetics of gene expression studies, involving both discrete DNA markers (SNPs) and continuous responses (gene expression values). In both cases the discrete variables are clearly prior to the continuous variables. The conditional Gaussian assumption is a distributional assumption that is not necessarily fulfilled for all continuous variables; but log-transformed gene expression values have been found to be approximately Gaussian, and this assumption provides the basis for conventional analyses of differential expression.

An attractive aspect of the algorithm is that it allows different measures of mutual information to be used – for example, measures based on specific genetic models. However, we consider it a key advantage of the models described here that they are embedded in a broader class of models for more general dependence structures, which provides an inferential framework for systematic model diagnostics and development.

Conclusion

The approach is generally useful as a preliminary step towards understanding the overall dependence structure of high-dimensional discrete and/or continuous data. Trees and forests are unrealistically simple models for biological systems, but can nevertheless provide useful insights. In microarray studies the method supplements lists of differentially regulated genes, by suggesting a possible network of interrelationsships between these. Other uses include the following: identification of distinct connected components, which can be analysed separately (dimension reduction); identification of neighbourhoods for more detailed analyses; as initial models for search algorithms with a larger search space, for example decomposable models or Bayesian networks; and identification of interesting features, such as hub nodes.

Methods

Modifying Kruskal's algorithm to find the maximum weight spanning SD-forest

We take as given an undirected graph $\mathcal{W} = (V, E_{\mathcal{W}})$ with positive edge weights, whose vertices are marked as either discrete and or continuous. We assume that the weights are distinct so that there is a unique spanning SD-forest with maximum weight. We consider the following modification of Kruskal's algorithm.

Starting with the null graph, repeatedly add the edge with the largest weight that does not form a cycle or a forbidden path. We claim that this finds the maximum weight SD-forest.

To prove this, let $T = (V, E_T)$ be the maximum weight spanning SD-forest, and let the edges chosen by the algorithm be $a_1 \ldots a_k$. Let $A_i = (V, E_i)$ be the SD-forest consisting of edges $a_1 \ldots a_i$, so that $E_i = \cup_{1 \le j \le i} \{a_j\}$. Suppose that $T \neq A_k$. Then either or both of (i) $E_k \nsubseteq E_T$ and (ii) $E_T \nsubseteq E_k$ must hold.

Suppose that (i) holds, and let a_i be the first edge of A_k which is not in E_T. The addition of a_i to T must result in a cycle or a forbidden path. Let $a_i = (u, v)$ and let the connected components (trees) of T containing u and v be S_u and S_v.

Suppose first that $S_u \neq S_v$. Addition of an edge between distinct components cannot create a cycle, but may create a forbidden path. Addition of an edge between discrete vertices cannot create a forbidden path, so one or both of u and v must be continuous. Suppose that u is discrete and v is continuous. Then (V,

$E_T \cup a_i$) contains a unique forbidden path of the form u, v, $v_1 \ldots v_m$, w for some $m \geq 0$ where $v_1 \ldots v_m$ are continuous and w is discrete. It is unique because the existence of two such paths would imply the existence in S_v of a cycle (if the paths have the same w) or a forbidden path (if they have different w's). Since A_i is an SD-forest at least one edge in this path, say e, must be absent from A_i. Then (V, $E_{i-1} \cup e$) is a SD-forest since it is contained in T. So the weight of e must be less than that of a_i. Consider (V, $E_T \backslash e$). The removal of e from S_v results in two subtrees, the one with v containing continuous vertices only. Hence (V, $E_T \cup a_i \backslash e$) is an SD-forest. But the weight of (V, $E_T \cup a_i \backslash e$) is greater than that of T, contradicting the definition of T. The proof when both u and v are continuous is similar.

Suppose now that $S_u = S_v$. Then (V, $E_T \cup a_i$) contains exactly one cycle, and may also contain a forbidden path. The cycle must contain a_i and also some edge e which is not in A_k. Then (V, $E_T \cup a_i \backslash e$) is a forest. Suppose that (V, $E_T \cup a_i$) contains no forbidden path. Then (V, $E_T \cup a_i \backslash e$) is an SD-forest. Since (V, $E_{i-1} \cup e$) is contained in T, it is an SD-forest, so the weight of e is less than that of a_i. But then the weight of (V, $E_T \cup a_i \backslash e$) is greater than that of T, contradicting the definition of T.

Suppose now that (V, $E_T \cup a_i$) contains a forbidden path, and let $a_i = (u, v)$. Suppose that u is discrete and v continuous. Then (V, $E_T \cup a_i$) contains a unique forbidden path of the form u, v, $v_1 \ldots v_m$, w for some $m \geq 0$ where $v_1 \ldots v_m$ are continuous and w is discrete. Let w, $w_1 \ldots w_n$, u for some $n \geq 0$ be the unique path in S_u between w and u. Since S_u is an SD-tree $w_1 \ldots w_m$ are discrete. Then the unique cycle in (V, $E_T \cup a_i$) takes the form u, v, $v_1 \ldots v_m$, w, $w_1 \ldots w_n$, u. Since A_i is an SD-forest at least one edge in the path u, v, $v_1 \ldots v_m$, w, say e, must be absent from A_i. Removal of e from (V, $E_T \cup a_i$) breaks the cycle and the forbidden path, so (V, $E_T \cup a_i \backslash e$) is an SD-forest. As before the weight of e is less than that of a_i, so the weight of (V, $E_T \cup a_i \backslash e$) is greater than that of T, contradicting the definition of T. The proof when both u and v are continuous is similar.

Hence $E_k \subseteq E_T$.

Suppose now that (ii) holds. But any edge $e \in E_T \backslash E_k$ would give rise to a cycle or a forbidden path if added to E_k. Since $E_k \subseteq E_T$ this implies that T contains a cycle or forbidden path, contradicting its definition. It follows that $E_T \subseteq E_k$ and hence $T = A_k$ as required.

Two theoretical properties of minimal AIC or BIC forests

In this section we prove the two theoretical properties of the selected models discussed above.

Firstly, suppose that we apply the algorithm to find the minimal AIC or BIC forest, say $\mathcal{G}$. Then the connected components of $\mathcal{G}$ are identical to the connected components of the minimal AIC/BIC strongly decomposable model. To see this, consider the connected components (that is, trees) of $\mathcal{G}$. Then any inter-component edge either corresponds to a negative penalized mutual information or would generate a forbidden path (since adding such an edge cannot form a cycle).

Suppose that we construct a global model $\mathcal{G}^*$ by using the strongly decomposable model with minimal AIC/BIC for each connected component of $\mathcal{G}$. It follows from decomposition properties of undirected graphical models [22] that adding an inter-component edge to $\mathcal{G}^*$ would result in the same change in AIC/BIC as when added to $\mathcal{G}$. Furthermore, if adding such an edge to $\mathcal{G}$ would generate a forbidden path it would do the same when added to $\mathcal{G}^*$. So $\mathcal{G}^*$ is, at least locally, a minimal AIC/BIC strongly decomposable model.

Secondly, in some cases the global optimality of the selected model holds under marginalization. That is to say, if $\mathcal{G}$ is the maximum likelihood tree (or minimal AIC or BIC forest) for a variable set V, then for some variable subsets $A \subseteq V$, the induced marginal subgraph of $\mathcal{G}$ on A, written $\mathcal{G}_A$, is the maximum likelihood tree (respectively, minimal AIC or BIC forest) for the variable set A. It is useful to characterize precisely the sets A for which this property holds in general.

Suppose initially that $\mathcal{G}$ is connected, that is, a tree. We claim that the property holds precisely for those sets A for which $\mathcal{G}_A$ is connected. Write $\mathcal{G}_A = (A, E_A)$ and consider application of the algorithm to A, that is, to the subset of the (possibly penalized) mutual information quantities that pertain to A. Suppose that this generates the graph $\mathcal{H} = (A, E^*)$. We need to show that when the algorithm is applied to V, the inclusion of an edge between vertices in A cannot create a cycle or forbidden path involving edges not in A. If this occurs during the course of the algorithm, it will also occur when added to $\mathcal{G}$, so it is sufficient to consider $\mathcal{G}$. If $\mathcal{G}_A$ is connected then precisely one vertex in each connected component of $\mathcal{G}_{V \backslash A}$ is adjacent to precisely one vertex of $\mathcal{G}_A$. So clearly the addition of an edge in A cannot create a cycle with edges not in A. Suppose it creates a forbidden path involving vertices not in A. This must link two discrete variables, say u and v, in distinct connected components of $\mathcal{G}_{V \backslash A}$. Since $\mathcal{G}$ is an SD-tree, all vertices in the unique path between the two vertices in $\mathcal{G}$ must be discrete. This path must include the two vertices, say w and x, that are adjacent to a vertex in the connected components. If inclusion of an edge in A creates a forbidden path between u and v, then this must pass through w and x. But then the forbidden path lies in A, contrary to assumption. It follows that $\mathcal{H} = \mathcal{G}_A$. Conversely, if $\mathcal{G}_A$ is not connected but $\mathcal{G}$ is, the inclusion of inter-component edges may give rise to cycles when the algorithm is applied to V but not when it is applied to A. Hence in general $\mathcal{H}$ and $\mathcal{G}_A$ will differ.

When the minimal AIC or BIC variants of the algorithm are used, $\mathcal{G}$ may be a forest. Let the connected components of $\mathcal{G}$ be C_1,... C_k, say. Using a similar logic we obtain that $\mathcal{G}_A$ is the minimal AIC (or BIC) forest for the variable set A provided that $\mathcal{G}_{A\cap C_i}$ is connected, for each i.

Availability

The analyses were performed using the R library gRapHD which we have made available to the R community via the CRAN repository (de Abreu GCG, Labouriau R, Edwards D: High-dimensional Graphical Model Search with gRapHD R package, submitted to J. Stat. Software).

Acknowledgements

DE was supported by the Danish National Advanced Technology Foundation through the ILSORM project. GCGA was financed by SABRETRAIN Project, funded by the Marie Curie Host Fellowships for Early Stage Research Training, as part of the 6th Framework Programme of the European Commission. RL was partially supported by the project "Metabolic programming in Foetal Life", Danish Research Agency, Ministry of Science Technology and Innovation.

Authors' contributions

DE conceived the algorithm, performed the analyses and drafted the paper. GCGA carried out the programming effort. All authors contributed discussions to the theoretical development, and read and approved the final manuscript.

Received: 4 June 2009
Accepted: 11 January 2010 Published: 11 January 2010

References

1. Friedman N: **Inferring cellular networks using probabilistic graphical models.** *Science* 2004, **303(5659)**:799-805.
2. Larrañaga P, Inza I, Flores J: **A Guide to the Literature on Inferring Genetic Networks by Probabilistic Graphical Models.** *Data Analysis and Visualization in Genomics and Proteomics* Wiley, New YorkAzuaje F, Dopazo J 2005, 215-238.
3. Andrade-Cetto L, Manolakos E: **A Graphical Model Formulation of the DNA Base-Calling Problem.** *Proc. IEEE Workshop on Machine Learning for Signal Processing* 2005, 369-374.
4. Chow C, Liu C: **Approximating discrete probability distributions with dependence trees.** *Information Theory, IEEE Transactions* 1968, **14(3)**:462-467.
5. Kruskal J: **On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem.** *Proc Am Math Soc* 1956, **7**:48-50.
6. Meila M: **An accelerated Chow and Liu algorithm: fitting tree distributions to high dimensional sparse data.** *Proceedings of the 16th International Conference on Machine Learning* 1999.
7. Pelleg D, Moore A: **Dependency Trees in sub-linear time and bounded memory.** *The International Journal on Very Large Databases* 2006, **15**:250-262.
8. Bach F, Jordan M: **Thin Junction Trees.** *Advances in Neural Information Processing Systems* Cambridge, MA: MIT PressDietterich TG, GZ Becker S 2002, **14**:569-576.
9. Ouerd M, Oommen B, Matwin S: **A formal approach to using data distributions for building causal polytree structures.** *Information Sciences* 2004, 111-132.
10. Srebro N: **Maximum likelihood bounded tree-width Markov networks.** *Artificial Intelligence* 2003, **143**:123-138.
11. Meila M, Jordan MI: **Learning with mixtures of trees.** *J Mach Learn Res* 2001, **1**:1-48.
12. Sudderth E, Sudderth E, Wainwright M, Willsky A: **Embedded trees: estimation of Gaussian Processes on graphs with cycles.** *IEEE Transactions on Signal Processing* 2004, **52(11)**:3136-3150.
13. Kirshner S, Smyth P, Robertson AW: **Conditional Chow-Liu tree structures for modeling discrete-valued vector time series.** *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence* Arlington, Virginia, United States: AUAI Press 2004, 317-324.
14. Chou C, Wagner T: **Consistency of an estimate of tree-dependent probability distribution.** *IEEE Transactions on Information Theory* 1973, **IT-19**:369-371.
15. Rissanen J: **Stochastic Complexity.** *J Royal Stat Soc B* 1987, **49**:223-239.
16. Akaike H: **A new look at the statistical identification problem.** *IEEE Trans Auto Control* 1974, **19**:716-723.
17. Schwarz G: **Estimating the Dimension of a Model.** *Annals of Statistics* 1978, **6**:461-464.
18. Burnham KP, Anderson DR: **Multimodel Inference: Understanding AIC and BIC in Model Selection.** *Sociological Methods Research* 2004, **33**:261-304.
19. Liang P, Srebro N: **Methods and experiments with bounded tree-width Markov networks.** *Tech rep* MIT 2004.
20. Panayidou K: **Estimation of Tree Structure for Variable Selection.** *PhD thesis* University of Oxford, to appear..
21. Edwards D: *Introduction to Graphical Modelling* New York: Springer-Verlag, second 2000.
22. Lauritzen SL: *Graphical Models* Oxford: Clarendon Press 1996.
23. Lauritzen S, Wermuth N: **Graphical models for associations between variables, some of which are qualitative and some quantitative.** *Ann Statist* 1989, **17**:31-57.
24. Frydenberg M, Lauritzen S: **Decomposition of maximum likelihood in mixed graphical interaction models.** *Biometrika* 1989, **76**:539-555.
25. Verma T, Pearl J: **Equivalence and synthesis of causal models.** *UAI '90: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence* New York, NY, USA: Elsevier Science Inc 1991, 255-270.
26. Andersson SA, Madigan D, Perlman MD: **On the Markov Equivalence of Chain Graphs, Undirected Graphs, and Acyclic Digraphs.** *Scandinavian Journal of Statistics* 1997, **24**:81-102.
27. Spirtes P, Glymour C, Scheines R: *Causation, Prediction and Search. New York* 1993, [Reprinted by MIT Press]..
28. Heckerman D, Geiger D, Chickering DM: **Learning Bayesian Networks: The combination of knowledge and statistical data.** *Machine Learning* 1995, **20**:197-243.
29. Hung S, Baldi P, Hatfield G: **Global Gene Expression Profiling in Escherichia coli K12.** *Journal of Biological Chemistry* 2002, **477**:40309-40323.
30. Cho BK, Barrett CL, Knight EM, Park YS, Palsson B: **Genome-scale reconstruction of the Lrp regulatory network in Escherichia coli.** *Proc Natl Acad Sci USA* 2008, **105(49)**:19462-19467.
31. Smyth GK: **Linear models and empirical bayes methods for assessing differential expression in microarray experiments.** *Stat Appl Genet Mol Biol* 2004, **3**:Article3.
32. Ligi P, Blumenthal R, Matthews R: **Activation from a Distance: Roles of Lrp and Integration Host Factor in Transcriptional Activation of gltBDF.** *Journal of Bacteriology* 2001, **183**:3910-3918.
33. Miller LD, Smeds J, George J, Vega VB, Vergara L, Ploner A, Pawitan Y, Hall P, Klaar S, Liu ET, Bergh J: **An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival.** *Proc Natl Acad Sci USA* 2005, **102(38)**:13550-13555.
34. Friedman J, Hastie T, Tibshirani R: **Sparse inverse covariance estimation with the graphical lasso.** *Biostatistics* 2008, **9(3)**:432-441.
35. Kalisch M, Buhlmann P: **Estimating High-dimensional Directed Acyclic Graphs with the PC-Algorithm.** *Journal of Machine Learning Research* 2007, **8**:613-636.
36. Castelo R, Roverato A: **Reverse engineering molecular regulatory networks from microarray data with qp-graphs.** *J Comput Biol* 2009, **16(2)**:213-227.
37. Wermuth N: **Model Search among Multiplicative Models.** *Biometrics* 1976, **32(2)**:253-263.
38. Thomas A, Camp NJ: **Graphical modeling of the joint distribution of alleles at associated loci.** *Am J Hum Genet* 2004, **74(6)**:1088-1101.
39. Chickering DM: **Learning Bayesian networks is NP-complete.** *Learning from Data: Artificial Intelligence and Statistics V. New York* Fisher D, Lenz HJ 1996, 121-130.

40. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chkovskii D, Alon U: **Network Motifs: Simple Building Blocks of Complex Networks.** *Science* 2002, **298**:824-827.
41. Ciriello G, Guerra C: **A review on models and algorithms for motif discovery in protein-protein interaction networks.** *Briefings in Functional Genomics and Proteomics* 2008, **7(2)**:147-156.
42. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A: **ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.** *BMC Bioinformatics* 2006, **7(Suppl 1)**:S7.
43. Steuer R, Kurths J, Daub CO, Weise J, Selbig J: **The mutual information: detecting and evaluating dependencies between variables.** *Bioinformatics* 2002, **18(Suppl 2)**:S231-S240.

doi:10.1186/1471-2105-11-18
Cite this article as: Edwards *et al.*: Selecting high-dimensional mixed graphical models using minimal AIC or BIC forests. *BMC Bioinformatics* 2010 **11**:18.