Bayesian Networks for High-dimensional Data with Complex Mean Structure

Jessica E. Kasza

Thesis submitted for the degree of

Doctor of Philosophy

in

Statistics

at

The University of Adelaide

(Discipline of Statistics, School of Mathematical Sciences, Faculty of Engineering, Mathematical and Computer Sciences)

February 25, 2010

Contents

List of Figures

List of Tables

Abstract

In a microarray experiment, it is expected that there will be correlations between the expression levels of different genes under study. These correlation structures are of great interest from both biological and statistical points of view. From a biological perspective, the identification of correlation structures can lead to an understanding of genetic pathways involving several genes, while the statistical interest, and the emphasis of this thesis, lies in the development of statistical methods to identify such structures. However, the data arising from microarray studies is typically very high-dimensional, with an order of magnitude more genes being considered than there are samples of each gene. This leads to difficulties in the estimation of the dependence structure of all genes under study. Graphical models and Bayesian networks are often used in these situations, providing flexible frameworks in which dependence structures for high-dimensional data sets can be considered.

The current methods for the estimation of dependence structures for high-dimensional data sets typically assume the presence of independent and identically distributed samples of gene expression values. However, often the data available will have a complex mean structure and additional components of variance. Given such data, the application of methods that assume independent and identically distributed samples may result in incorrect biological conclusions being drawn. In this thesis, methods for the estimation of Bayesian networks for gene expression data sets that contain additional complexities are developed and implemented. The focus is on the development of score metrics that take account of these complexities for use in conjunction with score-based methods for the estimation of Bayesian networks, in particular the High-dimensional Bayesian Covariance Selection algorithm.

The necessary theory relating to Gaussian graphical models and Bayesian networks is reviewed, as are the methods currently available for the estimation of dependence structures for high-dimensional data sets consisting of independent and identically distributed samples. Score metrics for the estimation of Bayesian networks when data sets are not independent and identically distributed are then developed and explored, and the utility and necessity of these metrics is demonstrated. Finally, the developed metrics are applied to a data set consisting of samples of grape genes taken from several different vineyards.

Signed Statement

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution to Jessica Kasza and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue, the Australasian Digital Theses Program (ADTP) and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

SIGNED: .... DATE: ....

Acknowledgements

First and foremost, thanks are due to my supervisors, Gary Glonek and Patty Solomon. None of this would have been possible were it not for their wisdom, guidance and support. Gary, your patience is seemingly infinite, and your encouragement allowed me to do more than I thought possible. Patty, your faith in me gave me faith in myself. Thank you both for all of the effort you have put into supervising me over the years.

I would also like to acknowledge the help of Dr. Christopher Davies of the C.S.I.R.O., both for the grape gene data analysed in Chapter 8, and for his help in understanding the results of the analyses. Thank you, Chris!

The biggest thanks of all got to my family, for their support and love. Josh, your gifts of delicious chocolate were appreciated more than you know. Mum, Dad: thank you for everything.

This achievement would be empty without good friends to share it with: thank you all for being who you are. Thanks are especially due to Rhys Bowden, for being who he is.

Chapter 1

Introduction

The inner workings of a cell are very complex, with many interacting components. Determining how the genes within a cell interact with one another is an important, and difficult, field of research. Systems of these interactions are known as genetic regulatory networks. The work comprising this thesis is motivated by the estimation of such networks.

Genes are composed of deoxyribonucleic acid (DNA), a structure consisting of two strands made up of four types of nucleotide molecules. Much of the information stored in a DNA molecule can be thought of as containing instructions for the construction of proteins. Proteins are the molecules that make up a cell, and allow the cell to perform certain functions. For example, in plants there exist proteins known as heat-shock proteins that protect the cells of an organism from the consequences of heat stress. Proteins are synthesised, not directly from DNA, but from molecules known as messenger ribonucleic acid (mRNA), which are derived from DNA via a process known as transcription. Proteins are then synthesised from these mRNA molecules through a process known as translation.

If the DNA of a particular gene has been transcribed to mRNA, it is said that the gene is expressed. Gene expression is the level of the transcription of the DNA of a gene to mRNA, and provides an indication of which proteins are being formed within a cell.

The expression levels of genes within a sample of cells may be measured using microarray experiments. These experiments allow the measurement of the expression levels of thousands of genes in a sample of cells simultaneously. The details of such experiments are briefly summarised below. For a more detailed description of microarray technology, the interested reader is directed to [61]. In this thesis gene expression data sets obtained from single channel short oligonucleotide microarrays, as used by Affymetrix, are considered.

A microarray consists of a slide on to which short gene sequences, known as probes, are attached. A solution containing the mRNA present within a sample of cells is then washed over the slide. If this sample contains mRNA that is complementary to a probe on the slide, that mRNA will bind to the appropriate probe, in a process known as hybridisation. A fluorescent dye is added, and after hybridisation, the slides are washed to remove any excess genetic material.

The fluorescence of the probes is then measured by scanning the slide with a laser. The more highly expressed a gene is within a sample, the greater the fluorescence of the corresponding probe on the microarray slide. The raw measurements take the form of a digital image, which requires pre-processing and normalisation before meaningful analyses can be conducted. Pre-processing and normalisation is performed to remove systematic components of non-biological variation from the data. It will be assumed that the normalised gene expression values reflect the actual levels of expression in the genetic material.

Note that while microarray experiments can measure the expression levels of thousands of genes simultaneously, the replication of these measurements is typically very low. Due to the expense involved in the production of microarray slides, an experiment will typically consist of fewer slides than there are genes of interest. Hence, the data sets obtained from microarray experiments are typically very high-dimensional, with an order of magnitude more genes being considered than there are samples of gene expression levels.

Within a genetic regulatory network, the expression levels of genes are controlled by the expression levels of other genes. Typically, the expression level of a particular gene will be influenced by the expression levels of a relatively small number of genes. There may exist feedback loops within a genetic regulatory network, whereby the expression level of a particular gene is regulated by itself.

The extent to which genetic regulatory networks may be inferred from observational gene expression data is unknown. To explore this question carefully, high-dimensional multivariate models, such as Bayesian networks and graphical models, need to be considered. Estimation of such structures allows insight into how the expression levels of large groups of genes are related to one another, which may, in turn, shed light on genetic regulatory networks involving the genes. It is hoped that, in due course, the results of such analyses will aid the work of biologists; indicating which aspects of genetic regulatory networks warrant further biological investigation, and guiding experimentation.

The intended purpose of the estimation of Bayesian networks for gene expression data is to discover novel regulatory networks. However, often the correlations between genes apparent in observational gene expression data will be due to external factors, instead of due to some regulatory association. For example, the expression levels of heat shock genes are known to be affected by changes in temperature. Hence, if the effect of temperature is unaccounted for in the analysis of the expression levels of heat shock genes, because of their common relationships with temperature, many pairs of genes will exhibit strong correlations. Unless the gross effect of temperature is removed, one cannot hope to detect more subtle associations between genes.

There are many methods available for the estimation of graphical models and Bayesian networks given gene expression data, or other high-dimensional data sets. The vast majority of these methods do not allow the inclusion of external factors, assuming data sets that consist of independent and identically distributed samples. The focus of this thesis is on the development of methods for the estimation of Bayesian networks from high-dimensional data sets that allow the inclusion of the effects of external factors, in the form of a complex mean structure.

Chapter 2 consists of a review of the graph theory necessary for the remainder of the thesis. The concept of conditional independence is reviewed, as are graphical models and Bayesian networks. It is noted that the estimation of a Gaussian graphical model, or a Bayesian network, which is then moralised, for a set of variables is equivalent to the estimation of the covariance matrix of these variables. In Section 2.3, the representation of genetic regulatory networks by graphical models and Bayesian networks is discussed.

In Chapter 3, many currently available methods for the estimation of Bayesian networks and graphical models are reviewed. Of particular interest are the score-based methods for the estimation of Bayesian networks given high-dimensional data, which are discussed in Section 3.1.1. Constraint-based methods for the estimation of Bayesian networks are briefly discussed in Section 3.1.2, as are methods for the estimation of graphical models in Section 3.2. The method used for the estimation of Bayesian networks in this thesis is High-dimensional Bayesian Covariance Selection, [23, 25], a score-based method that is discussed in Section 3.3.

In Chapter 4, new score metrics for data sets that are not independent and identically distributed are developed. These score metrics can be used in conjunction with any score-based method for the estimation of Bayesian networks. These score metrics are further explored in Chapter 5. In Chapter 6, a novel score metric motivated by residual maximum likelihood for use with samples that are not independent and identically distributed is developed.

In Chapter 7, the consequences of incorrectly assuming independent and identically distributed samples are illustrated through the estimation of Bayesian networks for data sets with a complex mean structure. The utility of the methods developed in Chapters 4 and 6 in the estimation of Bayesian networks for data sets with a complex mean structure is then demonstrated. The use of these methods is illustrated with the High-dimensional Bayesian Covariance Selection algorithm.

Chapter 8 consists of the application of the methods developed in Chapters 4 and 6 to a real gene expression data set. This data set consists of expression levels of grape genes known to be affected by temperature, where grapes were sampled from three different vineyards, and temperatures leading up to the times when the grapes were sampled. This data set does not consist of independent and identically distributed samples. Initially, a Bayesian network is estimated assuming independent and identically distributed samples. This Bayesian network is compared to those obtained when the complex mean structure of the data set is accounted for through the use of the methods developed in Chapters 4 and 6.

Finally, the results of the thesis are summarised in Chapter 9, and future research is discussed.

Chapter 2

Graph Theory and Graphical Modelling

In this chapter, the necessary graph theory will be reviewed, graphical models and Bayesian networks will be defined, and the application of such models to the analysis of gene expression data will be explained.

2.1 Required Graph Theory

In this section, the graph theory required for the remainder of the work is stated. Readers interested in a more detailed account of graph theory are directed to [50].

A graph is a pair, $\mathcal{G} = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of elements known as vertices, and $E \subseteq \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$, is the edge set. Note that pairs in the edge set are ordered, so that $(v_i, v_j) \neq (v_j, v_i)$.

We will write i for $v_i \in V$, and (i, j) for $(v_i, v_j) \in E$.

Consider a graph $\mathcal{G} = (V, E)$, and $i, j \in V$. If (i, j) and (j, i) are both in E, the edge between i and j is said to be undirected, and in the diagram of the graph, a line is drawn between the two vertices. If $(i, j) \in E$ and $(j, i) \notin E$, there is said to be a directed edge from i to j, and in the diagram an arrow is drawn from i to j. An undirected graph is a graph with only undirected edges, while a directed graph has only directed edges. In what follows, graphs with both directed and undirected edges will not be considered.

If there is an undirected edge between two vertices i and j then i and j are said to be adjacent. The set of nodes adjacent to node i is said to be the neighbourhood of i, written N_i. The degree of a vertex is the number of nodes it is adjacent to.

If it is clear that an undirected graph is being considered, the edges of a graph will be taken to be unordered pairs in $V \times V$.

If $A \subset V$, a subgraph $\mathcal{G}_A = (A, E_A)$ may be induced by taking $E_A = E \cap (A \times A)$. In plainer terms, the edges of the induced graph $\mathcal{G}_A$ are the edges of the original graph $\mathcal{G}$ that link vertices contained in A.

A graph is said to be complete if there are edges between all the vertices of that graph. A subset of the vertex set is said to be complete if it induces a complete subgraph. A clique is a complete subset that is maximal with respect to inclusion. That is, $A \subseteq V$ is a clique if the subgraph induced by A is complete, but the subgraph induced by $A \cup \{i\}$, for any $i \in V \setminus A$, is not complete.

A path of length m from i to j is a sequence of distinct vertices $i = v_0, \ldots, v_m = j$ such that $(v_{k-1}, v_k) \in E \ \forall k = 1, \ldots, m$. If there exists a path from i to j, it is said that i leads to j, written $i \rightsquigarrow j$. If $i \rightsquigarrow j$ and $j \rightsquigarrow i$, it is said that the two vertices are connected, written $i \rightleftharpoons j$. Note that in an undirected graph, $i \rightsquigarrow j \iff j \rightsquigarrow i$. It can be seen that $\rightleftharpoons$ is an equivalence relation on the vertex set of a graph, with equivalence classes $[i]$: $j \in [i] \iff i \rightleftharpoons j$. The equivalence classes of a graph, so defined, are called the connectivity components of $\mathcal{G}$.

A path $i = v_0, \ldots, v_m = i$, $m > 1$, is called a cycle.

A directed acyclic graph, often abbreviated as DAG, is a directed graph containing no cycles.

Consider $i, j \in V$. If $(i, j) \in E$, i is said to be the parent of j, and j the child of i. The set of parents of a vertex i is denoted by P_i, the set of children by C_i.

If there is a path such that $k \rightsquigarrow i$, but no path such that $i \rightsquigarrow k$, k is said to be an ancestor of i. The set of all ancestors of i is written A_i. The descendants of a vertex i, written D_i, is the set of vertices k such that $i \rightsquigarrow k$ and $k \not\rightsquigarrow i$. The non-descendants of a vertex i, written ND_i, consist of the vertices in $V \setminus \{D_i \cup \{i\}\}$.

Consider now a set of vertices, A. The boundary of this set of vertices, BD_A, is the set of vertices in $V \setminus A$ that are adjacent to or parents of any of the vertices in A. The closure of A, denoted by CL_A, is $A \cup BD_A$.

An (i, j)-separator is a set $C \subset V$ such that all paths from i to j intersect C. In an undirected graph, the notion can be expressed in terms of connectivity components: C is an (i, j)-separator $\Leftrightarrow [i]_{V \setminus C} \neq [j]_{V \setminus C}$. The subset C is said to separate A from B, $A, B \subseteq V$, if it is an (i, j)-separator $\forall\ i \in A, j \in B$.

The skeleton of a directed graph is the undirected graph obtained by replacing all directed edges with undirected edges.

The moral graph, $\mathcal{G}^m$, of a directed graph $\mathcal{G}$ is the undirected graph with the same vertex set as $\mathcal{G}$ but with i and j adjacent in $\mathcal{G}^m$ if and only if $(i, j) \in V$ or $(j, i) \in V$ or if i and j share a child. In the anachronistically named moral graph, all parents are "married"; that is, joined by an edge. If no edges have to be added to the directed graph to form the moral graph, the directed graph is said to be perfect.

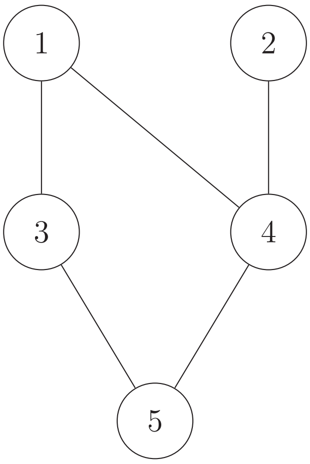


Figure 2.1: An undirected graph, discussed in Example 2.1

Example 2.1. *An example of an undirected graph.*

Consider the undirected graph $\mathcal{G}$ depicted in Figure 2.1. Here, $V = \{1, 2, 3, 4, 5\}$ and $E = \{(1, 3), (1, 4), (2, 4), (3, 5), (4, 5)\}$. This graph is represented in Figure 2.1. The boundary and closure sets of each vertex can be found: for example, $BD_1 = \{3, 4\}$, so $CL_1 = \{1, 3, 4\}$.

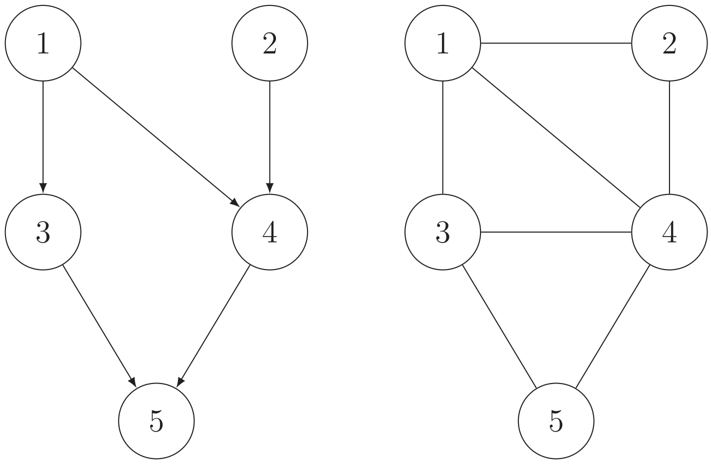


Figure 2.2: The graphs discussed in Example 2.2. On the left is an example of a directed acyclic graph, and on the right the moral graph.

Separators can also be found: $\{3, 4\}$ is a $(1, 5)$-separator. The graph can be seen to contain a cycle of length 4.

Example 2.2. *An example of a directed graph.*

The graph in Figure 2.2 has the same vertex and edge sets as in Example 2.1, however, the edge set now contains ordered pairs. The graph is acyclic and parent and child sets of each vertex can be found. For example, $P_4 = \{1, 2\}$ and $C_4 = \{5\}$. Similarly, $A_4 = \{1, 2\}, D_4 = \{5\}, ND_4 = \{1, 2, 3\}$. The corresponding moral graph is also shown in this figure, and the skeleton is given in Figure 2.1.

2.2 Graphical Models

In what follows, the emphasis is upon continuous random variables. For a review of graphical models for discrete random variables, or graphical models for both discrete and continuous random variables, the reader is directed to [13, 27, 50, 89].

2.2.1 Conditional Independence

The notion of the conditional independence of random variables is central to graphical modelling [13, 27, 50, 89]. A general discussion of conditional independence in statistics is given by [17]. It will be shown that graphs are well-suited for the representation of the conditional independence relationships of a set of random variables.

Definition 2.1. *Consider continuous random variables X, Y and Z. X is said to be conditionally independent of Y given Z, written $X \perp\!\!\!\perp Y|Z$, if and only if the joint density function satisfies*

$$f_{XYZ}(x, y, z) = g(x, z)h(y, z),$$

for some functions g and h. Note that this definition holds in the case that X, Y and Z are continuous random vectors.

The following properties of conditional independence can be shown to hold, for any measurable function k:

$(C1)$ $X \perp\!\!\!\perp Y|Z \implies Y \perp\!\!\!\perp X|Z$;

$(C2)$ $X \perp\!\!\!\perp Y|Z \implies k(X) \perp\!\!\!\perp Y|Z$;

$(C3)$ $X \perp\!\!\!\perp Y|Z \implies X \perp\!\!\!\perp Y|(Z, k(X))$;

$(C4)$ $X \perp\!\!\!\perp Y|Z$ and $X \perp\!\!\!\perp W|(Y, Z) \implies X \perp\!\!\!\perp (W, Y)|Z$.

See [17] for proofs of these properties.

2.2.2 Markov Properties

The Markov properties described in this section are the ties that bind conditional independence relationships to graph theory. It is these Markov properties that allow conditional independence relationships to be read directly off of graphs.

For a more in-depth analysis of the Markov properties of graphs, the interested reader is referred to Chapter 3 in [50], from which much of the present section is derived.

Definition 2.2. *Consider a set of random variables $\{X_1, X_2, \ldots, X_n\}$. An undirected graph $\mathcal{G} = (V, E)$ associated with this set of random variables has $V = \{1, 2, \ldots, n\}$, and E a subset of unordered pairs from $V \times V$.*

Let A, B and C be subsets of $\{1, 2, \ldots, n\}$. In what follows, if $X_A \perp\!\!\!\perp X_B | X_C$, we will write $A \perp\!\!\!\perp B|C$.

There are three Markov properties that may hold with respect to such graphs:

(P) the Pairwise Markov property holds if, for any i and j such that $(i, j) \notin E$,

$$i \perp\!\!\!\perp j | V \setminus \{i, j\};$$

(L) the Local Markov property holds if, for any $i \in V$,

$$i \perp\!\!\!\perp V \setminus CL_i | BD_i;$$

(G) the Global Markov property holds if, for S separating A and B,

$$A \perp\!\!\!\perp B|S.$$

It appears that (G) is the most powerful of these properties, and this is proven in the following theorem.

Theorem 2.1. ***(Markov Implication)*** *For any set of random variables $\{X_1, X_2, \ldots, X_n\}$ with associated undirected graph $\mathcal{G}$,*

$$(G) \implies (L) \implies (P).$$

Proof. $(G) \implies (L)$**:** Since for any $i \in V$, BD_i separates CL_i and $V \setminus CL_i$, this implication is proved.

$(L) \implies (P)$**:** Consider $i, j \in V$ with $(i, j) \notin E$. Then $j \in V \setminus CL_i$. Hence, $BD_i \cup \{(V \setminus CL_i) \setminus \{j\}\} = V \setminus \{i, j\}$, so (L) and $(C3)$ imply that $i \perp\!\!\!\perp V \setminus CL_i | V \setminus \{i, j\}$. Symmetry of conditional independence and $(C2)$ now imply that $i \perp\!\!\!\perp j | V \setminus \{i, j\}$.

□

A much stronger result holds when the random variables have a positive probability density function:

Theorem 2.2. ***(Equivalence of Markov Properties)*** *If a set of continuous random variables $\{X_1, X_2, \ldots, X_n\}$ have a positive probability density function, with associated graph $\mathcal{G}$, then*

$$(G) \iff (L) \iff (P).$$

Proof. This proof is adapted from [50]. Proof will be done in two separate parts: First, it will be shown for A, B, C, D disjoint subsets of $V = \{1, 2, \ldots, n\}$,

$$A \perp\!\!\!\perp B|(C \cup D) \text{ and } A \perp\!\!\!\perp C|(B \cup D) \implies A \perp\!\!\!\perp (B \cup C)|D. \tag{2.1}$$

Then it will be shown that Equation (2.1) implies the equivalence of the Markov properties.

Assume that $A \perp\!\!\!\perp B|(C \cup D)$ and $A \perp\!\!\!\perp C|(B \cup D)$. Then, by the definition of conditional independence,

$$\begin{aligned} f_{X_A,X_B,X_C,X_D}(x_A, x_B, x_C, x_D) &= k(x_A, x_C, x_D)l(x_B, x_C, x_D) \\ &= g(x_A, x_B, x_D)h(x_B, x_C, x_D), \end{aligned}$$

where k, l, m and n are strictly positive. Since the density function is continuous,

$$g(x_A, x_B, x_D) = \frac{k(x_A, x_C, x_D)l(x_B, x_C, x_D)}{h(x_B, x_C, x_D)}.$$

Setting $x_C = x_{C_0}$, $g(x_A, x_B, x_D) = m(x_A, x_D)n(x_B, x_D)$, where

$$\begin{aligned} m(x_A, x_D) &= k(x_A, x_{C_0}, x_D) \\ n(x_B, x_D) &= \frac{l(x_B, x_{C_0}, x_D)}{h(x_B, x_{C_0}, x_D)}. \end{aligned}$$

This implies that

$$\begin{aligned} f_{X_A,X_B,X_C,X_D}(x_A, x_B, x_C, x_D) &= m(x_A, x_D)n(x_B, x_D)h(x_B, x_C, x_D) \\ &= m(x_A, x_D)h^*(x_B, x_C, x_D), \end{aligned}$$

which, by the definition of conditional independence, implies that $A \perp\!\!\!\perp (B \cup C)|D$.

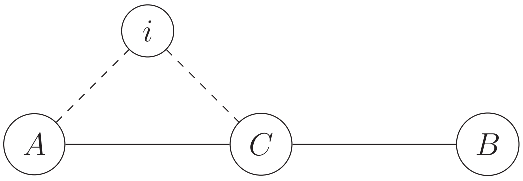
To now show that Equation (2.1) implies the equivalence of the 3 Markov properties, all that is required is a proof that $(P) \implies (G)$.

Suppose now that C separates A and B in the associated graph of $\{X_1, X_2, \ldots, X_n\}$. The proof is by backwards induction on $|C|$.

- If $|C| = |V| - 2$, then A and B each contain only one vertex, and (G) follows immediately from (P).
- Let $|C| < |V| - 2$, and suppose that $(P) \implies (G)$ for all separating sets with more elements. There are then 2 possibilities:
  - $A \cup B \cup C = V$. In this case, without loss of generality, it can be assumed that A has more than one element. If $i \in A$, then $C \cup \{i\}$ separates $A \setminus \{i\}$ and B, and $C \cup A \setminus \{i\}$ separates $\{i\}$ and B. The induction hypothesis implies

    $$A \setminus \{i\} \perp\!\!\!\perp B | C \cup \{i\} \text{ and } i \perp\!\!\!\perp B | C \cup A \setminus \{i\}.$$

    Equation (2.1) then implies that $B \perp\!\!\!\perp A|C$, as required.
  - $A \cup B \cup C \subset V$. Choose $i \in V \setminus \{A \cup B \cup C\}$. Then $C \cup \{i\}$ separates A and B, and, by the induction hypothesis, $A \perp\!\!\!\perp B|C \cup \{i\}$. Without loss of generality, this implies that $A \cup C$ separates B from $\{i\}$. The following diagram depicts the situation. Note that the dashed lines from i indicate the only places where there may be an edge from i.

    The induction hypothesis then implies $i \perp\!\!\!\perp B|A \cup C$. Applying Equation (2.1) to the previous two conditional independence relations then implies $A \cup \{i\} \perp\!\!\!\perp B|C$. (C2) then implies that $A \perp\!\!\!\perp B|C$.

□

Example 2.3. *The Markov Properties*

Consider a random vector $\boldsymbol{X} = (X_1, X_2, X_3, X_4, X_5)$. Suppose that associated with this random vector is the undirected graph in Figure 2.1. Further, suppose that (G) holds with respect to this graph. Then, by Theorem 2.1, all of the Markov properties hold. It can be seen that since $\{3, 4\}$ separates 1 and 5, $(G) \implies 1 \perp\!\!\!\perp 5|\{3, 4\}$. Further, since $BD_1 = \{3, 4\}$ and $CL_1 = \{1, 3, 4\}$, $(L) \implies 1 \perp\!\!\!\perp \{2, 5\}|\{3, 4\}$.

The definition of conditional independence implies a factorisation of the probability density function of interest. Similarly, there is a definition of factorisation of a probability density function over a graph:

Definition 2.3. *If the probability density function $f(\boldsymbol{x})$ of a random vector with associated graph $\mathcal{G}$ can be written as*

$$(F) \qquad f(\boldsymbol{x}) = \textstyle\prod_{A \text{ a complete subgraph of } \mathcal{G}} h_A(\boldsymbol{x}),$$

then f is said to factorise with respect to $\mathcal{G}$, and is said to have property (F).

This property can be shown to be equivalent to the above Markov properties under certain conditions:

Theorem 2.3. *If a set of random variables $\{X_1, X_2, \ldots, X_n\}$ has a positive and continuous probability density function, with associated graph $\mathcal{G}$, then*

$$(F) \iff (G) \iff (L) \iff (P).$$

The interested reader is directed to [50] for a proof of this theorem.

2.2.3 Independence Graphs

Definition 2.4. *Consider a random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$. The independence graph, also known as the concentration graph [13], associated with this random vector is the graph $\mathcal{G} = (V, E)$, $V = \{1, 2, \ldots, n\}$, and $(i, j) \in E$ if and only if $\{i\} \not\!\perp\!\!\!\perp \{j\}|V \setminus \{i, j\}$. That is, if $\{i\} \perp\!\!\!\perp \{j\}|V \setminus \{i, j\}$, then $(i, j) \notin E$.*

Note that since $X \perp\!\!\!\perp Y|Z \iff Y \perp\!\!\!\perp X|Z$, independence graphs are, by definition, undirected. Note also, that by definition, (P) holds for all independence graphs. Hence, if $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ has a positive and continuous probability function, all of the Markov properties hold.

Example 2.4. *An independence graph.*

Consider a random vector $\boldsymbol{X} = (X_1, X_2, X_3, X_4, X_5)$. Let $V = \{1, 2, 3, 4, 5\}$ and suppose the following conditional independence relationships hold:

- $1 \perp\!\!\!\perp 5|V \setminus \{1, 5\}$
- $2 \perp\!\!\!\perp 3|V \setminus \{2, 3\}$
- $2 \perp\!\!\!\perp 5|V \setminus \{2, 5\}$

Then the independence graph associated with this random vector is the undirected graph in Figure 2.2.

2.2.4 Gaussian Graphical Models

First, definitions of the multivariate normal distribution and graphical models are given.

Definition 2.5. *A random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ is said to have an n-dimensional multivariate normal distribution with mean $\boldsymbol{\mu}$ and variance matrix Σ, written $\boldsymbol{X} \sim N_n(\boldsymbol{\mu}, \Sigma)$, if it has density function*

$$f(\boldsymbol{x}) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right\}.$$

$\Omega = \Sigma^{-1}$ is called the precision or concentration matrix of the random vector.

Definition 2.6. *A graphical model for a random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$, with associated independence graph $\mathcal{G}$, is a probability distribution that satisfies (P) with respect to this graph. Hence, a Gaussian graphical model for a random vector with an associated independence graph is simply a multivariate normal distribution that satisfies the required conditional independence relations.*

Gaussian graphical models are often called covariance selection models, a term first coined by Dempster [21], or concentration graphs. The reason for the nomenclature is clarified through the following theorem.

Theorem 2.4. *Let $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$ have a multivariate normal distribution, $\boldsymbol{X} \sim N_n(\boldsymbol{\mu}, \Sigma)$, with concentration matrix Ω. Then there is no edge between the nodes i and j, $i, j \in \{1, 2, \dots, n\}, i \neq j$, in the independence graph associated with $\boldsymbol{X}$ if and only if ω_{ij}, the $(i,j)^{th}$ element of Ω, is equal to zero.*

Proof. By definition, there is no edge between i and j in the independence graph associated with $\boldsymbol{X}$ if and only if $i \perp\!\!\!\perp j | V \setminus \{i,j\}$. So we need to show that $i \perp\!\!\!\perp j | V \setminus \{i,j\} \iff \omega_{ij} = 0$.

Without loss of generality, take $\boldsymbol{\mu} = \boldsymbol{0}$, and consider the joint distribution:

$$\begin{aligned}
f(\boldsymbol{x}) &= (2\pi)^{-\frac{n}{2}} |\Omega|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\boldsymbol{x}^T \Omega \boldsymbol{x}\right\} \\
&\propto \prod_{k=1}^{n} \exp\left(-\omega_{kk} x_k^2\right) \prod_{k<l} \exp\left(-\omega_{kl} x_k x_l\right) \\
&\propto e^{-\omega_{ij} x_i x_j} \exp\left\{-\omega_{ii} x_i^2 - \sum_{k \in V \setminus \{i,j\}} \omega_{ki} x_k x_i\right\} \exp\left\{-\omega_{jj} x_j^2 - \sum_{k \in V \setminus \{i,j\}} \omega_{kj} x_k x_j\right\} \\
&\quad \times \prod_{k \in V \setminus \{i,j\}} e^{-\omega_{kk} x_k^2} \prod_{k \in V \setminus \{i,j\}, k<l} e^{-\omega_{kl} x_k x_l} \\
&\propto e^{-\omega_{ij} x_i x_j} g(x_i, x_{V \setminus \{i,j\}}) h(x_j, x_{V \setminus \{i,j\}}),
\end{aligned}$$

hence,

$$f(\boldsymbol{x}) = g(x_i, x_{V \setminus \{i,j\}}) h(x_j, x_{V \setminus \{i,j\}}) \iff \omega_{ij} = 0.$$

□

Theorem 2.4 implies that Gaussian graphical models are very easy to work with, as an edge between two vertices can be removed simply by setting the corresponding entry in the concentration matrix to 0 and matching the covariances in all other positions as well as the variances to the observed values.

Example 2.5. *Theorem 2.4 and Gaussian graphical models*

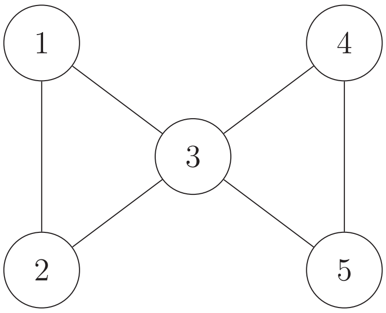
Consider a Gaussian graphical model for the vector $\boldsymbol{X} = (X_1, X_2, X_3, X_4, X_5)$ with associated independence graph given in Figure 2.1. By Theorem 2.4 the concentration matrix of this Gaussian graphical model has the following form:

$$\begin{pmatrix} \omega_{11} & 0 & \omega_{13} & \omega_{14} & 0 \\ 0 & \omega_{22} & 0 & \omega_{24} & 0 \\ \omega_{31} & 0 & \omega_{33} & 0 & \omega_{35} \\ \omega_{41} & \omega_{42} & 0 & \omega_{44} & \omega_{45} \\ 0 & 0 & \omega_{53} & \omega_{54} & \omega_{55} \end{pmatrix}.$$

Similarly, a normal random vector $\boldsymbol{Y} = (Y_1, Y_2, Y_3, Y_4, Y_5)$ with concentration matrix

$$\begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} & 0 & 0 \\ \omega_{21} & \omega_{22} & \omega_{23} & 0 & 0 \\ \omega_{31} & \omega_{32} & \omega_{33} & \omega_{34} & \omega_{35} \\ 0 & 0 & \omega_{43} & \omega_{44} & \omega_{45} \\ 0 & 0 & \omega_{53} & \omega_{54} & \omega_{55} \end{pmatrix}.$$

has an associated independence graph given by:



2.2.5 Directed Markov Properties

Consider a vector of random variables $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ with joint probability distribution $f(\boldsymbol{x})$, and an associated directed acyclic graph $\mathcal{G} = (V = \{1, 2, \ldots, n\}, E)$. If

$$f(\boldsymbol{x}) = \prod_{i=1}^{n} f_{i|P_i}(x_i|\boldsymbol{x}_{P_i}),$$

where P_i is the set of the parents of i in $\mathcal{G}$, then it is said that a recursive factorisation according to $\mathcal{G}$ is admitted, and it is said that (DF) holds. Note that for variables X_j with $P_j = \emptyset$, $f_{j|P_j}(x_j|\boldsymbol{x}_{P_j}) = f_j(x_j)$.

Theorem 2.5. *If $f(\boldsymbol{x})$ factorises recursively according to the directed acyclic graph $\mathcal{G}$, then $f(\boldsymbol{x})$ will factorise according to $\mathcal{G}^m$.*

Proof. Moralisation of a graph $\mathcal{G}$ results in the completeness of the sets $\{i\} \cup P_i$ in $\mathcal{G}^m$. Therefore, one can take $\psi_{\{i\}\cup P_i} = f_{i|P_i}$, proving the theorem. □

As there were three Markov properties that could hold with respect to undirected graphs, so there are three Markov properties that may hold with respect to directed acyclic graphs:

(DP) the Directed Pairwise Markov property holds if for any $i, j \in V$ that are non-adjacent, with $j \in ND_i$,

$$i \perp\!\!\!\perp j|ND_i \setminus \{j\};$$

(DL) the Directed Local Markov property holds if for any $i \in V$,

$$i \perp\!\!\!\perp ND_i|P_i;$$

(DG) the Directed Global Markov property holds if, for $S, A, B \subset V$, S separating A and B in $(\mathcal{G}_{A_{A\cup B\cup S}})^m$,

$$A \perp\!\!\!\perp B|S.$$

The following theorem is the directed analogue of Theorem 2.3.

Theorem 2.6. *Consider a directed acyclic graph $\mathcal{G}$, with a given density function. Then*

$$(DF) \iff (DL) \iff (DG).$$

The proof is omitted, and the interested reader is referred to [50].

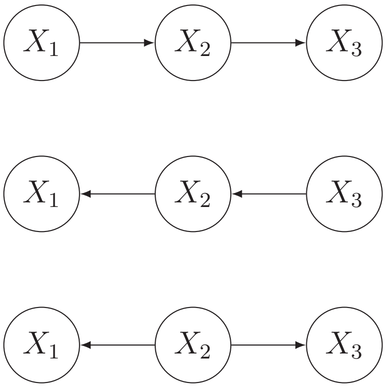
Together, Theorems 2.5 and 2.6 show how the Markov properties of directed and undirected graphs relate to one another: for all but the directed Pairwise Markov property, if a directed Markov property holds for a directed graph with a positive and continuous density function, the corresponding Markov property will hold for the moralised version of that graph.

2.2.6 Equivalence of Directed Acyclic Graphs

Definition 2.7. *Two directed acyclic graphs are said to be Markov Equivalent if they encode the same conditional independence constraints.*

From this point, we will take equivalent to mean Markov equivalent.

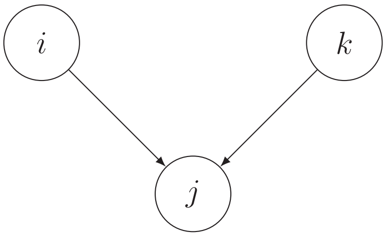
For example, consider the following three directed acyclic graphs:



The only conditional independence relationship that each of these graphs represent is that $X_1 \perp\!\!\!\perp X_3|X_2$. Hence, these graphs are examples of equivalent graphs.

Before equivalent graphs can be characterised, a definition is required.

Definition 2.8. *A directed acyclic graph $\mathcal{G} = (V, E)$ is said to contain a v-structure if, for some $i, j, k \in V$, $(i, j) \in E$ and $(k, j) \in E$, but $(i, k) \notin E$ and $(k, i) \notin E$. That is, $\mathcal{G}$ has a v-structure if there exists a subgraph of $\mathcal{G}$ of the following form:*



The following theorem, due to Verma and Pearl [85], characterises equivalent graphs.

Theorem 2.7. *Two directed acyclic graphs are equivalent if they have the same skeleton, and the same v-structures.*

See [85] for the proof of this theorem.

The following theorem, due to Chickering [8], shows that edge reversals are all that is required to move through an equivalence class of directed acyclic graphs:

Theorem 2.8. *Let $\mathcal{G}$ and $\mathcal{G}'$ be any pair of equivalent directed acylic graphs, and let $\Delta(\mathcal{G}, \mathcal{G}')$ be the set of edges in $\mathcal{G}$ that have opposite orientation in $\mathcal{G}'$. Then there exists a sequence of $|\Delta(\mathcal{G}, \mathcal{G}')|$ distinct edge reversals in $\mathcal{G}$ with the following properties:*

1. *Each edge (i, j) reversed in $\mathcal{G}$ is such that $P_j = P_i \cup \{i\}$;*
2. *After each edge reversal $\mathcal{G}$ is a directed acyclic graph, equivalent to $\mathcal{G}'$;*

such that after all edge reversals, $\mathcal{G} = \mathcal{G}'$.

See [8] for a proof.

2.2.7 Bayesian Networks

Bayesian networks, first introduced by Pearl in [63], much like graphical models, provide a representation of a joint probability distribution and exploit conditional independence relations in a similar way.

Definition 2.9. *A Bayesian network $B = (\mathcal{G}, \Theta)$ for a random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ and $\Theta = (\theta_1, \theta_2, \ldots, \theta_n)$ consists of two components:*

- *a directed acyclic graph $\mathcal{G} = (V, E)$, with $V = \{1, 2, \ldots, n\}$,*
- *conditional distributions for each random variable, $f(x_j | \boldsymbol{x}_{P_j}, \theta_j)$, where P_j is the set of parents of X_j in $\mathcal{G}$, and θ_j are the parameters associated with the distributions.*

The graph and the conditional distributions specify a joint distribution for $\boldsymbol{X}$:

$$f(\boldsymbol{x}|\Theta) = \prod_{j=1}^{n} f(x_j | \boldsymbol{x}_{P_j}, \theta_j).$$

By definition, a Bayesian network obeys (DF), and so by Theorem 2.6, also obeys (DL) and (DG).

Note that due to the acyclic nature of Bayesian networks, each network has associated with it at least one ordering of the variables, denoted by $Ord(\boldsymbol{X})$, such that there may only be a directed edge from i to j if j occurs after i in the ordering. That is, a node j will occur after its parents P_j in the ordering.

Example 2.6. *A Bayesian network*

Suppose that the Bayesian network for random vector $\boldsymbol{X} = (X_1, X_2, X_3, X_4, X_5)$ has the directed acyclic graph in Figure 2.2 and the required conditional distributions. Hence, the joint distribution is given by

$$f(\boldsymbol{x}) = f(x_1)f(x_2)f(x_3|x_1)f(x_4|x_1, x_2)f(x_5|x_3, x_4).$$

There are many orderings of the variables that are consistent with this Bayesian network. For example, $\{5, 4, 3, 2, 1\}$ and $\{5, 3, 4, 1, 2\}$ are both consistent orderings.

2.2.8 Linear Recursive Equations

As a consequence of their definition, Bayesian networks may be used for the representation of the relationships of a set of variables in a system of linear recursive equations, a fact first shown by Wermuth [88]. By a system of linear recursive equations, we mean, for the purposes of this work, a set of equations involving random variables $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)^T$, ordered such that

$$\begin{aligned}
X_1 &= \gamma_{12}X_2 + \gamma_{13}X_3 + \ldots + \gamma_{1k}X_k + \gamma_{1,k+1}X_{k+1} + \ldots + \gamma_{1n}X_n + \epsilon_1 \\
X_2 &= \gamma_{23}X_3 + \ldots + \gamma_{2k}X_k + \gamma_{2,k+1}X_{k+1} + \ldots + \gamma_{2n}X_n + \epsilon_2 \\
&\vdots \\
X_k &= \gamma_{k,k+1}X_{k+1} + \ldots + \gamma_{kn}X_n + \epsilon_k \\
&\vdots \\
X_{n-1} &= \gamma_{n-1,n}X_n + \epsilon_{n-1} \\
X_n &= \epsilon_n,
\end{aligned}$$

where $\epsilon_i \sim N(0, \psi_i)$ $\forall i = 1, 2, \ldots, n$ independently. If none of the γ_{ij} are set equal to zero, the system is called complete, while if some of the γ_{ij} are restricted to zero, the system is called incomplete.

Note that this system of equations can be written as

$$\boldsymbol{X} = \Gamma \boldsymbol{X} + \boldsymbol{\epsilon}, \tag{2.2}$$

where $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T$, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \Psi)$, $\Psi = diag(\psi_1, \psi_2, \ldots, \psi_n)$, and Γ is an upper triangular matrix:

$$\Gamma = \begin{pmatrix} 0 & \gamma_{12} & \gamma_{13} & \cdots & \gamma_{1k} & \gamma_{1,k+1} & \cdots & \gamma_{1n} \\ 0 & 0 & \gamma_{23} & \cdots & \gamma_{2k} & \gamma_{2,k+1} & \cdots & \gamma_{2n} \\ \vdots & & & & & & & \\ 0 & 0 & 0 & \cdots & 0 & \gamma_{k,k+1} & \cdots & \gamma_{kn} \\ \vdots & & & & & & & \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & \gamma_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Due to the triangular nature of the system, the joint probability density function of $\boldsymbol{X}$ can be written as

$$f(\boldsymbol{x}) = f(x_1|x_2, \ldots, x_n) f(x_2|x_3, \ldots, x_n) \cdots f(x_k|x_{k+1}, \ldots, x_n) \cdots f(x_{n-1}|x_n) f(x_n). \tag{2.3}$$

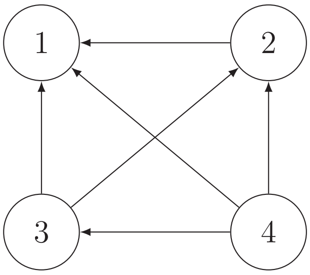
A Bayesian network over $\boldsymbol{X}$ then consists of a directed acyclic graph $\mathcal{G} = (V, E)$, where $V = \{1, 2, \ldots, n\}$, with $(i, j) \in E$ if and only if $\gamma_{ji} \neq 0$, and the conditional distributions given in Equation (2.3).

Example 2.7. *Linear recursive system*

Consider the following complete linear recursive system of equations:

$$\begin{aligned} X_1 &= \gamma_{12} X_2 + \gamma_{13} X_3 + \gamma_{14} X_4 + \epsilon_1 \\ X_2 &= \gamma_{23} X_3 + \gamma_{24} X_4 + \epsilon_2 \\ X_3 &= \gamma_{34} X_4 + \epsilon_3 \\ X_4 &= \epsilon_4. \end{aligned}$$

This system then has the following directed acyclic graph:



which has a complete moral graph.

Note that the moral graph of the directed acyclic graph associated with a complete system of recursive equations will always be complete, while the moral graph associated with an incomplete system of equations will not always be incomplete.

Just as a Gaussian system of recursive linear equations can be associated with a directed acyclic graph, any directed acyclic graph may be associated with a system of linear equations: if $(i, j) \in E$, then X_i will occur in the regression of X_j.

For example, the system of linear equations associated with the directed acyclic graph in Figure 2.2 is

$$\begin{aligned} X_5 &= \gamma_{53} X_3 + \gamma_{54} X_4 + \epsilon_5 \\ X_4 &= \gamma_{41} X_1 + \gamma_{42} X_2 + \epsilon_4 \\ X_3 &= \gamma_{31} X_1 + \epsilon_3 \\ X_2 &= \epsilon_2 \\ X_1 &= \epsilon_1. \end{aligned}$$

It should be noted that the definition of a system of linear recursive equations can be generalised to include random variables $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_m)^T$ that the random variables $\boldsymbol{X}$ may depend upon:

$$\begin{aligned}
X_1 = \gamma_{12}X_2 + \gamma_{13}X_3 + \ldots + \gamma_{1k}X_k + \gamma_{1,k+1}X_{k+1} + \ldots + \gamma_{1n}X_n + \lambda_{11}Z_1 + \ldots + \lambda_{m1}Z_m + \epsilon_1 \\
X_2 = \gamma_{23}X_3 + \ldots + \gamma_{2k}X_k + \gamma_{2,k+1}X_{k+1} + \ldots + \gamma_{2n}X_n + \lambda_{12}Z_1 + \ldots + \lambda_{m2}Z_m + \epsilon_2 \\
\vdots \\
X_k = \gamma_{k,k+1}X_{k+1} + \ldots + \gamma_{kn}X_n + \lambda_{1k}Z_1 + \ldots + \lambda_{mk}Z_m + \epsilon_k \\
\vdots \\
X_{n-1} = \gamma_{n-1,n}X_n + \lambda_{1,n-1}Z_1 + \ldots + \lambda_{m,n-1}Z_m + \epsilon_{n-1} \\
X_n = \lambda_{1n}Z_1 + \ldots + \lambda_{mn}Z_m + \epsilon_n,
\end{aligned} \tag{2.4}$$

$\epsilon_i \sim N(0, \psi_i)\ \forall i = 1, 2, \ldots, n$. Such variables $\boldsymbol{Z}$ are termed exogeneous: their effect on the system of linear recursive equations may be thought of as lying on the surface, influencing the random variables $\boldsymbol{X}$, without adding any additional structure to the system.

This generalised system can be written as

$$\boldsymbol{X} = \Gamma\boldsymbol{X} + \Lambda\boldsymbol{Z} + \boldsymbol{\epsilon}, \tag{2.5}$$

where Γ and $\boldsymbol{\epsilon}$ are as in Equation (2.2), and

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{21} & \cdots & \lambda_{m1} \\ \lambda_{12} & \lambda_{22} & \cdots & \lambda_{m2} \\ \vdots & & & \\ \lambda_{1n} & \lambda_{2n} & \cdots & \lambda_{mn} \end{pmatrix}.$$

2.3 Using Gaussian Graphical Models and Bayesian Networks to Model Genetic Regulatory Networks

Gene expression, as described briefly in Chapter 1, is a complex process, and the extent to which genes are expressed within a cell depend upon the extent to which certain other genes are expressed within that cell. If changes in the expression level of a particular gene cause changes in the expression level of another gene, the first gene is said to regulate the second. Of course, regulatory interactions are typically very complex, involving many genes. Systems of regulatory interactions are known as genetic regulatory networks, and knowledge of these genetic regulatory networks is essential if we wish to understand the functions of genes.

Knowledge of genetic regulatory networks typically arises from the analysis of gene expression data obtained from microarray experiments. The focus here is on observational gene expression data. Data obtained through experimental intervention, such as that obtained from perturbation experiments, which are discussed in [42], is not considered.

Gaussian graphical models and Bayesian networks provide tools for the representation of genetic regulatory networks that are able to cope with the noisy gene expression data obtained from microarray experiments. In both cases, noise is accounted for through the assumption of a normally distributed error structure for the log-scale gene expression data.

When used to model genetic regulatory networks, Gaussian graphical models and Bayesian networks examine conditional independence relationships of the genes being considered. For example, consider gene expression data for p genes, and let X_i represent the expression level of gene i. If $X_i \perp\!\!\!\perp X_j | X_k$, $k \subseteq \{1, 2, \ldots, p\} \setminus \{i, j\}$, then given the expression levels of genes k, the expression level of gene i does not affect the expression level of gene j.

In a graphical model of a genetic regulatory network on these p genes, the expression level of gene i is conditionally independent of the expression level of gene j given all remaining genes if and only if there is no edge between X_i and X_j in the associated independence graph. In a Bayesian network representation of a genetic regulatory network, the expression level of gene i is conditionally independent of the expression level of gene j given the expression levels of all remaining genes if and only if there is no edge between X_i and X_j in the moral graph of the directed acyclic graph of the Bayesian network. Gaussian graphical models and Bayesian networks are useful in the analysis of genetic regulatory networks because these conditional independence relationships allow genetic regulatory networks to be discussed in terms of locally interacting genes.

There are limitations to the use of Bayesian networks or Gaussian graphical models in modelling genetic regulatory networks. Bayesian networks are, by definition, acyclic, so that feedback relationships, known to occur in biological networks, may not be modelled when Bayesian networks are used. Dynamic Bayesian networks, which are not further discussed here, have been used, for example by Husmeier [41], in an attempt to allow the modelling of such biological structure using the language of Bayesian networks.

Both techniques assume that expression levels of different genes are linearly related. There do exist techniques that allow for more complicated relationships between genes. For example, the use of differential equations in the modelling of genetic regulatory networks, as reviewed in [19], allows for the description of non-linear relationships between the expression levels of genes.

For a review of the use of graph-based methods for the modelling of genetic regulatory networks, see [56]. For a review of several methods for the modelling of genetic regulatory networks, the interested reader is directed to [19].

Methods for the estimation of Gaussian graphical model and Bayesian network representations of genetic regulatory networks given observational gene expression data are described in the next chapter.

Chapter 3

Estimating Graphs for Gene Expression Data

Consider a gene expression data set produced by a microarray experiment, in which the expression levels of p genes are recorded for n samples. Note that the emphasis here is on pre-processed, normalised gene expression values: it is assumed that systematic sources of bias and error have been removed from the data. If $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ is a vector of log-scale expression levels for the p genes of interest, in what follows it is supposed that $\boldsymbol{X} \sim N_p(\mathbf{0}, \Sigma)$, with $\Omega = \Sigma^{-1}$. Given a complete data set $d = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^n\}$, consisting of n independent samples of $\boldsymbol{X}$, the interest, for the purposes of this work, lies in the estimation of the underlying dependence structure of all p genes, Σ. This is difficult, since, for gene expression data, p is typically much larger than n.

By Theorem 2.4, the problem of estimating a Gaussian covariance matrix Σ, subject to zero constraints on off-diagonal elements of its inverse, is equivalent to the estimation of a graphical model for the gene expression data. There are two slightly different approaches to this problem: the first involves the estimation of a Bayesian network for the genes, while the second estimates the dependence structure of the genes using a Gaussian graphical model framework. Within each of these approaches there are several methods for the estimation of the structure, many of which are briefly reviewed below.

This problem is a rich area of research, and, as such, old techniques are constantly being improved, and new techniques are constantly being developed. An overview of the area is presented, with emphasis on score-based methods for the estimation of Bayesian networks. Note that the method of particular interest herein is High-dimensional Bayesian Covariance Selection ([23, 25]) which exploits the links between Gaussian graphical models and Bayesian networks, and takes a score-based approach.

3.1 The Bayesian Network Approach

Bayesian networks are a popular tool for the analysis of gene expression data. Techniques for the estimation of such structures for high-dimensional data abound, with any new issue of Machine Learning or the Journal of Machine Learning Research containing new methods. Hence, there are a number of different approaches to the estimation of a Bayesian network for a given set of data. Following [75, 83], the methods can be grouped into two main categories: score-based methods, and constraint-based methods. Score-based methods attempt to maximise some score metric associated with the estimated Bayesian network, while constraint-based methods estimate conditional independence relationships directly from the data. These constraint-based methods are similar to the methods for the estimation of Gaussian graphical models, as they often involve the estimation of an undirected graph, which is then directed.

Recall the definition of a Bayesian network, given in Section 2.2.7. It can be seen that finding such a Bayesian network for a given set of genes is equivalent to the discovery of sets of parents for each gene and an ordering of the genes, such that gene i may be a parent of gene j if and only if i occurs after j in the ordering. Further, as described in Section 2.2.7, the ordering and the parent sets imply the conditional independence relationships of the genes. The aim is to find a Bayesian network, or a set of Bayesian networks, that describe the data well.

It has been shown [67] that the number of directed acyclic graphs on p vertices is

$$b_p = \sum_{i=1}^{p} (-1)^{i+1} \begin{pmatrix} p \\ i \end{pmatrix} 2^{i(p-i)} b_{p-i},$$

where $p > 0$ and $b_0 = 1$. Using this equation, it can be shown that the number of directed acyclic graphs on p vertices increases exponentially: for example, there are 3 possible directed acyclic graphs on 2 vertices, 25 on 3 vertices, 543 on 4 vertices and 29281 on 5 vertices. It has also been shown that the problem of finding a Bayesian network that maximises a given score metric is NP-hard [10]. These difficulties, coupled with the high-dimensional nature of gene expression data, mean that the best an algorithm for finding a Bayesian network can be hoped to do is explore a small part of the space of possible Bayesian networks for the data, and find networks that describe the data well relative to those explored.

In our approach to finding Bayesian networks, it is assumed that the expression levels of each gene depend upon the expression levels of their parent genes in a linear fashion:

$$X_i = \sum_{j \in P_i} \gamma_{ij} X_j + \epsilon_i, \;\; \epsilon_i \sim N(0, \psi_i), \;\; \forall i \in \{1, 2, \ldots, p\}, \tag{3.1}$$

where an ordering may be imposed upon the genes such that this set of equations forms a system of linear recursive equations. Without loss of generality, the ordering is taken to be such that this system of linear recursive equations can be written in the following structural form:

$$\boldsymbol{X} = \Gamma \boldsymbol{X} + \boldsymbol{\epsilon}, \;\; \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \Psi) \tag{3.2}$$

where $\boldsymbol{X}$, $\boldsymbol{\epsilon}$, Ψ and Γ are as in Equation (2.2), with $\gamma_{ij} = 0$ if and only if $j \notin P_i$.

Hence, the Bayesian network $B = (\mathcal{G}, \Theta)$ that we wish to find has the directed acyclic graph $\mathcal{G}$ with edges (j, i) such that $j \in P_i$, and $\Theta = \cup_{i=1}^{p} \{\boldsymbol{\gamma}_i, \psi_i\}$, where $\boldsymbol{\gamma}_i = (\gamma_{ij})^T_{j \in P_i}$.

In general, Bayesian networks are used merely as a tool for the discovery of a genetic structure: directed edges do not necessarily imply causal relationships. The estimation of such relationships would require extensive experimentation, while herein the emphasis is upon observational gene expression data.

3.1.1 Score-Based Methods

Using a score-based approach, how well a given Bayesian network B describes the data is determined through the calculation of the score of this network given the data: $S(B|d)$. Note that in general this score metric is not the familiar score statistic of likelihood theory: $\frac{\partial \ell(\theta)}{\partial \theta}$. In the techniques described below, the score metric is typically taken to be proportional to the posterior probability of the network:

$$S(B|d) = p(d|B)p(B), \tag{3.3}$$

although there do exist other scores: for example, the Bayesian Information Criterion may be used [29].

Since the emphasis herein is on observational gene expression data, it is difficult to apply notions such as causation to the Bayesian networks found for the data. Instead, the Bayesian networks found for gene expression data sets are interpreted as encoding conditional independence relationships between genes. As such, Bayesian networks that encode the same conditional independence relationships should not be discriminated between. This notion, known as score equivalence, [9], implies that if B and B' are Bayesian networks with equivalent directed acyclic graphs, as defined in Section 2.2.6, then

$$S(B|d) = S(B'|d). \tag{3.4}$$

In what follows, score metrics that satisfy the notion of score equivalence are constructed.

From the definition of the score metric in Equation (3.3), the score metrics considered herein consist of two components: $p(d|B)$ and $p(B)$. The term $p(d|B)$ is the marginal likelihood of the data d given a Bayesian network B, obtained through marginalising over the parameters Θ associated with B:

$$p(d|B) = \int p(d|B, \Theta)p(\Theta|B)d\Theta, \tag{3.5}$$

and the term $p(B)$ is the prior probability of the Bayesian network B. Hence, a prior distribution over the space of all possible Bayesian networks and the marginal likelihood of the data given a Bayesian network must be specified.

The prior distribution over the space of Bayesian networks varies between the different approaches taken, and the specification thereof is not considered to be as important as the specification of the marginal likelihood [31]. In order to satisfy score equivalence, the requirement placed upon the prior is that Bayesian networks with equivalent directed acyclic graphs have the same prior probability [39]. That is, if B is Markov equivalent to B',

$$p(B) \equiv p(B'). \tag{3.6}$$

Equation (3.5) implies that the calculation of the marginal likelihood $p(d|B)$ requires the specification of two distributions: $p(d|B,\Theta)$, the distribution of the data d given the network B and the parameters Θ, and the prior distribution of the parameters Θ given the network B, $p(\Theta|B)$. These distributions are specified so that for equivalent Bayesian networks B and B',

$$p(d|B) = p(d|B'),$$

a condition known as likelihood equivalence, [39]. This condition together with a prior distribution on the space of Bayesian networks as in Equation (3.6) implies a score metric that satisfies score equivalence.

Geiger and Heckerman, [35], discuss five assumptions that together imply likelihood equivalence. Three of these assumptions concern the distribution of $d|B,\Theta$, and the remaining two concern the prior distribution of the parameters $\Theta|B$.

To find $p(d|B,\Theta)$, note that Equation (3.2) can be reduced to obtain $\boldsymbol{X} = (I_p - \Gamma)^{-1}\boldsymbol{\epsilon}$ so that

$$\boldsymbol{X} \sim N_p(\mathbf{0}, \Sigma), \text{ where } \Sigma = (I_p - \Gamma)^{-1}\Psi(I_p - \Gamma)^{-T}, \tag{3.7}$$

and $\Omega = \Sigma^{-1} = (I_p - \Gamma)^T\Psi(I_p - \Gamma)$. Note that in what follows, $(I_p - \Gamma)^{-T} = ((I_p - \Gamma)^{-1})^T$. Hence,

$$d|B,\Theta \sim N_{np}\left(\mathbf{0}, \Sigma \otimes I_n\right),$$

which, due to the ordering of the variables, may be written as

$$f(d|B,\Theta) = \prod_{i=1}^{p} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i), \tag{3.8}$$

where $\boldsymbol{x}_i$ is the n-vector of observations of X_i and $\boldsymbol{x}_{P_i}$ is the $n \times |P_i|$ matrix with columns $\boldsymbol{x}_j$, $j \in P_i$. Equation (3.1) implies that

$$\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i \sim N_n(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i, \psi_i I_n). \tag{3.9}$$

This distribution satisfies the assumptions set out by Geiger and Heckerman, [35], so that $p(\Theta = \cup_{i=1}^{p}\{\boldsymbol{\gamma}_i, \psi_i\}|B)$ is then all that requires specification. It has been shown, [18, 35, 69], that an encompassing inverse Wishart prior distribution on Σ, the full covariance structure of the genes, satisfies the assumptions of Geiger and Heckerman. The implications of this prior for the form of the prior distributions of the parameters $\{\boldsymbol{\gamma_i}, \psi_i\}$ for each $i \in \{1, 2, \ldots, p\}$, will be detailed below, in the description of High-dimensional Bayesian Covariance Selection, in Section 3.3.

It should be noted that the assumptions of Geiger and Heckerman imply that the marginal model likelihood in Equation (3.5) has the following form:

$$p(d|B) = \prod_{i=1}^{p} f\left(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}\right),$$

where $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ is the marginal model likelihood of gene i:

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) = \int_0^\infty \int_{\mathbb{R}^{|P_i|}} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i) f(\boldsymbol{\gamma}_i, \psi_i|\boldsymbol{x}_{P_i}) d\boldsymbol{\gamma}_i d\psi_i. \tag{3.10}$$

Note that $f(\boldsymbol{\gamma}_i, \psi_i|\boldsymbol{x}_{P_i})$ is the joint prior distribution of the parameters associated with gene i when the genes in P_i are the parents of gene i.

The score of a Bayesian network B given a data set d then has the form

$$S(B|d) = p(B) \prod_{i=1}^{p} f\left(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}\right). \tag{3.11}$$

Note that this score metric has been referred to as the BGe metric by Geiger and Heckerman, [34], standing for "Bayesian metric for Gaussian networks having score equivalence".

The exact form of the marginal model likelihoods in Equation (3.10) are derived in Section 3.3.

Some score-based methods for the estimation of Bayesian networks for a given gene expression data set are now described.

Greedy hill-climbing, first introduced by Cooper and Herskovits [12] for the search over Bayesian networks, is a standard, and very general, score-based method in the estimation of Bayesian networks for high-dimensional data [5, 29, 83], and many algorithms based upon this method have been developed. The algorithm takes as input, along with a data set and a score function as in Equation (3.3), an initial Bayesian network, which may be the empty network. To move through the space of Bayesian networks, a set of operations on Bayesian networks are defined. In the archetypal formulation, these moves are edge addition, edge deletion, and edge reversal. At each iteration of the algorithm, the score function is evaluated for each possible operation (that still results in a Bayesian network), and the operation increasing the score the most is performed. The algorithm ceases when no permitted operation on the current Bayesian network results in a network with a higher score.

The algorithm is prone to entrapment in local maxima. To overcome this limitation greedy hill-climbing is typicially run several times with different initial graphs. Taboo lists and simulated annealing are two popular augmentations of the algorithm.

Greedy hill-climbing often forms the basis of different methods for the estimation of Bayesian networks. One augmentation is to first find a set of candidate parents for each gene, and then apply greedy hill-climbing, restricting the parent sets of each gene to be subsets of these candidate sets. This approach is taken in the Sparse Candidate algorithm [32, 33]. Max-Min Hill-Climbing [83] behaves similarly, although it may be more properly be classified as a hybrid of the score-based and constraint-based approaches, as it first estimates a skeleton, which is then directed using greedy hill-climbing. The Ideal Parent algorithm [29] further augments greedy hill-climbing by calculating ideal parents for each of the genes of interest, and comparing candidate parents to these ideal parents to help determine if proposed operations should occur.

Often, instead of performing hill-climbing over the space of Bayesian networks, another smaller space will be searched over. Chickering [9] defines a search over the space of equivalence classes of directed acyclic graphs, while Teyisser and Koller [78] search over the space of orderings of the variables. Friedman and Koller [31] also search over the orderings of the variables, but their interest lies in the estimation of the posterior probability of particular features of a Bayesian network.

Optimal Reinsertion [59] is a technique that estimates a Bayesian network by selecting a gene at each step, severing all edges directed to or from this gene, then estimating the optimal edges to or from this gene before reinserting the gene back into the network. Moore and Wong [59] discuss incorporating the Sparse Candidate algorithm into Optimal Reinsertion.

In [48, 73, 74], dynamic programming approaches are taken to maximising the score of a Bayesian network, and for networks with around 30 nodes, promising results are obtained. However, in the applications of interest in the present work, networks with an order of magnitude more nodes are considered.

3.1.2 Constraint-Based Methods

Constraint-based methods, which are discussed in detail in [75], instead of finding a Bayesian network that maximises a score metric, estimate conditional independence relationships directly from the data. These estimated conditional independence relationships are then used as constraints in the development of a Bayesian network for the genes of interest.

For example, the PC-Algorithm, [45, 75], consists of two algorithms, run consecutively. The first algorithm starts with a complete undirected graph on all genes. Partial correlation coefficients, described in Section 3.2.1, are calculated for each pair of genes. If the partial correlation coefficient for a pair of genes i and j is not significantly different from zero, the edge between i and j is removed from the undirected graph. The resultant undirected graph is treated as a skeleton graph by the second algorithm, which extends the graph through the imposition of directed edges, resulting in a representative of an equivalence class of directed acyclic graphs.

Three-Phase Dependency Analysis, [7], formulated for discrete variables, uses conditional mutual information as a measure of conditional dependence. The conditional mutual information of two variables X_i and X_j given some other variable set, $\boldsymbol{Y}$, is defined to be

$$I(X_i, X_j|\boldsymbol{Y}) = \sum_{x_i, x_j, \boldsymbol{y}} p(x_i, x_j|\boldsymbol{y}) \log \frac{p(x_i, x_j|\boldsymbol{y})}{p(x_i|\boldsymbol{y})p(x_j|\boldsymbol{y})}.$$

If $p(\cdot)$ is the true distribution from which samples are drawn, the conditional mutual information is zero if $X_i \perp\!\!\!\perp X_j|\boldsymbol{Y}$. Since $p(\cdot)$ is estimated, X_i is deemed to be conditionally independent of X_j given Y if $I(X_i, X_j|\boldsymbol{Y}) < \epsilon$, where ϵ is a user-specified positive value. These estimated conditional independence relationships are then algorithmically combined to form a Bayesian network.

In the present work, constraint-based methods for the estimation of Bayesian networks will not be considered further.

3.2 The Gaussian Graphical Model Approach

Estimating a Gaussian graphical model for a set of p genes differs from the estimation of a Bayesian network for the same genes in two key ways. First, since a Bayesian model is not being assumed, there is no need to specify prior distributions on parameters. Second, estimation of the direction of edges is no longer required. Note also that the definition of a Bayesian network for a set of genes does not allow the representation of structures such as feedback loops, while there are no such restrictions on Gaussian graphical models. It should, however, be noted that the requirement of a joint Gaussian distribution of the variables under consideration remains.

The estimation of a Gaussian graphical model for a set of genes, as noted previously, is equivalent to the estimation of which elements of the concentration matrix of the joint distribution are equal to zero. The high-dimensional nature of gene expression data prohibits the estimation of the matrix directly, and in this section, several approaches to this problem are outlined.

Many of the approaches can be classified as either shrinkage methods or limited-order partial correlation methods. The shrinkage methods attempt to indirectly estimate the full inverse covariance matrix, and, in particular, the zero elements of that matrix. Limited-order partial correlation methods, instead of estimating the full-order partial correlations, assume that in a Gaussian graphical model with few edges, these low-order conditional independence relationships will estimate the full-order relationships quite well.

3.2.1 Limited-Order Partial Correlation-Based Methods

In a Gaussian graphical model for a set of genes, there is no edge between gene i and gene j if and only if gene i is conditionally independent of gene j given all remaining genes. It was shown, in Theorem 2.4, that this occurs if and only if the (i, j)-entry of the concentration matrix, ω_{ij}, is equal to zero. Noting that the partial correlation coefficient of gene i and gene j, $\rho_{ij|V\setminus\{i,j\}} = \frac{-\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}$, an equivalent formulation would be to specify that this partial correlation coefficient be equal to zero.

Limited-order partial correlation-based methods, instead of attempting to estimate these partial correlations, estimate lower-order partial correlations. This approach is based on the assumption that, in a graphical model with few edges, low-order conditional independence relationships will approximate full-order conditional independence relationships well.

A q-order partial correlation of i and j is defined by considering a set Q, $|Q| = q$, such that $Q \subset V$, $i, j \notin Q$:

$$\rho_{ij|Q} = \frac{-\omega_{ij}^{Q\cup\{i,j\}}}{\sqrt{\omega_{ii}^{Q\cup\{i,j\}}\omega_{jj}^{Q\cup\{i,j\}}}},$$

where $\omega_{ij}^{Q\cup\{i,j\}}$ is the (i,j)-entry of the concentration matrix of $X_{Q\cup\{i,j\}}$. It is clear that $X_i \perp\!\!\!\perp X_j|X_Q \iff \rho_{ij|Q} = 0$.

The graphs based on these limited-order partial correlation coefficients are called q-partial graphs, a term coined by Castelo and Roverato [6]. A q-partial graph is defined such that there is an edge between genes i and j if and only if $\rho_{ij|Q} = 0$, for all sets Q such that $|Q| = q$. Castelo and Roverato [6] develop the theory of q-partial graphs as well as a procedure to estimate such graphs for a given data set.

Castelo and Roverato, in their development of the theory of q-partial graphs, unified many previous approaches to the estimation of Gaussian graphical models. Wille and Bühlmann [90] developed a method for the discovery of $0, 1$-partial graphs, while de la Fuente *et al.* [20] described a method that uses up to second-order partial correlations. Magwene and Kim [55] considered first-order conditional independence relationships in their method.

3.2.2 Shrinkage-Based Methods

It is well known that for high-dimensional data the maximum likelihood estimate of the covariance matrix is singular, and, as such, does not provide a good estimate of the concentration matrix. Shrinkage-based methods provide non-singular estimates of the full covariance matrix of high-dimensional data sets through the shrinkage of partial correlation coefficients towards zero.

Wong, Carter and Kohn [91] take a Bayesian approach to the shrinkage estimation of the concentration matrix. The concentration matrix is factorised as $\Omega = T \times C \times T$, where $T = diag(\omega_{ii})$, and $C = (c_{ij})$ is a correlation matrix, such that $-c_{ij} = \rho_{ij|V\setminus\{i,j\}}$. Prior distributions are then placed upon ω_{ii}, $i = 1, 2, \ldots, n$, and upon each c_{ij}, $i < j$. The concentration matrix is then estimated using a reversible jump Metropolis-Hastings method.

Schäfer and Strimmer [71] specify their approach in terms of the partial correlation matrix, introducing new small sample point estimators thereof. They note that when $n < p$, the sample covariance matrix, $\hat{\Sigma} = \frac{1}{n-1}(\boldsymbol{X} - \bar{\boldsymbol{X}})^T(\boldsymbol{X} - \bar{\boldsymbol{X}})$, may be inverted, and then used to obtain an estimate of the partial correlation matrix. To estimate the partial correlation matrix from a small sample, they apply the Moore-Penrose pseudo-inverse [64] to the sample covariance matrix, stabilising the estimator through the application of the bootstrap and bagging [4]. An empirical Bayes approach is then used to determine which elements of the estimated partial correlation matrix are equal to zero.

In their subsequent approach to the problem, Schäfer and Strimmer [72], develop an alternative small-sample estimator, following the work of Ledoit and Wolf [52] on the shrinkage estimation of small-sample covariance matrices. The performance of this new estimator is compared to that of their earlier estimators, and to the approach of Meinshausen and Bühlmann [57]. Note that [72] provides a good review of the shrinkage approach to the estimation of Gaussian graphical models.

Meinshausen and Bühlmann [57], taking an approach similar to that taken in High-dimensional Bayesian Covariance Selection, cast the problem of estimating a Gaussian graphical model for a set of variables as a variable selection problem, choosing regression models for each variable. They use the lasso [80] to obtain these regression models, hence obtaining shrinkage estimates of regression parameters. The predictors in a regression model for a particular variable form the neighbourhood of that variable in the graphical model obtained. Applying their technique to a simulated data set with 1000 nodes and 600 observations, they were able to correctly identify 1109 edges out of 1747.

3.2.3 Other Methods

Some earlier approaches to the estimation of Gaussian graphical models for gene expression data do not fit well into either of the above categories. For completeness they are briefly summarised here.

Kishino and Waddell [46] use multiple regressions with variable selection procedures to estimate partial correlation coefficients for each pair of genes.

Toh and Horimoto [81, 82] deal with high-dimensional gene expression data using a combination of cluster analysis and Gaussian graphical models. Their approach reduces the dimension of the problem by clustering the genes, then finds a Gaussian graphical model for these clusters, instead of for the genes. This method results in conditional independence relationships that are quite difficult to interpret, as not all genes of one cluster are necessarily associated with all genes of another cluster.

3.3 High-Dimensional Bayesian Covariance Selection

High-dimensional Bayesian Covariance Selection, [23, 25], seeks to identify high-scoring, high-dimensional, sparse Gaussian graphical models for a given gene expression data set. The High-dimensional Bayesian Covariance Selection algorithm uses a score-based method to find Bayesian networks for the data, which are then moralised to obtain Gaussian graphical models.

For each $i \in \{1, 2, \ldots, p\}$, it is assumed that the expression level of gene i depends linearly on the expression levels of genes in gene i's parent set, P_i, with normally distributed random error:

$$X_i = \sum_{j \in P_i} \gamma_{ij} X_j + \epsilon_i, \;\; \epsilon_i \sim N(0, \psi_i). \tag{3.12}$$

As discussed in Section 3.1.1, it is assumed that this set of equations forms a system of linear recursive equations, so that

$$\boldsymbol{X} \sim N_p\left(\mathbf{0}, \Sigma\right), \text{ where } \Sigma = \left(I_p - \Gamma\right)^{-1} \Psi \left(I_p - \Gamma\right)^{-T}, \tag{3.13}$$

where Γ is an upper triangular matrix as in Equation (3.7), with $\gamma_{ij} = 0$ if and only if $j \notin P_i$.

Recall that if $\boldsymbol{x}_i$ is the n-vector of gene expression levels for gene i, Equation (3.12) implies

$$\boldsymbol{x}_i | \boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i \sim N_n(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i, \psi_i I_n), \tag{3.14}$$

where $\boldsymbol{x}_{P_i}$ is the $n \times |P_i|$ matrix with columns $\boldsymbol{x}_j$, $j \in P_i$, and $\boldsymbol{\gamma}_i = (\gamma_{ij})^T_{j \in P_i}$.

The focus of High-dimensional Bayesian Covariance Selection is on the estimation of sparse Bayesian networks for the given gene expression data. A sparse Bayesian network or graphical model is one that has few edges relative to the number of vertices in the graph. Recall, from Theorem 2.4, that a missing edge in the independence graph associated with a Gaussian graphical model corresponds to a zero in the concentration matrix of that model. Hence, a sparse Gaussian graphical model is one with a sparse concentration matrix. From Equation (3.13), the concentration matrix of the genes under consideration is assumed to have the form

$$(I_p - \Gamma)^T \, \Psi^{-1} \, (I_p - \Gamma) \, .$$

This matrix will be sparse when Γ contains few non-zero entries, which will occur when $|P_i|$ is small for all values of i.

The notion of sparsity reduces the number of parameters requiring estimation, allows access to a sparse Cholesky decomposition of the concentration matrix, and reflects the view that the expression level of a particular gene will be directly influenced by relatively few other genes. Sparse Bayesian networks are found through the use of a score metric that gives higher scores to such sparse networks. The construction of such a score metric is detailed in the following section.

3.3.1 Construction of the High-dimensional Bayesian Covariance Selection Score Metric

Recall, from Equation (3.11), that for the purposes of this work, the score of a Bayesian network B for a data set d, $S(B|d)$, has the form

$$S(B|d) = p(B) \prod_{i=1}^{p} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}),$$

where $p(B)$ is the prior probability of the network and $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ is the marginal model likelihood of gene i.

The estimation of sparse Bayesian networks in High-dimensional Bayesian Covariance Selection is achieved through the construction of a prior distribution on the space of Bayesian networks that induces $a_i = |P_i|$, to be small $\forall i$. The probability of gene i having any particular gene in its parent set P_i is β, where β small induces sparser structures. Dobra *et al.* [23, 25] implement this through the imposition of a binomial prior distribution on the space of all possible Bayesian networks:

$$p(B) \propto \left(\frac{\beta}{1 - \beta} \right)^{|B|} ,$$

where $|B| = \sum_{i=1}^{p} a_i$ is the number of edges in B. It is a consequence of Theorem 2.8 that equivalent directed acyclic graphs have the same number of vertices, so that if B and B' are Bayesian networks with equivalent directed acyclic graphs, $p(B) = p(B')$.

The construction of the marginal likelihood of the data given the network, $p(d|B)$, then proceeds through the application of an inverse Wishart prior distribution on the full covariance structure Σ, as discussed in Section 3.1.1. This inverse Wishart prior distribution on Σ induces prior distributions on the regression parameters for each gene, $\boldsymbol{\gamma}_i$ and ψ_i. Dobra *et al.*, [23, 25], use the following prior on the covariance matrix:

$$\Sigma \sim IW_p(\delta, \tau I_p),$$

where $IW_p(\delta, \tau I_p)$ represents an Inverse-Wishart distribution, with probability density function

$$f(\Sigma) = \left(2^{\frac{\delta p}{2}} \pi^{\frac{p(p-1)}{4}} \prod_{i=1}^{p} \Gamma\left(\frac{\delta + 1 - i}{2}\right) \right)^{-1} \tau^{\frac{\delta p}{2}} |\Sigma|^{-\left(\frac{\delta+p+1}{2}\right)} \exp\left\{ -\frac{1}{2} tr(\tau \Sigma^{-1}) \right\},$$

with degrees of freedom δ and scale matrix τI_p, $\delta > 0$ and $\tau > 0$. A noninformative prior distribution is obtained as $\delta \to 0$. The parameter δ can be thought of as a shrinkage parameter; as δ gets larger, graphs with larger numbers of edges are encouraged, so setting δ to be small encourages sparsity. The diagonal scale matrix, τI_p, is reasonable if, a priori, gene expressions are thought to be independent with a common scale. Note that τ should be selected to be on a scale consistent with the data [44].

Given that an encompassing inverse Wishart prior distribution is used, consider, then, the variance matrix of all genes associated with the regression of gene i:

$$\Sigma_{\{i\} \cup P_i} = \begin{pmatrix} \sigma_{ii} & \boldsymbol{\sigma}_i^T \\ \boldsymbol{\sigma}_i & \Sigma_{P_i} \end{pmatrix},$$

where $\boldsymbol{\sigma}_i$ is the vector of length a_i consisting of the covariances of X_i with X_j, $j \in P_i$. The prior implies that $\Sigma_{\{i\} \cup P_i} \sim IW_{a_i+1}(\delta, \tau I_{a_i+1})$. Then, by Proposition C.5 in [50], $X_i | \boldsymbol{x}_{P_i} \sim N(\boldsymbol{x}_{P_i}^T \boldsymbol{\gamma}_i, \psi_i)$, where

$$\boldsymbol{\gamma}_i = \Sigma_{P_i}^{-1} \boldsymbol{\sigma}_i \text{ and } \psi_i = \sigma_{ii} - \boldsymbol{\sigma}_i^T \Sigma_{P_i}^{-1} \boldsymbol{\sigma}_i.$$

The implied priors on the regression parameters for gene i are then given by

$$\begin{aligned}\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I_{a_i}), \text{ and}\\ \psi_i^{-1} &\sim Ga\left(\frac{\delta+a_i}{2}, \frac{\tau}{2}\right).\end{aligned} \tag{3.15}$$

The marginal model likelihood, $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$, can then be found for each gene i:

$$\begin{aligned}f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) =& \int f(\boldsymbol{x}_i|\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{x}_{P_i}) f(\boldsymbol{\gamma}_i|\psi_i) f(\psi_i) d\boldsymbol{\gamma_i} d\psi_i\\ =& \int (2\pi)^{-\left(\frac{n+a_i}{2}\right)} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \psi^{-\left(\frac{n+2a_i+\delta}{2}+1\right)}\\ &\times \exp\left\{-\frac{\tau}{2\psi_i} - \frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i - \frac{1}{2\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\right\} d\boldsymbol{\gamma}_i d\psi_i.\end{aligned}$$

Where the terms involving $\boldsymbol{\gamma}_i$ can be manipulated as follows:

$$\begin{aligned}&-\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i - \frac{1}{2\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\\ =& -\frac{1}{2\psi_i}\left(\boldsymbol{\gamma}_i - \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\boldsymbol{x}_i\right)^T\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)\left(\boldsymbol{\gamma}_i - \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\boldsymbol{x}_i\right)\\ &-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left[I_n - \boldsymbol{x}_{P_i}\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\right]\boldsymbol{x}_i,\end{aligned}$$

so that

$$\begin{aligned}&f(\boldsymbol{x}_i\boldsymbol{\gamma}_i, \psi_i|\boldsymbol{x}_{P_i}) =\\ &(2\pi)^{-\left(\frac{n+a_i}{2}\right)} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \psi_i^{-\left(\frac{n+2a_i+\delta}{2}+1\right)} \exp\left\{-\frac{1}{2\psi_i}\left(\tau + \boldsymbol{x}_i^T\left[I_n - \boldsymbol{x}_{P_i}\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\right]\boldsymbol{x}_i\right)\right\}\\ &\times \exp\left\{-\frac{1}{2\psi_i}\left(\boldsymbol{\gamma}_i - \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\boldsymbol{x}_i\right)^T\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)\left(\boldsymbol{\gamma}_i - \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\boldsymbol{x}_i\right)\right\},\end{aligned} \tag{3.16}$$

The terms involving $\boldsymbol{\gamma}_i$ then form the kernel of an a_i-dimensional normal distribution with covariance matrix

$$\psi_i\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}.$$

Integrating over $\boldsymbol{\gamma}_i$ then results in

$$\begin{aligned}f(\boldsymbol{x}_i|\psi_i, \boldsymbol{x}_{P_i}) =& (2\pi)^{-\left(\frac{n}{2}\right)} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \left|\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\\ &\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)} \exp\left\{-\frac{1}{2\psi_i}\left(\tau + \boldsymbol{x}_i^T\left[I_n - \boldsymbol{x}_{P_i}\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\right]\boldsymbol{x}_i\right)\right\}.\end{aligned}$$

This has the form of an Inverse-Gamma(α,β) kernel in ψ_i, with

$$\begin{aligned}\alpha &= \frac{n+a_i+\delta}{2}\\ \beta &= \frac{\tau}{2}+\frac{1}{2}\boldsymbol{x}_i^T\left[I_n-\boldsymbol{x}_{P_i}\left(\tau I_{a_i}+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\right]\boldsymbol{x}_i.\end{aligned}$$

Integrating over ψ_i then gives the marginal model likelihood:

$$\begin{aligned}f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) = &(2\pi)^{-\left(\frac{n}{2}\right)}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\left|\tau I_{a_i}+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\Gamma\left(\frac{n+a_i+\delta}{2}\right)\\ &\times\left(\frac{\tau}{2}+\frac{1}{2}\boldsymbol{x}_i^T\left[I_n-\boldsymbol{x}_{P_i}\left(\tau I_{a_i}+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\right]\boldsymbol{x}_i\right)^{-\left(\frac{n+a_i+\delta}{2}\right)} \qquad (3.17)\end{aligned}$$

which has the form of a multivariate t-distribution,

$$\boldsymbol{x}_i|\boldsymbol{x}_{P_i} \sim t_{\delta+a_i}(\mathbf{0},\Sigma_i), \qquad (3.18)$$

where

$$\Sigma_i = \frac{\tau}{\delta+a_i}\left(I_n-\boldsymbol{x}_{P_i}(\tau I_{a_i}+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i})^{-1}\boldsymbol{x}_{P_i}^T\right)^{-1}.$$

In any Bayesian network, there will always exist at least one variable with no parents. Hence, the marginal model likelihood for gene i when $P_i=\emptyset$ also needs to be calculated. In such a case, $P_i=\emptyset$ implying that $a_i=0$, and

$$\boldsymbol{x}_i \sim t_\delta\left(\mathbf{0},\frac{\tau}{\delta}I_n\right). \qquad (3.19)$$

The High-dimensional Bayesian Covariance Selection score metric, hereafter referred to as S_0, then has a closed form:

$$\begin{aligned}S_0(B|d) &= \prod_{i=1}^{p} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})\left(\frac{\beta}{1-\beta}\right)^{|P_i|}\\ &= \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^{p}a_i}\prod_{i=1}^{p} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}), \qquad (3.20)\end{aligned}$$

where $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ is given by Equation (3.17) for all i such that $P_i\neq\emptyset$, and

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) \equiv f(\boldsymbol{x}_i) = (2\pi)^{-\left(\frac{n}{2}\right)}\left(\frac{\tau}{2}\right)^{\frac{\delta}{2}}\frac{\Gamma\left(\frac{n+\delta}{2}\right)}{\Gamma\left(\frac{\delta}{2}\right)}\left(\frac{\tau}{2}+\frac{1}{2}\boldsymbol{x}_i^T\boldsymbol{x}_i\right)^{-\left(\frac{n+\delta}{2}\right)} \qquad (3.21)$$

for all i such that $P_i=\emptyset$.

As noted above, in Section 3.1.1, this score metric is also known as the BGe metric.

3.3.2 Posterior Distributions

If an estimate of the joint covariance matrix of a subset of genes $K \subseteq \{1, 2, \ldots, p\}$ is required, posterior estimates of the regression parameters $\boldsymbol{\gamma}_i$ and ψ_i $\forall i \in K$ must be obtained. Note the posterior independence of the regression parameters associated with different genes, so that

$$f\left(\boldsymbol{\gamma}_{i \in K}, \psi_{i \in K}|d\right) = \prod_{i \in K} f(\boldsymbol{\gamma}_i, \psi_i|d).$$

The posterior distributions required to obtain such estimates are those of $\boldsymbol{\gamma}_i|\psi_i, \boldsymbol{x}_{P_i}$, $\psi_i|\boldsymbol{x}_{P_i}$ and ψ_i, $\forall i \in K$, all of which have a closed form. To find these posterior distributions, note that

$$\begin{aligned}
&f(\boldsymbol{\gamma}_i|\psi_i, \boldsymbol{x}_i, \boldsymbol{x}_{P_i}) \propto f(\boldsymbol{x}_i, \boldsymbol{\gamma}_i|\boldsymbol{x}_{P_i}, \psi_i) \\
&\propto \exp\left\{-\frac{1}{2\psi_i}\left(\boldsymbol{\gamma}_i - \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T \boldsymbol{x}_i\right)^T \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i}\right) \left(\boldsymbol{\gamma}_i - \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T \boldsymbol{x}_i\right)\right\},
\end{aligned}$$

which, as noted previously, is the kernel of a normal distribution in $\boldsymbol{\gamma}_i$, so that

$$\boldsymbol{\gamma}_i|\psi_i, \boldsymbol{x}_i, \boldsymbol{x}_{P_i} \sim N_{a_i}\left((\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i})^{-1} \boldsymbol{x}_{P_i}^T \boldsymbol{x}_i, \psi_i (\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i})^{-1}\right). \tag{3.22}$$

Similarly,

$$\begin{aligned}
f(\psi_i|\boldsymbol{x}_i, \boldsymbol{x}_{P_i}) \propto & f(\boldsymbol{x}_i, \psi_i|\boldsymbol{x}_{P_i}) \\
& \propto \psi^{-\left(\frac{n+a_i+\delta}{2}+1\right)} \exp\left\{-\frac{1}{2\psi_i}\left(\tau + \boldsymbol{x}_i^T \left[I_n - \boldsymbol{x}_{P_i}\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T\right] \boldsymbol{x}_i\right)\right\},
\end{aligned}$$

which has already been recognised as an Inverse-Gamma kernel in ψ_i, so

$$\psi_i^{-1}|\boldsymbol{x}_i, \boldsymbol{x}_{P_i} \sim Ga\left(\frac{\delta + a_i + n}{2}, \frac{\tau + \boldsymbol{x}_i^T \left(I_{a_i} - \boldsymbol{x}_{P_i}\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T\right) \boldsymbol{x}_i}{2}\right). \tag{3.23}$$

The final posterior distribution required is the posterior distribution of the variance of gene i, when gene i has no parents:

$$\begin{aligned}
f(\psi_i|\boldsymbol{x}_i) \propto & f(\boldsymbol{x}_i, \psi_i) \\
& \propto \psi^{-\left(\frac{n+\delta}{2}+1\right)} \exp\left\{-\frac{1}{2\psi_i}\left(\tau + \boldsymbol{x}_i^T \boldsymbol{x}_i\right)\right\}
\end{aligned}$$

so that

$$\psi_i^{-1}|\boldsymbol{x}_i \sim Ga\left(\frac{\delta + n}{2}, \frac{\tau + \boldsymbol{x}_i^T \boldsymbol{x}_i}{2}\right).$$

3.3.3 The High-dimensional Bayesian Covariance Selection Algorithm

The High-dimensional Bayesian Covariance Selection algorithm, described in detail below, is based upon the estimation and combination of good regression models for each gene. The algorithm consists of the following stages:

- The first stage of the algorithm involves estimating initial sets of parents, P_i, for each gene. This is equivalent to finding good regression models for each gene.
- It is in the second stage of the algorithm that the regression models found in stage 1 are modified so that they imply a system of linear recursive equations, and hence a Bayesian network. That is, the sets P_i are modified so that they are consistent with an ordering: $j \in P_i$ if and only if j occurs after i in the ordering.
- The Bayesian network found in stage 2 is then moralised to obtain a Gaussian graphical model for the data, and this, along with the posterior distributions for the regression parameters, is used for inference.

Stage 1: Initialisation of the Network

In this stage, initial regression models for each gene are found. These regression models are unrestricted in the sense that there is, at this stage, no ordering associated with this set of regressions, as there would be if the regressions formed a recursive linear system. Hence, any gene can be a predictor for any other gene. Let $R_i \subseteq V \setminus \{i\}$ be the set of predictors for gene i found at this stage.

Initially, Gibbs sampling was used to find these regression models. Later, shotgun stochastic search, [38, 44], was recommended, and currently, work in Mode-Oriented Stochastic Search, [22, 24, 53], is progressing. These approaches to the identification of suitable regression models for each gene are described below.

Gibbs Sampling To start the Gibbs Sampling, for each i, the sets R_i are randomly selected. For each gene $i \in V$, cycling through $j \in V \setminus \{i\}$ in random order, the j^{th} gene is then included in the regression model for $\boldsymbol{x}_i$, that is, $R_i \to R_i \cup \{j\}$, with probability

$$P(j \in R_i | R_i \setminus \{j\}) = \frac{a}{a+b}$$

where

$$a = f(\boldsymbol{x}_i | \boldsymbol{x}_{R_i \cup \{j\}}) p(\text{having } |R_i \cup \{j\}| \text{ predictors}) = f(\boldsymbol{x}_i | \boldsymbol{x}_{R_i \cup \{j\}}) \left(\frac{\beta}{1-\beta} \right)^{|R_i \cup \{j\}|},$$

$$b = f(\boldsymbol{x}_i | \boldsymbol{x}_{R_i \setminus \{j\}}) p(\text{having } |R_i \setminus \{j\}| \text{ predictors}) = f(\boldsymbol{x}_i | \boldsymbol{x}_{R_i \setminus \{j\}}) \left(\frac{\beta}{1-\beta} \right)^{|R_i \setminus \{j\}|}.$$

Note that $f(\boldsymbol{x}_i | \boldsymbol{x}_{R_i \setminus \{j\}})$ is as given by Equation (3.17), where P_i is taken to be $R_i \setminus \{j\}$.

One Gibbs sampling iteration for a gene i is a full cycle through all possible predictors in $V \setminus \{i\}$.

This sampling can be run independently for each variable, but due to the large number of potential predictors for each gene in the data sets, it is possible that potentially strong predictors for a gene will not be included in R_i.

Shotgun Stochastic Search This method for finding regressions for each gene proceeds as follows. Consider gene i. For each regression model R_i, where R_i contains k predictors for this gene, the neighbourhood of that regression model, $N(R_i)$ is defined as follows: $N(R_i) = \{R_i^+, R_i^\circ, R_i^-\}$, where

- R_i^+ is the set of neighbouring models containing $k+1$ predictors, obtained by adding any gene to R_i,
- R_i° is the set of neighbouring models containing k predictors, obtained by replacing any of the predictors in R_i by another
- R_i^- is the set of neighbouring models containing $k-1$ predictors, obtained by taking any one of the predictors out of R_i.

When $2 \leq k < p$, $|R_i^+| = p - k$, $|R_i^\circ| = k(p-k)$ and $|R_i^-| = k$, and $R_i^+ = \emptyset$ if $k = p$. It is noted that when p is large, these three sets are of very different sizes: $|R_i^\circ| \gg |R_i^+| \gg |R_i^-|$. Further noted is that as $p \to \infty$, if all models were to have equal weight, and a model were to be selected from $N(R_i)$, the probability of randomly selecting a model from $R_i^\circ \to \frac{k}{k+1}$, $R_i^+ \to \frac{1}{k+1}$ and $R_i^- \to 0$.

The score of a particular model R_i for gene i is defined to be the unnormalised posterior probability:

$$\begin{aligned} S(R_i) = p(R_i|\boldsymbol{x}_i) &\propto f(\boldsymbol{x}_i|R_i)p(R_i) \\ &\propto f(\boldsymbol{x}_i|R_i)\left(\frac{\beta}{1-\beta}\right)^{|R_i|}, \end{aligned}$$

where $f(\boldsymbol{x}_i|R_i) \equiv f(\boldsymbol{x}_i|\boldsymbol{x}_{R_i})$, as given by Equation (3.17).

The method for finding a regression model for gene i proceeds as follows:

0 Evaluate the scores for the null model, and all one-variable models. Choose K, the maximum number of models. Given a starting model $R_i^{[0]}$, set $\Gamma^* = \{R_i^{[0]}\}$, and for $t = 1, 2, \ldots, T$:

1 Compute $S(R_i)$ for all $R_i \in N(R_i^{[t]})$, so that R_i^+, R_i°, R_i^- are constructed. $\Gamma^* \to \Gamma^* \cup N(R_i^{[t]})$; if $|\Gamma^*| > K$, remove the $|\Gamma^*| - K$ lowest scoring models.

2 Sample models R_i^{+*}, $R_i^{\circ *}$ and R_i^{-*} from R_i^+, R_i°, R_i^- with probabilities proportional to $S(R_i)$, normalised within each set.

3 Sample $R_i^{[t+1]}$ from $\{R_i^{+*}, R_i^{\circ *}, R_i^{-*}\}$ with probability proportional to $S(R_i)$, normalised within this set.

At the completion of the algorithm, Γ^* contains the K highest scoring models in $\bigcup_{t=0}^{t-1} N(R_i^{[t]})$.

It is clear that shotgun stochastic search allows movement across dimension to occur with greater probability than in the Gibbs sampling paradigm.

Mode Oriented Stochastic Search This method for finding regression models for each gene seeks to identify models such that the ratio of the posterior probability of these models and the posterior probability of the best model is above a threshold. That is, if $\mathcal{R}$ is the set of all possible regression models for a given gene, the algorithm identifies models in

$$\mathcal{R}(c) = \left\{ R \in \mathcal{R} | \ p(R|\boldsymbol{x}_i) \geq c \max_{R' \in \mathcal{R}} p(R'|\boldsymbol{x}_i) \right\},$$

where $c \in (0, 1)$.

The algorithm, which is not presented here, is described in detail in [22, 24] and [53].

Note that both the Stochastic Shotgun Search and Mode Oriented Stochastic Search algorithms may be used to directly estimate Σ, the joint covariance matrix of all genes, instead of being used to estimate regression models for each gene, which are then combined via the High-dimensional Bayesian Covariance algorithm. This involves directly exploring the space of Gaussian graphical models, instead of indirectly exploring such models through the generation of systems of linear recursive equations, as High-dimensional Bayesian Covariance Selection does. For details on how the Shotgun Stochastic Search algorithm may be used in this way, see [44], and for details on how this is done using the Mode Oriented Stochastic Search algorithm, see [53].

Stage 2: Finding a Bayesian Network

The second stage of the High-dimensional Bayesian Covariance Selection algorithm involves finding a Bayesian network for the genes, based upon the sets R_i found in the first stage of the algorithm. Set an iterate counter $t = 0$. Let $Ord(\boldsymbol{X})$ be the vector containing the ordering of the genes, initially the null vector. Let $C = \{1, 2, \ldots, p\}$ be the candidate variable set, which contains the indices of unordered genes. Assign each gene $X_j, j \in C$ an explanatory score:

$$s_j = \prod_{i \in C} f(\boldsymbol{x}_i | \boldsymbol{x}_{R_i \setminus \{j\}}) \left(\frac{\beta}{1-\beta} \right)^{|R_i \setminus \{j\}|}.$$

The score of a gene is related to the usefulness of that gene's expression level in the explanation of all other genes' expression levels. The more useful a particular gene, the lower the score of that gene.

The algorithm for finding a Bayesian network is as follows.

For $t = 1, 2, \ldots, p-1$:

1. Choose the gene with the highest score: $j_t = argmax_{j \in C}\{s_j\}$

2. x_{j_t} is the next variable in the ordering: $Ord(\boldsymbol{X}) \rightarrow \{Ord(\boldsymbol{X}), j_t\}$. For this variable, take P_{j_t} to be R_{j_t}.

3. Remove j_t from the candidate variable set: $C \rightarrow C \setminus \{j_t\}$.

4. $\forall i \in C$ with $j_t \in R_i$, find a new set R_i such that $R_i \subset C$. Then calculate updated scores s_i for all such genes.

At $t = p$, there remains one unordered gene; the gene with the lowest score that is, ostensibly, the gene that is most useful in the prediction of the expression levels of other genes. This gene is appended to the ordering $Ord(\boldsymbol{X})$. This ordering, the sets P_i and the conditional distributions of each gene given its predictors then define a Bayesian network on the genes. Various aspects of the model can then be explored, such as dependence structures for subsets of genes, or the posterior analysis of the regression coefficients $\{\boldsymbol{\gamma}_i, \psi_i\}$.

Stage 3: Estimation

It is in this stage of the algorithm that the Bayesian network obtained at stage 2 is moralised, and various aspects of the resultant Gaussian graphical model explored.

Recall from Equation (3.7) that the concentration matrix for all genes may be expressed as $\Omega = (I_p - \Gamma)^T \Psi^{-1}(I_p - \Gamma)$. Note that a Cholesky decomposition is available:

$$\Omega = LL^T, \text{ where } L = (I_p - \Gamma)^T \Psi^{-1/2}.$$

Note further that since each gene is expected to have a small number of parent genes, Γ will be a sparse upper triangular matrix, and hence L will be a sparse lower triangular matrix. This implies that finding the joint covariance matrix $\Sigma = \Omega^{-1} = L^{-T}L^{-1}$, while perhaps not particularly easy for the entire set of genes, is feasible for smaller sets of genes, W:

$$\Sigma_W = L_W^{-T} L_W^{-1},$$

where L_W is the matrix consisting of the rows of L corresponding to the variables in W. Estimates thereof may be obtained through sampling from the posterior distributions of ψ_i and $\boldsymbol{\gamma}_i|\psi_i$, $i \in W$.

Generation of Multiple Graphs

This algorithm is deterministic, meaning that the result of the algorithm is a single Bayesian network. If multiple Bayesian networks for the given gene expression data could be found, then uncertainty associated with the model could be investigated, using a technique such as Bayesian model averaging [40, 54, 66].

In the algorithm, at each iteration, the gene with the highest score is chosen. This results in the generation of a single ordering of the genes. If, instead of simply selecting the gene with the highest score, a gene is randomly sampled, multiple orderings will be generated with multiple runs of the algorithm. Hence the extension of the algorithm:

1^*. Sample a gene j_t from C with probability proportional to $(s_{j_t})^A$

for an annealing parameter $A > 0$. This "annealed" approach introduces stochasticity into the algorithm, leading to the generation of multiple Bayesian networks for the genes.

Dobra *et al* apply the algorithm to a breast cancer data set, see [23], with $p = 12558$ and $n = 158$. Taking $A = 25$, 150 graphs were saved, with biologically plausible sub-graphs.

3.3.4 The High-dimensional Bayesian Covariance Selection Program

The High-dimensional Bayesian Covariance Selection program is available to download from `http://www.stat.duke.edu/~adobra/hdbcs.html`. There are two versions of the program: a serial version, and a parallel version, which makes use of Message Passing Interface (MPI) libraries. The serial version, which is claimed to work well for $p < 250$, is used here. In each case, the program is written in C++, and requires the use of the Fortran77 LAPACK library, and the C++ Blitz++ library.

The program requires that the data matrix have n rows and p columns; a column for each gene, and a row for each sample. Also required is that the gene expression data for each gene be centered and scaled such that each column of the matrix has sample mean 0 and sample variance 1. Also required are the parameters τ, δ and β. The `README` file included in the download lists the other required inputs, which are mostly related to the number of graphs that the user decides to save, and how many iterations of the algorithm are desired.

In Chapter 7, the program is used in the analysis of data sets with a known network structure, and in Chapter 8, is used in the analysis of a gene expression data set.

3.4 Extensions and Use of the Methods

Since the publication of the methods reviewed in this Chapter, much subsequent work has focussed on the improvement and further development of methods for the estimation of Bayesian networks and Gaussian graphical models. For example, methods such as Anjum *et al*'s BoostiGraph, [2] and Bottolo and Richardson's Evolutionary Stochastic Search, [3], have been developed for the estimation of Gaussian graphical models given high-dimensional data sets. In addition, Meinshausen and Bühlmann's recently developed Stability Selection, [58], is a widely-applicable method that may be used to estimate Gaussian graphical models, controlling the number of falsely discovered edges.

Much work has also been done on the incorporation of prior knowledge in the estimation of Bayesian networks and Gaussian graphical models. Angelopoulos and Cussens, [1], review the inclusion of prior structural information in the estimation of Bayesian networks, and use Bayesian model averaging as a tool in the estimation of Bayesian networks for some examples. Mukherjee and Speed, [60], discuss the inclusion of prior biological information in the estimation of Bayesian networks, and present an example involving proteins.

Koller and Friedman recently published a book, [47], drawing together much recent research on Bayesian networks and graphical models, including several chapters on the learning of such structures.

It seems that the biological applications of the methods reviewed in this chapter have been largely confined to the initial biological applications present in the originally published articles. That is, popular uptake of the methods by biologists does not seem to have occurred as yet, perhaps due to the statistical and mathematical complexity of these methods.

Chapter 4

Score Metrics for Data Sets with Complex Mean Structures

4.1 Motivation for the Inclusion of Complex Mean Structures

High-dimensional Bayesian Covariance Selection, like many score-based methods for the estimation of Bayesian networks for the log-scale gene expression levels of p genes, $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^T \sim N_p(\mathbf{0}, \Sigma)$, assumes that the data set, $d = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^n\}$, consists of n independent and identically distributed samples of expression levels for the p genes.

Typically, the data available will be more complicated. For example, consider the grape gene data introduced in Chapter 1. The data available consists of gene expression values for 132 genes, with 174 samples in total; 68 samples taken from grapes grown at one vineyard, 68 samples taken from grapes grown at another vineyard, and 38 samples taken from grapes grown at a third vineyard. It is apparent that since different vineyards have different characteristics, expression levels of individual genes will be expected to vary not only from sample to sample, but also from vineyard to vineyard.

If we wished to use High-dimensional Bayesian Covariance Selection or one of the other score-based methods discussed in Chapter 3 to estimate the covariance structure of the grape genes given such a data set, one of a number of approaches could be taken. For example,

- The method could be applied to the data from each vineyard separately;
- The possible effects of vineyards upon gene expressions could be ignored, and the method applied to the entire data set;
- An ad-hoc method to remove the possible vineyard effects could be used; for example, the mean expression level at each vineyard could be subtracted, and the method applied to this augmented data set, or
- Vineyards could be included as vertices in a Bayesian network or Gaussian graphical model.

None of these approaches are particularly appealing; each has obvious associated problems. For example,

- Applying the method to data from each vineyard separately may result in the estimation of different graphs at each vineyard, which may negate the utility of said graphs. That is, if different graphs are obtained for gene expressions taken from two different vineyards, using either of these graphs to explain relationships between genes at any other vineyard would be difficult;
- If possible vineyard effects are ignored, graphs with spurious edges may result. Further, if either of these first two approaches are taken, the effect of each vineyard on the expression level of each gene may not be estimable;
- There may be several ad-hoc methods available for the removal of possible vineyard effects and different choices may lead to different results;
- The difficulty with the final suggestion, including variables such as vineyard as vertices, is that models for mixed variables may be required if the variable is discrete, and sparsity constraints may have to be modified, if such variables are expected to affect the expression levels of all genes.

The aim of the current chapter is the development of a new score metric for the estimation of Bayesian networks that may be applied to data sets more complex than those consisting of independent and identically distributed samples of gene expression levels. Initially, a score metric that takes account of the possible effects that different sites may have on the expression levels of different genes is presented. This score metric, motivated by the example discussed above involving grape genes, includes the effect of each site on the expression level of each gene as a random effect, [49]. This score metric is then extended to include more general random effects. Note that the score metrics developed herein may be used in conjunction with any score-based method for the estimation of Bayesian networks; here the method of choice is High-dimensional Bayesian Covariance Selection.

Consider, then, a gene expression data set of the form of the grape gene data set, consisting of n samples of p genes, with samples taken from m different sites; n_s samples taken at site s, such that $\sum_{s=1}^{m} n_s = n$. If X_{is} is the expression level of gene i measured on a sample taken from site s, the regression model in Equation (3.1) becomes

$$X_{is} = \sum_{j \in P_i} \gamma_{ij} X_{js} + b_{is} + \epsilon_{is}, \;\; \epsilon_{is} \sim N(0, \psi_i), \tag{4.1}$$

where $i = 1, 2, \ldots, p$ and $s = 1, 2, \ldots, m$, and b_{is} is the effect of site s on the expression level of gene i. These site effects will be treated as random effects, having some probability density function $f(b_{is})$, not dependent upon ϵ_{is}, with variance parameter ϕ_i. That is, the expression level of gene i measured at site s is linearly dependent upon the expression levels of the genes in P_i, with normally distributed error, and some gene-specific site effect, independent of the error.

The model for each gene can then be expressed as

$$\boldsymbol{x}_i | \boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i \sim N_n(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q\boldsymbol{b}_i, \psi_i I), \tag{4.2}$$

where $\boldsymbol{x}_i$ is the n-vector of expression levels for gene i:

$$\boldsymbol{x}_i = \left(x_{i11}, x_{i21}, \ldots, x_{in_11}, x_{i12}, x_{i22}, \ldots, x_{in_22}, \ldots\ldots, x_{i1m}, x_{i2m}, \ldots, x_{in_mm}\right)^T,$$

$\boldsymbol{x}_{P_i}$ is the $n \times |P_i|$ matrix with columns $\boldsymbol{x}_j$, $j \in P_i$, $\boldsymbol{b}_i$ is the m-vector of site effects for gene i:

$$\boldsymbol{b}_i = \left(b_{i1}, b_{i2}, \ldots, b_{im}\right)^T,$$

and Q is the $n \times m$ matrix:

$$Q = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Through the choice of an appropriate Q matrix, the formulation in Equations (4.1) and (4.2) can be generalised to include any random effects of interest.

For example, consider a data set consisting of n samples of p grape genes, now all taken from one vineyard, and the temperature at the time each sample was taken. Suppose that the effect of the temperature upon the expression levels of different genes is of interest, and that temperature may influence how genes relate to one another. The data set now consists of np gene expression levels and n temperatures; x_{ik} is the expression level of gene i in sample k, and q_k the temperature at the time sample k was taken, regarded as being fixed. Through the choice of an appropriate Q matrix, the effect of temperature upon gene expression values may be included as a random effect in the analysis of the dependence structure of the grape genes.

If the effect of temperature is included in the model, the model for each gene can be expressed as in Equation (4.2), with

$$\begin{aligned} \boldsymbol{x}_i =& \left(x_{i1}, x_{i2}, \ldots, x_{in}\right)^T, \qquad (4.3) \\ \boldsymbol{b}_i =& b_i \text{ and } Q = \left(q_1, q_2, \ldots, q_n\right)^T. \end{aligned}$$

If m general random effects are included in the model, the model for each gene is again as in Equation (4.2), with $\boldsymbol{x}_i$ as in Equation (4.3), and

$$Q = \begin{pmatrix} q_{11} & q_{21} & \cdots & q_{m1} \\ q_{12} & q_{22} & \cdots & q_{m2} \\ \vdots & \vdots & & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{mn} \end{pmatrix}. \tag{4.4}$$

Hence, in general, the linear model for sample k of gene i may be written as

$$X_{ik} = \sum_{j \in P_i} \gamma_{ij} X_{jk} + \sum_{r=1}^{m} q_{rk} b_{ir} + \epsilon_{ik}, \;\; \epsilon_{ik} \sim N(0, \psi_i), \;\; i \in \{1, 2, \ldots, p\}. \tag{4.5}$$

Similarly to when no random effects were assumed to be present, it is assumed that there exists an ordering of the genes such that the system of equations in (4.5) forms a generalised system of linear recursive equations, as given in Equation (2.4). In this way, the random effects that are present may be thought of as exogenous variables, affecting the expression levels of the genes, but not adding any additional structure to the genetic regulatory network that is present.

4.2 Derivation of the Score Metric

The application of the High-dimensional Bayesian Covariance Selection algorithm to the random effects formulation requires the derivation of a score metric. Recall the score metric under the original formulation of High-dimensional Bayesian Covariance Selection:

$$S_0(B|d) = \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^{p} a_i} \prod_{i=1}^{p} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}), \tag{4.6}$$

where $a_i = |P_i|$,

$$\begin{aligned} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) &= \int f(\boldsymbol{x}_i|\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{x}_{P_i}) f(\boldsymbol{\gamma}_i|\psi_i) f(\psi_i) d\boldsymbol{\gamma_i} d\psi_i \\ &= \frac{\Gamma\left(\frac{n+\delta+a_i}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)\pi^{\frac{n}{2}}\tau^{\frac{n}{2}}} \left| I_n - \boldsymbol{x}_{P_i}\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T \right|^{\frac{1}{2}} \\ &\quad \times \left[1 + \frac{1}{\tau}\boldsymbol{x}_i^T \left\{ I_n - \boldsymbol{x}_{P_i}\left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T \right\} \boldsymbol{x}_i \right]^{-\left(\frac{n+\delta+a_i}{2}\right)}, \end{aligned}$$

and, when $P_i = \emptyset$,

$$\begin{aligned} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) \equiv f(\boldsymbol{x}_i) = & \int f(\boldsymbol{x}_i|\psi_i, \boldsymbol{x}_{P_i}) f(\psi_i) d\psi_i \\ = & \frac{\Gamma\left(\frac{n+\delta}{2}\right)}{\Gamma\left(\frac{\delta}{2}\right) \pi^{\frac{n}{2}} \tau^{\frac{n}{2}}} \left(1 + \frac{1}{\tau} \boldsymbol{x}_i^T \boldsymbol{x}_i\right)^{-\left(\frac{n+\delta}{2}\right)} . \end{aligned}$$

Recall that δ and τ are hyper parameters of the prior distribution of Σ, the joint covariance matrix.

The score metric when there exists a more complex mean structure is taken to have the same form as that given in Equation (4.6). Hence, the calculation of the marginal model likelihoods $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ and $f(\boldsymbol{x}_i)$, $\forall i \in \{1, 2, \ldots, p\}$, when

$$\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i \sim N_n(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q\boldsymbol{b}_i, \psi_i I)$$

and

$$\boldsymbol{x}_i|\psi_i, \boldsymbol{b}_i, \phi_i \sim N_n(Q\boldsymbol{b}_i, \psi_i I)$$

are required.

The calculation of the marginal model likelihood requires the specification of the joint distribution of the data for gene i and the parameters relating to that gene, for each $i \in \{1, 2, \ldots, p\}$. Hence, prior distributions for the parameters $\boldsymbol{\gamma}_i$, ψ_i, $\boldsymbol{b}_i$ and ϕ_i need to be specified. Initially, it will be assumed that the random effects for each gene are independent and identically distributed. This assumption will be explored in Chapter 5.

Geiger and Heckerman, [35], show that in order for score equivalence, as defined in Equation (3.4), to be satisfied under the assumption of a more complex mean structure, it is required that $f(\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i|\phi_i)$ have a normal-inverse gamma form $\forall i \in \{1, 2, \ldots, p\}$, and that the random effects for one gene are a priori independent of those for all other genes. Hence, the prior distributions for $\boldsymbol{\gamma}_i|\psi_i$ and ψ_i given in (3.15) are retained, and it is supposed that

$$\boldsymbol{b}_i|\phi_i \sim N_m\left(\boldsymbol{0}, \phi_i I\right), \ \ \forall i \in \{1, 2, \ldots, p\}.$$

There are several possible assumptions about the form of the prior distribution of the variance of the random effects for gene i, $f(\phi_i)$, that may be made. The four assumptions considered herein are:

$M1$ ϕ_i is known, implying that $\boldsymbol{b}_i|\phi_i \equiv \boldsymbol{b}_i \sim N(\boldsymbol{0}, \phi_i I_m)$.

$M2$ The random effects, $\boldsymbol{b}_i$, vary in the same way as the regression coefficients $\boldsymbol{\gamma}_i$: $\boldsymbol{b}_i|\phi_i \equiv \boldsymbol{b}_i|\psi_i \sim N(\boldsymbol{0}, \upsilon^{-1}\psi_i)$.

$M3$ Following [36], a Uniform$(0, \kappa)$ prior distribution is placed upon $\phi_i^{\frac{1}{2}}$, where κ is a specified upper bound, implying that $f(\phi_i) = \frac{1}{2\kappa}\phi_i^{-\frac{1}{2}}$.

$M4$ ϕ_i has an Inverse-Gamma(α, β) prior distribution, with some fixed values of α and β.

For each choice of prior distribution, the marginal model likelihood for each gene can be calculated, through marginalisation over the required parameters:

$$\begin{aligned} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) &= \int f(\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i|\boldsymbol{x}_{P_i}) d\boldsymbol{\gamma}_i d\psi_i d\boldsymbol{b}_i d\phi_i \\ &= \int f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i) f(\boldsymbol{\gamma}_i|\psi_i) f(\psi_i) f(\boldsymbol{b}_i|\phi_i) f(\phi_i) d\boldsymbol{\gamma}_i d\psi_i d\boldsymbol{b}_i d\phi_i. \end{aligned}$$

When gene i has no parents, the required marginal model likelihood may be calculated similarly:

$$\begin{aligned} f(\boldsymbol{x}_i) &= \int f(\boldsymbol{x}_i, \psi_i, \boldsymbol{b}_i, \phi_i) d\psi_i d\boldsymbol{b}_i d\phi_i \\ &= \int f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \psi_i, \boldsymbol{b}_i, \phi_i) f(\psi_i) f(\boldsymbol{b}_i|\phi_i) f(\phi_i) d\psi_i d\boldsymbol{b}_i d\phi_i. \end{aligned}$$

Geiger and Heckerman [35] note that when $\phi_i \neq \upsilon^{-1}\psi_i$, any choice of prior distribution on ϕ_i will result in a marginal model likelihood without a closed form. When this is the case, Gaussian quadrature, as described in Appendix A, is used to compute the marginal model likelihoods. Of course, other approximate methods, such as MCMC, could be used to perform these computations.

Each assumption on the prior form of the distribution of ϕ_i results in a different score metric: let $S_i(B|d)$ be the score metric induced by assumption $i \in \{1, 2, 3, 4\}$. These score metrics are now derived, and after discussions about posterior sampling of parameters and the estimation of the joint covariance matrix of a subset of genes, the utility of each score metric is discussed in Section 4.5.

4.2.1 Assuming ϕ_i known: Derivation of S_1

The simplest assumption that may be made about the variance of the random effects is that it is known. In this case, the required prior distributions are

$$\begin{aligned}\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0},\tau^{-1}\psi_i I),\\ \psi_i &\sim \text{Inv Gamma}\left(\frac{\delta+a_i}{2},\frac{\tau}{2}\right), \text{ and}\\ \boldsymbol{b}_i &\sim N_m(\mathbf{0},\phi_i I),\end{aligned}$$

where ϕ_i is a positive constant.

In this case,

$$\begin{aligned}f(\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i|\boldsymbol{x}_{P_i}) =&(2\pi)^{-\left(\frac{n+a_i+m}{2}\right)}\phi_i^{-\frac{m}{2}}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\psi_i^{-\left(\frac{n+2a_i+\delta}{2}+1\right)}\exp\left\{-\frac{\tau}{2\psi_i}-\frac{1}{2\phi_i}\boldsymbol{b}_i^T\boldsymbol{b}_i\right.\\ &\left.-\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)\right\}.\end{aligned}$$

The terms involving $\boldsymbol{b}_i$ may be written as

$$\begin{aligned}&-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)-\frac{1}{2\phi_i}\boldsymbol{b}_i^T\boldsymbol{b}_i\\ =&-\frac{1}{2}\left\{\boldsymbol{b}_i-\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)^{-1}\frac{1}{\psi_i}Q^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\right\}^T\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)\\ &\times\left\{\boldsymbol{b}_i-\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)^{-1}\frac{1}{\psi_i}Q^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\right\}\\ &-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T\left\{I-\frac{1}{\psi_i}Q\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)^{-1}Q^T\right\}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right).\end{aligned}$$

The quadratic term in $\boldsymbol{b}_i$ can be interpreted as the kernel of a normal distribution and hence, integrating out $\boldsymbol{b}_i$ results in

$$\begin{aligned}f(\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i|\boldsymbol{x}_{P_i}) =&(2\pi)^{-\left(\frac{n+a_i}{2}\right)}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\left|I+\frac{\phi_i}{\psi_i}Q^TQ\right|^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+2a_i+\delta}{2}+1\right)}\\ &\times\exp\left[-\frac{\tau}{2\psi_i}-\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i\right.\\ &\left.-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T\left\{I-\frac{1}{\psi_i}Q\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)^{-1}Q^T\right\}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\right].\end{aligned}$$

Letting

$$J_1^{\psi_i} = I - Q\left(\frac{\psi_i}{\phi_i}I + Q^TQ\right)^{-1}Q^T, \tag{4.7}$$

terms involving $\boldsymbol{\gamma}_i$ may be written as

$$\begin{aligned}
&-\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i - \frac{1}{2\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T J_1^{\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\\
=&-\frac{1}{2\psi_i}\left\{\boldsymbol{\gamma}_i - \left(\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_i\right\}^T\left(\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_{P_i}\right)\\
&\qquad\qquad\times\left\{\boldsymbol{\gamma}_i - \left(\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_i\right\}\\
&-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_1^{\psi_i} - J_1^{\psi_i}\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T J_1^{\psi_i}\right\}\boldsymbol{x}_i.
\end{aligned}$$

The quadratic term in $\boldsymbol{\gamma}_i$ is the kernel of a normal distribution in $\boldsymbol{\gamma}_i$, so integrating out $\boldsymbol{\gamma}_i$ gives

$$\begin{aligned}
f(\boldsymbol{x}_i, \psi_i|\boldsymbol{x}_{P_i}) =&(2\pi)^{-\frac{n}{2}}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\left|I + \frac{\phi_i}{\psi_i}Q^TQ\right|^{-\frac{1}{2}}\left|\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\tau}{2\psi_i} - \frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_1^{\psi_i} - J_1^{\psi_i}\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T J_1^{\psi_i}\right\}\boldsymbol{x}_i\right].
\end{aligned} \tag{4.8}$$

The appearance of ψ_i in $J_1^{\psi_i}$ means that, unlike in Equation (3.16), there does not exist a closed-form expression for $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ under this assumption. Hence, $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ is approximated through numerical integration over ψ_i. In particular, q-point Gauss Laguerre quadrature, as described in Appendix A, is used.

If $w_1, w_2, \ldots, w_q$ are the weights and $u_1, u_2, \ldots, u_q$ the abscissae for such quadrature, then

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) \approx \sum_{s=1}^{q} w_s e^{u_s} f(\boldsymbol{x}_i, u_s|\boldsymbol{x}_{P_i}). \tag{4.9}$$

The marginal model likelihood when gene i has no parents also requires calculation. In such a case,

$$f(\boldsymbol{x}_i, \psi_i) =(2\pi)^{-\frac{n}{2}}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta}{2}}}{\Gamma\left(\frac{\delta}{2}\right)}\left|I + \frac{\phi_i}{\psi_i}Q^TQ\right|^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+\delta}{2}+1\right)}\exp\left(-\frac{\tau}{2\psi_i} - \frac{1}{2\psi_i}\boldsymbol{x}_i^T J_1^{\psi_i}\boldsymbol{x}_i\right),$$

which, while a more compact expression than that given by Equation (4.8), still does not admit a closed-form expression for $f(\boldsymbol{x}_i)$. Hence, numerical integration, as applied to obtain an approximation of $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ in Equation (4.9), is again required:

$$f(\boldsymbol{x}_i) \approx \sum_{s=1}^{q} w_s e^{u_s} f(\boldsymbol{x}_i, u_s).$$

Under this assumption, the score of a Bayesian network B given a data set d is taken to be

$$S_1(B|d) = \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^{p} a_i} \prod_{i=1}^{p} \left\{ \sum_{s=1}^{q} w_s e^{u_s} f(\boldsymbol{x}_i, u_s|\boldsymbol{x}_{P_i}) \right\}, \tag{4.10}$$

where $f(\boldsymbol{x}_i, u_s|\boldsymbol{x}_{P_i}) = f(\boldsymbol{x}_i, u_s)$ when gene i has no parents.

4.2.2 Assuming $\boldsymbol{b}_i$ vary as $\boldsymbol{\gamma}_i$: Derivation of S_2

Assuming that the random effects $\boldsymbol{b}_i$ vary in the same way as the regression coefficents $\boldsymbol{\gamma}_i$ does not require the specification of a new prior distribution for the variance of the random effects. Under this assumption, $\phi_i = \upsilon^{-1}\psi_i$. Hence, the required prior distributions are

$$\begin{aligned} \boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I), \\ \boldsymbol{b}_i|\psi_i &\sim N_m(\mathbf{0}, \upsilon^{-1}\psi_i I), \text{ and} \\ \psi_i &\sim \text{Inv Gamma}\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right), \end{aligned}$$

where υ is a positive parameter. If $\upsilon = \tau$, then $\boldsymbol{b}_i$ and $\boldsymbol{\gamma}_i$ are independent and identically distributed. Taking $\upsilon > \tau$ implies that $\boldsymbol{b}_i$ are less variable than $\boldsymbol{\gamma}_i$, while $\upsilon < \tau$ implies that the $\boldsymbol{b}_i$ are more variable than the $\boldsymbol{\gamma}_i$. Note that υ is treated in the same way as τ was treated by Dobra *et al.* [23, 25]: τ controls how the variance of the $\boldsymbol{\gamma}_i$s relates to the variance of the gene expression values, and is a user-defined value.

Hence,

$$\begin{aligned} f(\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i|\boldsymbol{x}_{P_i}) =& (2\pi)^{-\left(\frac{n+a_i+m}{2}\right)} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \upsilon^{\frac{m}{2}} \psi_i^{-\left(\frac{n+2a_i+m+\delta}{2}+1\right)} \exp\left\{ -\frac{\tau}{2\psi_i} - \frac{\upsilon}{2\psi_i}\boldsymbol{b}_i^T\boldsymbol{b}_i \right. \\ & \left. -\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i - \frac{1}{2\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right) \right\}. \end{aligned}$$

The terms involving $\boldsymbol{b}_i$ may be rearranged to give

$$\begin{aligned}
&-\frac{\upsilon}{2\psi_i}\boldsymbol{b}_i^T\boldsymbol{b}_i-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)\\
=&-\frac{1}{2\psi_i}\left\{\boldsymbol{b}_i-\left(\upsilon I+Q^TQ\right)^{-1}Q^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\right\}^T\left(\upsilon I+Q^TQ\right)\left\{\boldsymbol{b}_i-\left(\upsilon I+Q^TQ\right)^{-1}Q^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\right\}\\
&-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T\left\{I-Q\left(\upsilon I+Q^TQ\right)^{-1}Q^T\right\}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right).
\end{aligned}$$

Substituting this into $f(\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i|\boldsymbol{x}_{P_i})$ gives a normal kernel in $\boldsymbol{b}_i$, so integrating over $\boldsymbol{b}_i$ gives

$$\begin{aligned}
f(\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i|\boldsymbol{x}_{P_i})=&(2\pi)^{-\left(\frac{n+a_i}{2}\right)}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\upsilon^{\frac{m}{2}}\left|\upsilon I+Q^TQ\right|^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+2a+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\tau}{2\psi_i}-\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i\right.\\
&\left.-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T\left\{I-Q\left(\upsilon I+Q^TQ\right)^{-1}Q^T\right\}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\right],
\end{aligned}$$

Letting

$$J_2=I-Q\left(\upsilon I+Q^TQ\right)^{-1}Q^T, \tag{4.11}$$

the terms involving $\boldsymbol{\gamma}_i$ may be written as

$$\begin{aligned}
&-\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)^T\left\{I-Q\left(\upsilon I+Q^TQ\right)^{-1}Q^T\right\}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right)\\
=&-\frac{1}{2\psi_i}\left\{\boldsymbol{\gamma}_i-\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_i\right\}^T\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)\left\{\boldsymbol{\gamma}_i-\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_i\right\}\\
&-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_2-J_2\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2\right\}\boldsymbol{x}_i,
\end{aligned}$$

which, when substituted back into $f(\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i|\boldsymbol{x}_{P_i})$, is the kernel of a normal distribution in $\boldsymbol{\gamma}_i$. Integrating over $\boldsymbol{\gamma}_i$ then gives

$$\begin{aligned}
f(\boldsymbol{x}_i,\psi_i|\boldsymbol{x}_{P_i})=&(2\pi)^{-\frac{n}{2}}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\upsilon^{\frac{m}{2}}\left|\upsilon I+Q^TQ\right|^{-\frac{1}{2}}\left|\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\tau}{2\psi_i}-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_2-J_2\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2\right\}\boldsymbol{x}_i\right].
\end{aligned}$$

This is an Inverse-Gamma kernel in ψ_i, with

$$\begin{aligned}
\alpha=&\frac{n+a_i+\delta}{2}\\
\beta=&\frac{\tau}{2}+\frac{1}{2}\boldsymbol{x}_i^T\left\{J_2-J_2\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2\right\}\boldsymbol{x}_i
\end{aligned}$$

so integrating over ψ_i gives

$$\begin{aligned} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) =&(2\pi)^{-\frac{n}{2}} \frac{\Gamma\left(\frac{\delta+a_i+n}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}} \tau^{\frac{a_i}{2}} \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} \left|\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} \\ &\times \left[\frac{\tau}{2} + \frac{1}{2}\boldsymbol{x}_i^T \left\{J_2 - J_2 \boldsymbol{x}_{P_i} \left(\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T J_2\right\} \boldsymbol{x}_i\right]^{-\left(\frac{\delta+a_i+n}{2}\right)} \\ =&\pi^{-\frac{n}{2}} \tau^{\frac{a_i-n}{2}} \frac{\Gamma\left(\frac{\delta+a_i+n}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} \left|\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} \\ &\times \left[1 + \frac{1}{\tau}\boldsymbol{x}_i^T \left\{J_2 - J_2 \boldsymbol{x}_{P_i} \left(\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T J_2\right\} \boldsymbol{x}_i\right]^{-\left(\frac{\delta+a_i+n}{2}\right)}. \qquad (4.12) \end{aligned}$$

Hence, when $\boldsymbol{b}_i \sim N_m(\mathbf{0}, \upsilon^{-1}\psi_i I)$, $\boldsymbol{x}_i|\boldsymbol{x}_{P_i} \sim t_{\delta+a_i}\left(\mathbf{0}, \Sigma_{\boldsymbol{x}_i|\boldsymbol{x}_{P_i}}\right)$, with

$$\Sigma_{\boldsymbol{x}_i|\boldsymbol{x}_{P_i}} = \frac{\tau}{\delta+a_i}\left\{J_2 - J_2 \boldsymbol{x}_{P_i} \left(\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T J_2\right\}^{-1}.$$

The case when $\boldsymbol{x}_i$ has no parents also needs to be considered. In such a case, $a_i = 0$, and $\boldsymbol{x}_i \sim t_\delta\left(\mathbf{0}, \Sigma_{\boldsymbol{x}_i}\right)$, with

$$\Sigma_{\boldsymbol{x}_i} = \frac{\tau}{\delta} J_2^{-1} = \frac{\tau}{\delta}\left\{I - Q\left(\upsilon I + Q^T Q\right)^{-1} Q^T\right\}^{-1}.$$

Under this assumption, the score of a Bayesian network B given a data set d has a closed form:

$$S_2(B|d) = \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^p a_i} \prod_{i=1}^p f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}), \qquad (4.13)$$

where

$$\begin{aligned} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) =&\pi^{-\frac{n}{2}} \tau^{\frac{a_i-n}{2}} \frac{\Gamma\left(\frac{\delta+a_i+n}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} \left|\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} \\ &\times \left[1 + \frac{1}{\tau}\boldsymbol{x}_i^T \left\{J_2 - J_2 \boldsymbol{x}_{P_i} \left(\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T J_2\right\} \boldsymbol{x}_i\right]^{-\left(\frac{\delta+a_i+n}{2}\right)}, \end{aligned}$$

and, when $P_i = \emptyset$

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) =f(\boldsymbol{x}_i) = \pi^{-\frac{n}{2}} \tau^{-\frac{n}{2}} \frac{\Gamma\left(\frac{\delta+n}{2}\right)}{\Gamma\left(\frac{\delta}{2}\right)} \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} \left(1 + \frac{1}{\tau}\boldsymbol{x}_i^T J_2 \boldsymbol{x}_i\right)^{-\left(\frac{\delta+n}{2}\right)}.$$

4.2.3 Assuming $\phi_i^{\frac{1}{2}} \sim$ Uniform $(0, \kappa)$: Derivation of S_3

In [36], Gelman discusses possible choices of prior distributions for variance parameters and suggests a uniform prior on the square root of the variance. Assuming such a prior distribution on $\phi_i^{1/2}$, the full set of prior distributions are

$$\begin{aligned}\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I),\\ \psi_i &\sim \text{Inv Gamma}\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right),\\ \boldsymbol{b}_i|\phi_i &\sim N_m(\mathbf{0}, \phi_i I), \text{ and}\\ \phi_i^{\frac{1}{2}} &\sim \text{ Uniform}(0, \kappa).\end{aligned}$$

This implies that $f(\phi_i) = \frac{1}{2\kappa}\phi_i^{-\frac{1}{2}}, \ \phi_i \in (0, \kappa^2)$, so that the joint distribution is

$$\begin{aligned}&f(\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i|\boldsymbol{x}_{P_i})\\ &\quad=(2\pi)^{-\left(\frac{n+a_i+m}{2}\right)}(2\kappa)^{-1}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\phi_i^{-\left(\frac{m+1}{2}\right)}\psi_i^{-\left(\frac{n+2a_i+\delta}{2}+1\right)}\\ &\quad\quad\times\exp\left\{-\frac{\tau}{2\psi_i} - \frac{1}{2\phi_i}\boldsymbol{b}_i^T\boldsymbol{b}_i - \frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i - \frac{1}{2\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)\right\}.\end{aligned}$$

After integrating over $\boldsymbol{b}_i$ and $\boldsymbol{\gamma}_i$,

$$\begin{aligned}&f(\boldsymbol{x}_i, \psi_i, \phi_i|\boldsymbol{x}_{P_i})\\ &\quad=\frac{(2\pi)^{-\frac{n}{2}}}{2\kappa}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\left|I + \frac{\phi_i}{\psi_i}Q^TQ\right|^{-\frac{1}{2}}\left|\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\phi_i^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\ &\quad\quad\times\exp\left[-\frac{\tau}{2\psi_i} - \frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_1^{\psi_i,\phi_i} - J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T J_1^{\psi_i,\phi_i}\right\}\boldsymbol{x}_i\right],\end{aligned} \tag{4.14}$$

where

$$J_1^{\psi_i,\phi_i} = I - Q\left(\frac{\psi_i}{\phi_i}I + Q^TQ\right)^{-1}Q^T. \tag{4.15}$$

Again, due to the appearance of ψ_i and ϕ_i in $J_1^{\psi_i,\phi_i}$, evaluation of $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ requires numerical integration over both ψ_i and ϕ_i. Gauss Laguerre quadrature is used over ψ_i, and Gauss Legendre quadrature is used to marginalise over the variance of the random effects.

Since Gauss Legendre quadrature performs numerical integration over the range $(-1, 1)$, the following change of variables is required:

$$\phi_i = \frac{\kappa^2}{2}(1+t), \;\; t \in (-1, 1).$$

This implies that

$$\begin{aligned}
f(\boldsymbol{x}_i,&\psi_i, t|\boldsymbol{x}_{P_i}) \\
=&\frac{\kappa^2}{2} f\left(\boldsymbol{x}_i, \psi_i, \frac{\kappa^2}{2}(1+t)|\boldsymbol{x}_{P_i}\right) \\
=&2^{-\frac{3}{2}}(2\pi)^{-\frac{n}{2}} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \left| I + \frac{\kappa^2(1+t)}{2\psi_i} Q^T Q \right|^{-\frac{1}{2}} \left|\tau I + \boldsymbol{x}_{P_i}^T J_*^{\psi_i,t} \boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} (1+t)^{-\frac{1}{2}} \\
&\times \psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)} \exp\left[-\frac{\tau}{2\psi_i} - \frac{1}{2\psi_i}\boldsymbol{x}_i^T \left\{ J_*^{\psi_i,t} - J_*^{\psi_i,t}\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T J_*^{\psi_i,t}\boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T J_*^{\psi_i,t} \right\} \boldsymbol{x}_i \right],
\end{aligned}$$

where

$$J_*^{\psi_i,t} = I - Q\left(\frac{2\psi_i}{\kappa^2(1+t)} I + Q^T Q\right)^{-1} Q^T. \tag{4.16}$$

Hence, if $w_1, w_2, \ldots, w_{q_1}$ are the weights and $u_1, u_2, \ldots, u_{q_1}$ the abscissae for q_1-point Gauss Laguerre quadrature, and $v_1, v_2, \ldots, v_{q_2}$ the weights and $t_1, t_2, \ldots, t_{q_2}$ the abscissae for q_2-point Gauss Legendre quadrature, then

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) \approx \sum_{s=1}^{q_1}\sum_{h=1}^{q_2} w_s v_h e^{u_s} f(\boldsymbol{x}_i, u_s, t_h|\boldsymbol{x}_{P_i}). \tag{4.17}$$

When gene i has no parents,

$$\begin{aligned}
f(\boldsymbol{x}_i, \psi_i, t) =&2^{-\frac{3}{2}}(2\pi)^{-\frac{n}{2}} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta}{2}}}{\Gamma\left(\frac{\delta}{2}\right)} \left| I + \frac{\kappa^2(1+t)}{2\psi_i} Q^T Q \right|^{-\frac{1}{2}} (1+t)^{-\frac{1}{2}} \\
&\times \psi_i^{-\left(\frac{n+\delta}{2}+1\right)} \exp\left(-\frac{\tau}{2\psi_i} - \frac{1}{2\psi_i}\boldsymbol{x}_i^T J_*^{\psi_i,t}\boldsymbol{x}_i\right),
\end{aligned} \tag{4.18}$$

so $f(\boldsymbol{x}_i)$ is obtained via numerical integration, as $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ was in Equation (4.17):

$$f(\boldsymbol{x}_i) \approx \sum_{s=1}^{q_1}\sum_{h=1}^{q_2} w_s v_h e^{u_s} f(\boldsymbol{x}_i, u_s, t_h).$$

This assumption gives rise to the following score metric:

$$S_3(B|d) = \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^{p} a_i} \prod_{i=1}^{p} \left\{ \sum_{s=1}^{q_1}\sum_{h=1}^{q_2} w_s v_h e^{u_s} f(\boldsymbol{x}_i, u_s, t_h|\boldsymbol{x}_{P_i}) \right\}, \tag{4.19}$$

where $f(\boldsymbol{x}_i, u_s, t_h|\boldsymbol{x}_{P_i}) = f(\boldsymbol{x}_i, u_s, t_h)$ when gene i has no parents.

4.2.4 Assuming $\phi_i \sim$ Inverse Gamma (α, β): Derivation of S_4

A typical choice for a non-informative prior distribution on a variance parameter is the Inverse-Gamma (α, β) distribution, with α and β small. Under this assumption, the prior distributions are

$$\begin{aligned}\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I),\\ \psi_i &\sim \text{Inv Gamma}\left(\frac{\delta+a_i}{2}, \frac{\tau}{2}\right),\\ \boldsymbol{b}_i|\phi_i &\sim N_m(\mathbf{0}, \phi_i I), \text{ and}\\ \phi_i &\sim \text{Inv Gamma}(\alpha, \beta),\end{aligned}$$

where α and β are positive constants.

The joint distribution is then

$$\begin{aligned}&f(\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i|\boldsymbol{x}_{P_i})\\ &=(2\pi)^{-\left(\frac{n+a_i+m}{2}\right)} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \frac{\beta^\alpha}{\Gamma(\alpha)} \phi_i^{-\left(\alpha+\frac{m}{2}+1\right)} \psi_i^{-\left(\frac{n+2a_i+\delta}{2}+1\right)} \exp\left\{-\frac{\tau}{2\psi_i} - \frac{\beta}{\phi_i}\right\}\\ &\quad\times \exp\left\{-\frac{1}{2\phi_i}\boldsymbol{b}_i^T\boldsymbol{b}_i - \frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i - \frac{1}{2\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)\right\}.\end{aligned}$$

After integrating over $\boldsymbol{b}_i$ and $\boldsymbol{\gamma}_i$,

$$\begin{aligned}&f(\boldsymbol{x}_i, \psi_i, \phi_i|\boldsymbol{x}_{P_i})\\ &=(2\pi)^{-\frac{n}{2}} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \frac{\beta^\alpha}{\Gamma(\alpha)} \left|I + \frac{\phi_i}{\psi_i}Q^TQ\right|^{-\frac{1}{2}} \left|\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} \phi_i^{-(\alpha+1)} \psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\ &\quad\times \exp\left[-\frac{\beta}{\phi_i} - \frac{\tau}{2\psi_i} - \frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_1^{\psi_i,\phi_i} - J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T J_1^{\psi_i,\phi_i}\right\}\boldsymbol{x}_i\right],\end{aligned} \tag{4.20}$$

where $J_1^{\psi_i,\phi_i}$ is as given in (4.15).

The appearance of ψ_i and ϕ_i in $J_1^{\psi_i,\phi_i}$ means that, as in the case of S_1, numerical integration is required to evaluate S_3. Two-dimensional Gauss Laguerre quadrature is then used to marginalise over ψ_i and ϕ_i. If $w_1, w_2, \ldots, w_q$ are the weights and $u_1, u_2, \ldots, u_q$ the abscissae for q-point Gauss Laguerre quadrature, then

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) \approx \sum_{s=1}^{q}\sum_{h=1}^{q} w_s w_h e^{u_s} e^{u_h} f(\boldsymbol{x}_i, u_s, u_h|\boldsymbol{x}_{P_i}). \tag{4.21}$$

When gene i has no parents,

$$\begin{aligned} f(\boldsymbol{x}_i, \psi_i, \phi_i) =& (2\pi)^{-\frac{n}{2}} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta}{2}}}{\Gamma\left(\frac{\delta}{2}\right)} \frac{\beta^{\alpha}}{\Gamma\left(\alpha\right)} \left| I + \frac{\phi_i}{\psi_i} Q^T Q \right|^{-\frac{1}{2}} \phi_i^{-(\alpha+1)} \psi_i^{-\left(\frac{n+\delta}{2}+1\right)} \\ & \times \exp\left(-\frac{\beta}{\phi_i} - \frac{\tau}{2\psi_i} - \frac{1}{2\psi_i} \boldsymbol{x}_i^T J_1^{\psi_i,\phi_i} \boldsymbol{x}_i \right). \end{aligned}$$

Hence, numerical integration is also required to obtain an estimate of $f(\boldsymbol{x}_i)$:

$$f(\boldsymbol{x}_i) \approx \sum_{s=1}^{q} \sum_{h=1}^{q} w_s w_h e^{u_s} e^{u_h} f(\boldsymbol{x}_i, u_s, u_h).$$

The score metric under this assumption is then

$$S_4(B|d) = \left(\frac{\beta}{1-\beta} \right)^{\sum_{i=1}^{p} a_i} \prod_{i=1}^{p} \left\{ \sum_{s=1}^{q} \sum_{h=1}^{q} w_s w_h e^{u_s} e^{u_h} f(\boldsymbol{x}_i, u_s, u_h | \boldsymbol{x}_{P_i}) \right\}, \tag{4.22}$$

where $f(\boldsymbol{x}_i, u_s, u_h|\boldsymbol{x}_{P_i}) = f(\boldsymbol{x}_i, u_s, u_h)$ when gene i has no parents.

4.2.5 The score metrics when ϕ_i is small relative to ψ_i

It is interesting to investigate the behaviours of the above score metrics when the variance of the random effects is small relative to the variance of the gene expression values given the random effects. That is, the behaviour of these score metrics when $\phi_i \ll \psi_i$ is of interest. When ϕ_i is small relative to ψ_i, this implies that any random effects that may be present do not contribute much to the variability of gene expression values. Note that if $\phi_i = 0$, $\forall i$, then the samples are again independent and identically distributed, and may be analysed using the original score metric.

The behaviour of the score metrics developed above as $\frac{\phi_i}{\psi_i} \to 0$ is now considered.

When ϕ_i is assumed known for all i, from Equation (4.8) it can be seen that ϕ_i enters the marginal model likelihood of the i^{th} gene through the matrices

$$I + \frac{\phi_i}{\psi_i} Q^T Q \ \text{ and } \ I - Q \left(\frac{\psi_i}{\phi_i} I + Q^T Q \right)^{-1} Q^T.$$

Note that

$$I - Q \left(\frac{\psi_i}{\phi_i} I + Q^T Q \right)^{-1} Q^T = I - \frac{\phi_i}{\psi_i} Q \left(I + \frac{\phi_i}{\psi_i} Q^T Q \right)^{-1} Q^T,$$

so that as $\frac{\phi_i}{\psi_i} \to 0$

$$I + \frac{\phi_i}{\psi_i} Q^T Q \to I$$

and

$$I - Q \left(\frac{\psi_i}{\phi_i} I + Q^T Q \right)^{-1} Q^T \to I.$$

This implies that when ϕ_i is known, as $\frac{\phi_i}{\psi_i} \to 0$,

$$\begin{aligned} f(\boldsymbol{x}_i, \psi_i | \boldsymbol{x}_{P_i}) = &(2\pi)^{-\left(\frac{n}{2}\right)} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \left| \tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i} \right|^{-\frac{1}{2}} \\ &\times \psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)} \exp\left(-\frac{1}{2\psi_i} \left[\tau + \boldsymbol{x}_i^T \left\{ I_n - \boldsymbol{x}_{P_i} \left(\tau I_{a_i} + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i} \right)^{-1} \boldsymbol{x}_{P_i}^T \right\} \boldsymbol{x}_i \right] \right), \end{aligned} \tag{4.23}$$

which is identical to the distribution of $\boldsymbol{x}_i, \psi_i | \boldsymbol{x}_{P_i}$ when no random effects are present. Hence, integrating over ψ_i gives the same marginal model likelihood for gene i as was calculated when no random effects were present, given in Equation (3.17).

Under the assumption that $\phi_i = \upsilon^{-1}\psi_i$, $\frac{\phi_i}{\psi_i} \to 0$ if and only if $\upsilon^{-1} \to 0$. Examination of the marginal model likelihood of gene i under this assumption, given in Equation (4.12), shows that υ appears in the marginal model likelihood in the terms

$$\upsilon^{\frac{m}{2}} \left| \upsilon I + Q^T Q \right|^{-\frac{1}{2}} \text{ and } I - Q \left(\upsilon I + Q^T Q \right)^{-1} Q^T.$$

These terms may be written as follows:

$$\begin{aligned} \upsilon^{\frac{m}{2}} \left| \upsilon I + Q^T Q \right|^{-\frac{1}{2}} =& \left| I + \upsilon^{-1} Q^T Q \right|^{-\frac{1}{2}} \\ I - Q \left(\upsilon I + Q^T Q \right)^{-1} Q^T =& I - \upsilon^{-1} Q \left(I + \upsilon^{-1} Q^T Q \right)^{-1} Q^T, \end{aligned}$$

so that as $\upsilon^{-1} \to 0$

$$\begin{aligned} \upsilon^{\frac{m}{2}} \left| \upsilon I + Q^T Q \right|^{-\frac{1}{2}} &\to 1, \\ I - Q \left(\upsilon I + Q^T Q \right)^{-1} Q^T &\to I. \end{aligned}$$

Implying that

$$\begin{aligned} f(\boldsymbol{x}_i | \boldsymbol{x}_{P_i}) \to& \pi^{-\frac{n}{2}} \tau^{\frac{a_i - n}{2}} \frac{\Gamma\left(\frac{\delta+a_i+n}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \left| \tau I + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i} \right|^{-\frac{1}{2}} \\ &\times \left[1 + \frac{1}{\tau} \boldsymbol{x}_i^T \left\{ I - \boldsymbol{x}_{P_i} \left(\tau I + \boldsymbol{x}_{P_i}^T \boldsymbol{x}_{P_i} \right)^{-1} \boldsymbol{x}_{P_i}^T \right\} \boldsymbol{x}_i \right]^{-\left(\frac{\delta+a_i+n}{2}\right)}, \end{aligned}$$

which is the same marginal model likelihood as was obtained in the absence of random effects.

When a uniform prior distribution is placed on $\phi_i^{\frac{1}{2}}$, it was shown in Section 4.2.3 that

$$\begin{aligned}
f(\boldsymbol{x}_i,&\psi_i,\phi_i|\boldsymbol{x}_{P_i})\\
=&\frac{(2\pi)^{-\frac{n}{2}}}{2\kappa}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\left|I+\frac{\phi_i}{\psi_i}Q^TQ\right|^{-\frac{1}{2}}\left|\tau I+\boldsymbol{x}_{P_i}^TJ_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\phi_i^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\tau}{2\psi_i}-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_1^{\psi_i,\phi_i}-J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_1^{\psi_i,\phi_i}\right\}\boldsymbol{x}_i\right].
\end{aligned}$$

Hence, as $\frac{\phi_i}{\psi_i}\to 0$,

$$\begin{aligned}
f(\boldsymbol{x}_i,\psi_i,\phi_i|\boldsymbol{x}_{P_i})\to&\frac{(2\pi)^{-\frac{n}{2}}}{2\kappa}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\left|\tau I+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\phi_i^{-\frac{1}{2}}\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\tau}{2\psi_i}-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{I-\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\right\}\boldsymbol{x}_i\right].
\end{aligned}$$

Recall from Section 4.2.3 that if $\phi_i^{\frac{1}{2}}\sim$ Uniform$(0,\kappa)$, then $f(\phi_i)=\frac{1}{2\kappa}\phi_i^{-\frac{1}{2}}$. Hence, integrating over ϕ_i in the above equation gives $f(\boldsymbol{x}_i,\psi_i|\boldsymbol{x}_{P_i})$ as in Equation (4.23).

When it is assumed that $\phi_i\sim$ Inverse Gamma(α,β), from Equation (4.20),

$$\begin{aligned}
f(\boldsymbol{x}_i,&\psi_i,\phi_i|\boldsymbol{x}_{P_i})\\
=&(2\pi)^{-\frac{n}{2}}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\frac{\beta^\alpha}{\Gamma(\alpha)}\left|I+\frac{\phi_i}{\psi_i}Q^TQ\right|^{-\frac{1}{2}}\left|\tau I+\boldsymbol{x}_{P_i}^TJ_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\phi_i^{-(\alpha+1)}\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\beta}{\phi_i}-\frac{\tau}{2\psi_i}-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_1^{\psi_i,\phi_i}-J_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_1^{\psi_i,\phi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_1^{\psi_i,\phi_i}\right\}\boldsymbol{x}_i\right],
\end{aligned}$$

so that as $\frac{\phi_i}{\psi_i}\to 0$,

$$\begin{aligned}
f(\boldsymbol{x}_i,\psi_i,\phi_i|\boldsymbol{x}_{P_i})\to&(2\pi)^{-\frac{n}{2}}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\frac{\beta^\alpha}{\Gamma(\alpha)}\left|\tau I+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\phi_i^{-(\alpha+1)}\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\beta}{\phi_i}-\frac{\tau}{2\psi_i}-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{I-\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\right\}\boldsymbol{x}_i\right].
\end{aligned}$$

In this equation, terms involving ϕ_i have the form of an Inverse Gamma kernel, so integrating over ϕ_i gives $f(\boldsymbol{x}_i,\psi_i|\boldsymbol{x}_{P_i})$ as in Equation (4.23). Hence, when ϕ_i has an Inverse Gamma distribution, and ϕ_i is small relative to ψ_i, the marginal model likelihood of gene i has the same form as when no random effects are present.

Hence, for each prior assumption on ϕ_i made in Section 4.2, the marginal model likelihood of gene i as $\frac{\phi_i}{\psi_i} \to 0$ reduces to the marginal model likelihood of gene i when random effects are absent. The implication is that when the variance of the random effects is small, the Bayesian networks obtained by the High-dimensional Bayesian Covariance Selection algorithm when random effects are ignored will be similar to those obtained when random effects are included in the analysis. This is because when ϕ_i is small $\forall i$, the samples obtained can be considered to be similar to independent and identically distributed samples.

4.3 Estimation of the Joint Covariance Matrix

The score functions derived in Sections 4.2.1–4.2.3 may be incorporated in the High-dimensional Bayesian Covariance Selection algorithm, allowing the estimation of Gaussian graphical models for high-dimensional data sets that are not independent and identically distributed samples. The implementation is discussed in Section 4.6. Given a data set consisting of expression levels for p genes, the High-dimensional Bayesian Covariance Selection program will produce an ordering of the genes, and a list of edges in the graph. Without loss of generality, we will suppose that the ordering of the genes that has been produced is $1, 2, \ldots, p$.

The joint covariance matrix of all genes is of interest. However, typically, the number of genes in the data set will be very large, so the joint covariance matrix of all genes will be too large to deal with. However, covariance matrices of small subsets of genes may be dealt with easily. To estimate such structures, the structural form of all genes must be considered.

Consider the regression model for gene i, assuming the presence of m random effects, for a given sample k:

$$X_{ik} = \sum_{j \in P_i} \gamma_{ij} X_{jk} + \sum_{r=1}^{m} q_{rk} b_{ir} + \epsilon_i, \;\; \epsilon_i \sim N(0, \psi_i), \;\; \boldsymbol{b}_i \sim N_m\left(\boldsymbol{0}, \phi_i I\right). \tag{4.24}$$

The structural form of these equations may be written as

$$\boldsymbol{X}^* = \Gamma^* \boldsymbol{X}^* + Q^* \boldsymbol{b} + \boldsymbol{\epsilon}, \;\; \boldsymbol{\epsilon} \sim N_{pn}(\boldsymbol{0}, \Psi), \;\; \boldsymbol{b} \sim N_{pm}(\boldsymbol{0}, \Phi). \tag{4.25}$$

The similarity of this equation to Equation (2.5) must be noted. Note also that when the variables in $\boldsymbol{b}$ are discrete, the conditional distribution of $\boldsymbol{X}^*$ given $\boldsymbol{b}$ is the conditional

Gaussian distribution, as described by Lauritzen and Wermuth, [51]. Edwards, [26], provided a program for the estimation of conditional independence relationships for these conditional Gaussian models, and, with others in [28], developed programs for the estimation of such relationships for high-dimensional data. In the remainder of this thesis, the High-dimensional Bayesian Covariance Selection algorithm is generalised directly.

In Equation (4.25),

$$\boldsymbol{X}^* = \begin{pmatrix} \boldsymbol{X}^1 \\ \boldsymbol{X}^2 \\ \vdots \\ \boldsymbol{X}^n \end{pmatrix} = (X_{11}, X_{21}, \dots, X_{p1}, X_{12}, X_{22}, \dots, X_{p2}, \dots\dots, X_{1n}, X_{2n}, \dots, X_{pn})^T,$$

$$\Gamma^* = \begin{pmatrix}
0 & \gamma_{12} & \gamma_{13} & \cdots & \gamma_{1p} & 0 & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \gamma_{23} & \cdots & \gamma_{2p} & 0 & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & \gamma_{12} & \gamma_{13} & \cdots & \gamma_{1p} & \cdots\cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \gamma_{23} & \cdots & \gamma_{2p} & \cdots\cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & & & & & \\
\vdots & & & & & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & \gamma_{12} & \gamma_{13} & \cdots & \gamma_{1p} \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & 0 & \gamma_{23} & \cdots & \gamma_{2p} \\
\vdots & & & & & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots\cdots & 0 & 0 & 0 & \cdots & 0
\end{pmatrix}$$

$$= I_n \otimes \begin{pmatrix} 0 & \gamma_{12} & \gamma_{13} & \cdots & \gamma_{1p} \\ 0 & 0 & \gamma_{23} & \cdots & \gamma_{2p} \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$= I_n \otimes \Gamma$$

where γ_{ij} is the coefficient of gene j in the regression of gene i, and Γ is as in Equation (3.2).

$$
\begin{aligned}
Q^* =&\begin{pmatrix}
q_{11} & 0 & 0 & \cdots & 0 & q_{21} & 0 & 0 & \cdots & 0 & \cdots\cdots & q_{m1} & 0 & 0 & \cdots & 0 \\
0 & q_{11} & 0 & \cdots & 0 & 0 & q_{21} & 0 & \cdots & 0 & \cdots\cdots & 0 & q_{m1} & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & q_{11} & 0 & 0 & 0 & \cdots & q_{21} & \cdots\cdots & 0 & 0 & 0 & \cdots & q_{m1} \\
q_{12} & 0 & 0 & \cdots & 0 & q_{22} & 0 & 0 & \cdots & 0 & \cdots\cdots & q_{m2} & 0 & 0 & \cdots & 0 \\
0 & q_{12} & 0 & \cdots & 0 & 0 & q_{22} & 0 & \cdots & 0 & \cdots\cdots & 0 & q_{m2} & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & q_{12} & 0 & 0 & 0 & \cdots & q_{22} & \cdots\cdots & 0 & 0 & 0 & \cdots & q_{m2} \\
\vdots & & & & & & & & & & & & & & & \\
\vdots & & & & & & & & & & & & & & & \\
q_{1n} & 0 & 0 & \cdots & 0 & q_{2n} & 0 & 0 & \cdots & 0 & \cdots\cdots & q_{mn} & 0 & 0 & \cdots & 0 \\
0 & q_{1n} & 0 & \cdots & 0 & 0 & q_{2n} & 0 & \cdots & 0 & \cdots\cdots & 0 & q_{mn} & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & & & & & \\
0 & 0 & 0 & \cdots & q_{1n} & 0 & 0 & 0 & \cdots & q_{2n} & \cdots\cdots & 0 & 0 & 0 & \cdots & q_{mn}
\end{pmatrix} \\
=&\begin{pmatrix}
q_{11} & q_{21} & \cdots & q_{m1} \\
q_{12} & q_{22} & \cdots & q_{m2} \\
\vdots & \vdots & & \vdots \\
q_{1n} & q_{2n} & \cdots & q_{mn}
\end{pmatrix} \otimes I_p \\
=&Q \otimes I_p
\end{aligned}
$$

where Q is the same matrix as given in Equation (4.4), and

$$
\begin{aligned}
\boldsymbol{b} =&\left(b_{11}, b_{21}, \ldots, b_{p1}, b_{21}, b_{22}, \ldots, b_{2p}, \ldots\ldots, b_{1m}, b_{2m}, \ldots, b_{pm}\right)^T \\
\Phi =&I_m \otimes diag\left(\phi_1, \phi_2, \ldots, \phi_p\right) \\
\boldsymbol{\epsilon} =&\left(\epsilon_1, \epsilon_2, \ldots, \epsilon_p, \epsilon_1, \epsilon_2, \ldots, \epsilon_p, \ldots\ldots, \epsilon_1, \epsilon_2, \ldots, \epsilon_p\right)^T \\
\Psi =&I_n \otimes diag\left(\psi_1, \psi_2, \ldots, \psi_p\right).
\end{aligned}
$$

When $\boldsymbol{b}_i \sim N_m\left(\mathbf{0}, \upsilon^{-1}\psi_i I\right)$

$$
\Phi =\upsilon^{-1} I_m \otimes diag\left(\psi_1, \psi_2, \ldots, \psi_p\right).
$$

Equation (4.25) may be rearranged to obtain

$$
\boldsymbol{X}^* = \left(I - \Gamma^*\right)^{-1}\left(Q^*\boldsymbol{b} + \boldsymbol{\epsilon}\right),
$$

which implies that

$$\begin{aligned}
E\left[\boldsymbol{X}^*\right] =& \left(I-\Gamma^*\right)^{-1} E\left[Q^*\boldsymbol{b}+\boldsymbol{\epsilon}\right] \\
=& \left(I-\Gamma^*\right)^{-1}\left(Q^* E\left[\boldsymbol{b}\right]+E\left[\boldsymbol{\epsilon}\right]\right) \\
=& \mathbf{0} \\
\operatorname{var}\left(\boldsymbol{X}^*\right) =& \left(I-\Gamma^*\right)^{-1} \operatorname{var}\left(Q^*\boldsymbol{b}+\boldsymbol{\epsilon}\right)\left(I-\Gamma^*\right)^{-T} \\
=& \left(I-\Gamma^*\right)^{-1}\left\{\operatorname{var}\left(Q^*\boldsymbol{b}\right)+\operatorname{var}\left(\boldsymbol{\epsilon}\right)\right\}\left(I-\Gamma^*\right)^{-T} \\
=& \left(I-\Gamma^*\right)^{-1}\left(Q^*\Phi Q^{*T}+\Psi\right)\left(I-\Gamma^*\right)^{-T}.
\end{aligned}$$

Hence,

$$\boldsymbol{X}^* \sim N_{pn}\left(\mathbf{0},\Sigma^*\right),$$

with

$$\Sigma^* = \left(I-\Gamma^*\right)^{-1}\left(Q^*\Phi Q^{*T}+\Psi\right)\left(I-\Gamma^*\right)^{-T}, \tag{4.26}$$

which may be estimated, given estimates of Γ^*, Φ and Ψ.

This joint covariance matrix may also be decomposed as

$$\begin{aligned}
\Sigma^* = \operatorname{var}\left(\boldsymbol{X}^*\right) &= E\left[\operatorname{var}\left(\boldsymbol{X}^*|\boldsymbol{b}\right)\right]+\operatorname{var}\left(E\left[\boldsymbol{X}^*|\boldsymbol{b}\right]\right) \\
&= E\left[\left(I-\Gamma^*\right)^{-1}\Psi\left(I-\Gamma^*\right)^{-T}\right]+\operatorname{var}\left(\left(I-\Gamma^*\right)^{-1}Q^*\boldsymbol{b}\right) \\
&= \left(I-\Gamma^*\right)^{-1}\Psi\left(I-\Gamma^*\right)^{-T}+\left(I-\Gamma^*\right)^{-1}Q^*\Phi Q^{*T}\left(I-\Gamma^*\right)^{-T}.
\end{aligned}$$

Note that the first component of this variance matrix is of the same form as the variance matrix obtained when no random effects are present. The second component contains the variability of the gene expression values that can be explained by the presence of random effects, and also encodes the dependence between samples.

It can be seen that when the variances of the random effects are small, the contribution of this second component to the joint covariance matrix will be negligible. Hence, when random effects do not vary much, ignoring the presence of random effects will not affect the joint covariance matrix to a great degree. Conversely, when the variance of the random effects is large, neglecting to include such effects in the model will result in underestimation of the elements of the matrix.

The assumption of sparsity in the original formulation of High-dimensional Bayesian Covariance Selection carries over to the random effects formulation. Sparsity in the original formulation implies that Γ will be a sparse upper-triangular matrix, with zeros on the diagonal. In the random effects formulation, the sparsity assumption again implies that Γ is a sparse upper-triangular matrix, with zeros on the diagonal. This, in turn, implies that Γ^* is a sparse upper-triangular matrix, with zeros on the diagonal. Hence, the lower triangular matrix

$$L = (I - \Gamma^*)^T \left(Q^* \Phi Q^{*T} + \Psi\right)^{-1/2} \tag{4.27}$$

will be sparse, and the concentration matrix, $\Omega^* = \Sigma^{*-1}$, has a Cholesky decomposition

$$\Omega^* = LL^T.$$

Hence, for a given subset of genes U, Σ^*_U, the joint variance matrix of the genes in U, is given by

$$\Sigma^*_U = L_U^{-T} L_U^{-1},$$

where L_U consists of the rows in L corresponding to the variables in U.

By Equations (4.26) and (4.27), it can be seen that estimation of Σ^*_U requires estimates of γ_{ij}, ψ_i and ϕ_i, for $i \in U$ and j such that $j \in P_i \; \forall i \in U$. Schemes for obtaining such posterior estimates are discussed in Section 4.4.

4.4 Posterior Estimation of Parameters

Posterior distributions of the parameters $\boldsymbol{\gamma}_i$, ψ_i, $\boldsymbol{b}_i$ and ϕ_i allow a detailed analysis of the relationships between the expression levels of different genes and of the relationships between random effects and these expression levels. As discussed above, posterior estimates of $\boldsymbol{\gamma}_i$, ψ_i and ϕ_i, $\forall i \in U$, are required to obtain estimates of $\Sigma_{RE;U}$, the joint covariance matrix of genes of interest. Note that due to the posterior independence of the parameters associated with each gene,

$$f\left(\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \psi_i, \forall i \in U | d\right) = \prod_{i \in U} f\left(\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \psi_i | d\right),$$

so that posterior estimates of the parameters associated with each gene may be obtained separately.

When it is assumed that the random effects $\boldsymbol{b}_i$ vary as $\boldsymbol{\gamma}_i$, the posterior distributions of $\boldsymbol{\gamma}_i|\psi_i$, $\boldsymbol{b}_i|\psi_i$ and ψ_i all have closed form. When ϕ_i is assumed to be known, only the posterior distribution of $\boldsymbol{\gamma}_i|\psi_i$ has a closed form, with the posterior distributions of $\boldsymbol{b}_i$ and ψ_i requiring numerical integration. Under the remaining two assumptions, none of the posterior distributions of $\boldsymbol{\gamma}_i|\psi_i$, ψ_i, $\boldsymbol{b}_i|\phi_i$ or ϕ_i have closed forms, again requiring the use of numerical integration. Hence, excepting the case when $\boldsymbol{b}_i$ has variance proportional to ψ_i, it is cumbersome to sample from the posterior distributions that may be of the most interest. Instead, Gibbs sampling is used to obtain samples from the joint posterior distribution of $\boldsymbol{\gamma}_i$, ψ_i, $\boldsymbol{b}_i$ and ϕ_i.

To perform Gibbs sampling, the following conditional posterior distributions are required:

$$\begin{aligned}
&\boldsymbol{\gamma}_i|\boldsymbol{x}_i, \psi_i, \boldsymbol{b}_i, \phi_i, \boldsymbol{x}_{P_i} \\
&\psi_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \boldsymbol{b}_i, \phi_i, \boldsymbol{x}_{P_i} \\
&\boldsymbol{b}_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \phi_i, \boldsymbol{x}_{P_i} \\
&\phi_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i}.
\end{aligned} \tag{4.28}$$

These distributions are derived under the assumptions of known ϕ_i, $\phi_i \sim \text{Inverse Gamma}(\alpha, \beta)$, and $\phi_i^{\frac{1}{2}} \sim \text{Uniform}(0, \kappa)$. Assuming that $\boldsymbol{b}_i$ varies as $\boldsymbol{\gamma}_i$ results in closed forms for posterior distributions of interest, and as such, Gibbs sampling is not required under that assumption.

The sampling scheme given these distributions is discussed in Section 4.4.5.

4.4.1 Posteriors assuming ϕ_i known

The required conditional posterior distributions for Gibbs sampling under this assumption are:

$$\begin{aligned}
\boldsymbol{\gamma}_i|\boldsymbol{x}_i,\psi_i,\boldsymbol{b}_i,\boldsymbol{x}_{P_i} \sim& N_{a_i}\left(\left(\tau I+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\left(\boldsymbol{x}_i-Q\boldsymbol{b}_i\right),\psi_i\left(\tau I+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\right), \qquad (4.29)\\
\psi_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\boldsymbol{b}_i,\boldsymbol{x}_{P_i} \sim& \text{Inv-Gamma}\left(\alpha_{\psi_i},\beta_{\psi_i}\right); \qquad (4.30)\\
\alpha_{\psi_i} =& \frac{n+2a_i+\delta}{2},\\
\beta_{\psi_i} =& \frac{\tau}{2}+\frac{\tau}{2}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i+\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right),\\
\boldsymbol{b}_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{x}_{P_i} \sim& N_m\left(\left(\frac{\psi_i}{\phi_i}I+Q^TQ\right)^{-1}Q^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right),\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)^{-1}\right).
\end{aligned}$$

When $\boldsymbol{x}_i$ has no parents in the graph, that is, when $a_i = 0$, the required posterior distributions are

$$\begin{aligned}
\psi_i|\boldsymbol{x}_i,\boldsymbol{b}_i \sim& \text{Inv-Gamma}\left(\frac{n+\delta}{2},\frac{\tau}{2}+\frac{1}{2}\left(\boldsymbol{x}_i-Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i-Q\boldsymbol{b}_i\right)\right); \qquad (4.31)\\
\boldsymbol{b}_i|\boldsymbol{x}_i,\psi_i \sim& N_m\left(\left(\frac{\psi_i}{\phi_i}I+Q^TQ\right)^{-1}Q^T\boldsymbol{x}_i,\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)^{-1}\right).
\end{aligned}$$

4.4.2 Posteriors assuming $\boldsymbol{b}_i$ vary as $\boldsymbol{\gamma}_i$

The posterior distributions under this assumption have closed form expressions:

$$\boldsymbol{\gamma}_i|\boldsymbol{x}_i,\psi_i,\boldsymbol{x}_{P_i} \sim N_{a_i}\left(\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_i,\psi_i\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\right),$$

where J_2 is as given in Equation (4.11). Further,

$$\boldsymbol{b}_i|\boldsymbol{x}_i,\psi_i,\boldsymbol{x}_{P_i} \sim N_m\left(\left(\upsilon I+Q^TJ_2^*Q\right)^{-1}Q^TJ_2^*\boldsymbol{x}_i,\psi_i\left(\upsilon I+Q^TJ_2^*Q\right)^{-1}\right),$$

where

$$J_2^* = I-\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T,$$

and

$$\psi_i|\boldsymbol{x}_i,\boldsymbol{x}_{P_i} \sim \text{ Inv Gamma}\left(\frac{n+a_i+\delta}{2},\frac{\tau}{2}+\frac{1}{2}\boldsymbol{x}_i^T\left\{J_2-J_2\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2\right\}\boldsymbol{x}_i\right).$$

When $\boldsymbol{x}_i$ has no parents in the graph,

$$\boldsymbol{b}_i|\boldsymbol{x}_i, \psi_i \sim N_m\left(\left(\upsilon I + Q^TQ\right)^{-1} Q^T\boldsymbol{x}_i, \psi_i\left(\upsilon I + Q^TQ\right)^{-1}\right),$$

and

$$\psi_i|\boldsymbol{x}_i \sim \text{ Inv Gamma}\left(\frac{n+\delta}{2}, \frac{\tau}{2} + \frac{1}{2}\boldsymbol{x}_i^T J_2 \boldsymbol{x}_i\right).$$

Hence, Gibbs sampling is not required under this assumption, as these posterior distributions provide all of the required information about the parameters, and are easily sampled from. However, for the purposes of comparison, the conditional posterior distributions in Equation (4.28) are derived.

$$\begin{aligned}
\boldsymbol{\gamma}_i|\boldsymbol{x}_i, \psi_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i} \sim & N_{a_i}\left(\left(\tau I + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\left(\boldsymbol{x}_i - Q\boldsymbol{b}_i\right), \psi_i\left(\tau I + \boldsymbol{x}_{P_i}^T\boldsymbol{x}_{P_i}\right)^{-1}\right),\\
\psi_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i} \sim & \text{Inv-Gamma}\left(\alpha_{\psi_i}, \beta_{\psi_i}\right);\\
\alpha_{\psi_i} = & \frac{n + 2a_i + m + \delta}{2},\\
\beta_{\psi_i} = & \frac{\tau}{2} + \frac{\upsilon}{2}\boldsymbol{b}_i^T\boldsymbol{b}_i + \frac{\tau}{2}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i + \frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right),\\
\boldsymbol{b}_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{x}_{P_i} \sim & N_m\left(\left(\upsilon I + Q^TQ\right)^{-1} Q^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right), \psi_i\left(\upsilon I + Q^TQ\right)^{-1}\right).
\end{aligned}$$

4.4.3 Posteriors assuming $\phi_i^{\frac{1}{2}} \sim$ Uniform $(0, \kappa)$

Under this assumption, $f(\boldsymbol{\gamma}_i|\boldsymbol{x}_i, \psi_i, \boldsymbol{b}_i, \phi_i, \boldsymbol{x}_{P_i}) = f(\boldsymbol{\gamma}_i|\boldsymbol{x}_i, \psi_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i})$ and has the form given in (4.29). Similarly, $f(\psi_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \boldsymbol{b}_i, \phi_i, \boldsymbol{x}_{P_i}) = f(\psi_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i})$, and has the form given in (4.30). The conditional posterior distributions of the random effect parameters are

$$\boldsymbol{b}_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \phi_i, \boldsymbol{x}_{P_i} \sim N_m\left(\left(\frac{\psi_i}{\phi_i}I + Q^TQ\right)^{-1} Q^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\right), \left(\frac{1}{\phi_i}I + \frac{1}{\psi_i}Q^TQ\right)^{-1}\right), \tag{4.32}$$

$$\phi_i|\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i} \equiv \phi_i|\boldsymbol{b}_i \sim \text{Inv Gamma}\left(\frac{m-1}{2}, \frac{1}{2}\boldsymbol{b}_i^T\boldsymbol{b}_i\right). \tag{4.33}$$

Note that the variance of this distribution only exists when at least 6 random effects are present, hence, the use of S_3 is only recommended when $m \geq 6$.

When $\boldsymbol{x}_i$ has no parents, the conditional posterior for ϕ_i is unchanged, the conditional posterior for ψ_i is as given in Equation (4.31), and the conditional posterior of $\boldsymbol{b}_i$ is

$$\boldsymbol{b}_i|\boldsymbol{x}_i,\psi_i,\phi_i \sim N_m\left(\left(\frac{\psi_i}{\phi_i}I+Q^TQ\right)^{-1}Q^T\boldsymbol{x}_i,\left(\frac{1}{\phi_i}I+\frac{1}{\psi_i}Q^TQ\right)^{-1}\right). \tag{4.34}$$

4.4.4 Posteriors assuming $\phi_i \sim$ Inverse Gamma (α,β)

When assuming this form of the prior distribution of the variance of the random effects, the distributions of $\boldsymbol{\gamma}_i|\boldsymbol{x}_i,\psi_i,\boldsymbol{b}_i,\phi_i,\boldsymbol{x}_{P_i}$, $\psi_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\boldsymbol{b}_i,\phi_i,\boldsymbol{x}_{P_i}$ and $\boldsymbol{b}_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\phi_i,\boldsymbol{x}_{P_i}$ are as given under the assumption that $\phi_i^{1/2}$ has a Uniform distribution. That is, the distributions are as given by (4.29), (4.30) and (4.32). However, the posterior distribution of ϕ_i given all remaining parameters is now given by

$$\phi_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i,\boldsymbol{x}_{P_i} \equiv \phi_i|\boldsymbol{b}_i \sim \text{Inv Gamma}\left(\frac{m-2\alpha}{2},\frac{1}{2}\boldsymbol{b}_i^T\boldsymbol{b}_i+\beta\right). \tag{4.35}$$

Note that the variance of this distribution only exists when

$$\frac{m-2\alpha}{2} > 2,$$

which will only occur when $m > 4+2\alpha$. Hence, the use of an Inverse Gamma prior when there are fewer than 5 random effects give no information on how the variances of the given random effects vary, and the use of this score metric when $m < 5$ is not recommended. This is consistent with the conclusions drawn by Gelman [36].

Again, when $\boldsymbol{x}_i$ has no parents, the conditional posterior for ϕ_i is unchanged, and the conditional posteriors for ψ_i and $\boldsymbol{b}_i$ are as they were under the assumption that $\phi_i^{1/2}$ has a Uniform distribution.

4.4.5 Gibbs sampling from the joint posterior distribution

Given distributions for

$$\begin{aligned}
&\boldsymbol{\gamma}_i|\boldsymbol{x}_i,\psi_i,\boldsymbol{b}_i,\phi_i,\boldsymbol{x}_{P_i} \equiv \boldsymbol{\gamma}_i|\boldsymbol{x}_i,\psi_i,\boldsymbol{b}_i,\boldsymbol{x}_{P_i}\\
&\psi_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\boldsymbol{b}_i,\phi_i,\boldsymbol{x}_{P_i} \equiv \psi_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\boldsymbol{b}_i,\boldsymbol{x}_{P_i}\\
&\boldsymbol{b}_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\phi_i,\boldsymbol{x}_{P_i}\\
&\phi_i|\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i,\boldsymbol{x}_{P_i} \equiv \phi_i|\boldsymbol{b}_i,
\end{aligned} \tag{4.36}$$

the joint posterior distribution $f(\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i | \boldsymbol{x}_i, \boldsymbol{x}_{P_i})$ can be sampled from.

To perform such sampling, let θ^d be sample d of parameter θ, where $\theta \in \{\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i\}$. To obtain a sample of size N from $f(\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i | \boldsymbol{x}_i, \boldsymbol{x}_{P_i})$:

1. Initialize the sampling by taking arbitrary values $\boldsymbol{\gamma}_i^0$, ψ_i^0, $\boldsymbol{b}_i^0$ and ϕ_i^0.

2. For $d = 1, 2, \ldots, N$:

   (a) sample $\boldsymbol{\gamma}_i^d$ from $f(\boldsymbol{\gamma}_i | \boldsymbol{x}_i, \psi_i^{d-1}, \boldsymbol{b}_i^{d-1}, \boldsymbol{x}_{P_i})$,

   (b) sample ψ_i^d from $f(\psi_i | \boldsymbol{x}_i, \boldsymbol{\gamma}_i^d, \boldsymbol{b}_i^{d-1}, \boldsymbol{x}_{P_i})$,

   (c) sample $\boldsymbol{b}_i^d$ from $f(\boldsymbol{b}_i | \boldsymbol{x}_i, \boldsymbol{\gamma}_i^d, \psi_i^d, \phi_i^{d-1}, \boldsymbol{x}_{P_i})$,

   (d) sample ϕ_i^d from $f(\phi_i | \boldsymbol{b}_i^d)$.

If ϕ_i is assumed to be known, the sampling proceeds as above, setting $\phi_i^d = \phi_i$, $\forall d \in \{0, 1, \ldots, N\}$, and omitting step 2(d). Similarly, if $\boldsymbol{x}_i$ has no parents, the sampling proceeds by ignoring references to $\boldsymbol{\gamma}_i$, and using $f(\psi_i | \boldsymbol{x}_i, \boldsymbol{b}_i^{d-1})$ and $f(\boldsymbol{b}_i | \boldsymbol{x}_i, \psi_i^d, \phi_i^{d-1})$ in place of the distributions given in steps 2(b) and 2(c).

4.5 Discussion

In the discussion of the score metrics S_1, S_2, S_3 and S_4, derived in Section 4.2, there are two factors that must be considered. The first factor is how easily the prior distribution of the variance of the random effects may be specified, while the second factor is the speed at which the score of a particular Bayesian network may be computed given this prior specification.

The four choices for the prior distribution of the variance of the random effects, ϕ_i, considered herein are:

1. ϕ_i known,

2. $\phi_i \propto \psi_i$,

3. $\phi_i^{\frac{1}{2}} \sim \text{Uniform}(0, \kappa)$, and

4. $\phi_i \sim \text{Inv Gamma}\,(\alpha, \beta)$.

The first choice, ϕ_i known, requires the specification of p variance parameters. The specification of these parameters requires information about the random effects of each gene, which, typically, will be unavailable. However, instead of specifying p separate parameters, one variance parameter for the random effects of all genes, $\phi_i = \phi$, $\forall i \in \{1, 2, \ldots, p\}$, could be specified. While this simplifies the problem, it still requires the choice of a parameter about which there is little information, and the validity of the assumption that the random effects of each gene have the same variance may not be appropriate.

Sensitivity analysis may be used to determine appropriate values of ϕ_i, $i \in \{1, 2, \ldots, p\}$, or ϕ, which leads to the consideration of the computational speed of S_1. As shown in Equation (4.10), the calculation of S_1 requires q-point numerical integration. Hence, to determine the score for a given Bayesian network on p genes, pq evaluations of $f(\boldsymbol{x}_i, \psi_i | \boldsymbol{x}_{P_i})$ are required. Assuming the data set consists of expression levels for 1000 genes, and 20-point numerical integration is used, $20,000$ evaluations of $f(\boldsymbol{x}_i | \boldsymbol{x}_{P_i})$ will be required for the calculation of the score of one Bayesian network for the 1000 genes. Hence, for very large problems, the use of S_1 may be intractable. However, the use of S_1, when $\phi_i = \phi\ \forall i$ is investigated and compared to other score metrics in Chapter 7, and the sensitivity of this score metric to choices ϕ and quadrature size is investigated. It is somewhat paradoxical that known values of ϕ_i lead to a rather complicated expression for $f(\boldsymbol{x}_i | \boldsymbol{x}_{P_i})$.

The second choice, $\phi_i \propto \psi_i$, requires the selection of υ, which controls how the variance of the random effects of gene i is related to the variance of the regression coefficients $\boldsymbol{\gamma}_i$. Again, this information will typically be unavailable. However, since the score metric S_2, given in Equation (4.13), has a closed form, sensitivity analysis is available here, so the effect that different values of υ have on the graphical models produced by High-dimensional Bayesian Covariance Selection may be determined.

The validity of this assumption, that $\phi_i \propto \psi_i$, may be questioned. This assumption implies that the variance of the random effects is related to the variance of the regression parameters in the same way for each gene. Dependent upon the choice of υ, the assumption implies that the variance of the random effects is either always smaller, or always larger, than the variance of the regression parameters. Of course, how ϕ_i is related to ψ_i probably varies from gene to gene: for some genes, the random effects may be more variable than the regression parameters, while for other genes, the random effects may have a smaller variance than the regression parameters. In theory, a separate υ_i could be selected for each gene i; however, this creates the same problem as when ϕ_i was assumed known for all i. To circumvent this problem of specification, a hyperprior distribution could be placed upon each υ_i. However, due to the form of the distribution of $\boldsymbol{x}_i|\boldsymbol{x}_{P_i}$ as given in Equation (4.12), numerical integration over υ would then be required to evaluate the score metric S_2.

The third choice places a uniform prior distribution on $\phi_i^{\frac{1}{2}}$. This choice of prior is recommended in [36], however, it is recommended that such a prior be used only when there are 6 or more random effects of interest. From the form of the posterior distribution of $\phi_i|\boldsymbol{b}_i$, in (4.33), it can be seen that when m is small, the posterior estimates of ϕ_i will be large. In fact, when $m = 1$, this posterior distribution is improper. This prior distribution requires the specification of κ, the upper bound of $\phi_i^{\frac{1}{2}}$. The choice of κ is typically guided by context, and the scale of the data, and in the case of log-scale gene expression data, choosing $\kappa = 4$ is large enough.

S_3, the score metric defined by this prior choice, given in Equation (4.19), requires double q-point numerical integration. This implies that for a given Bayesian network B on p genes, q^2p function evaluations are required to find the score of that network. Taking $q = 20$, and supposing a Bayesian network on 1000 genes, 400,000 evaluations of $f(\boldsymbol{x}_i, \psi_i, \phi_i|\boldsymbol{x}_{P_i})$ are required to obtain the score of a single Bayesian network. Hence, for large problems, the use of S_3 is even more intractable than the use of S_1.

The final choice of prior distribution for ϕ_i is an Inverse Gamma distribution with parameters α and β. Typically, α and β will be the same for all values of i. Choosing α and β small, $\alpha = \beta = 0.001$, say, is an attempt at specifying a noninformative prior for ϕ_i.

Again, a sensitivity analysis would be desirable, to determine the reliance of the algorithm on these values. However, under this assumption, the calculation of S_4 given in Equation (4.22), like S_3, requires numerical integration over two dimensions to evaluate the score of a given Bayesian network. If both numerical integrations are taken to be over 20 points, and the data set contain gene expression data for 1000 genes, 400,000 function evaluations are required to obtain the score of one Bayesian network, which is computationally prohibitive. In any case, this choice of prior distribution was discussed by Gelman in [36], and was not recommended, as when ϕ_i is estimated to be close to zero, the resulting inferences are sensitive to the choice of α and β. Due to the large number of function evaluations required, how sensitive inferences are to these choices is difficult to determine. For these reasons, S_4 will not be considered further.

Due to the simplicity of its form, it appears that S_2 is the most useful of the four scores considered. Numerical integration is not required in the calculation of S_2, and the specification of the prior distribution of the variance of the random effects is simple, with sensitivity analysis of said specification available. However, the assumption that leads to S_2 may be questionable. It is for this reason that S_1 and S_3 will also be considered for problems with small p, and the results compared in Chapter 7.

4.6 Implementation

The score metrics S_1, S_2 and S_3 may be incorporated into any score-based method for the estimation of Bayesian networks. Note that if the score metric initially used is S_0, no additional programming is required for the calculation of the S_2 score metric, a point elaborated upon below. C++ code for the calculation of the score metrics S_1 and S_3, for use in conjunction with the High-dimensional Bayesian Covariance Selection algorithm, is provided in Appendix B.

Given a score-based method for the estimation of Bayesian networks, the use of S_1 or S_3 requires, along with the data and any method-specific inputs, values for δ, τ, ϕ when S_1 is used, κ when S_3 is used, the Q matrix containing the data relating to the random effects, and the desired quadrature size. The method will then proceed as it did when random effects were assumed to be absent. For example, When S_1 or S_3 are used in conjunction with the High-dimensional Bayesian Covariance Selection algorithm, output is produced as it is when S_0 is used: for each Bayesian network saved, the score of that network, the ordering of the genes and a list of edges in the directed acyclic graph are recorded.

If the score metric associated with the score-based method being used is S_0, the use of S_2 does not require any additional programming or the input of any information additional to that required when S_0 is used. However, further pre-processing of the data is required.

Recall from Section 4.2.2 that

$$S_2(B|d) = \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^{p} a_i} \prod_{i=1}^{p} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}),$$

where

$$\begin{aligned} f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) =& \pi^{-\frac{n}{2}} \tau^{\frac{a_i-n}{2}} \frac{\Gamma\left(\frac{\delta+a_i+n}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} \left|\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} \\ & \times \left[1 + \frac{1}{\tau} \boldsymbol{x}_i^T \left\{J_2 - J_2 \boldsymbol{x}_{P_i} \left(\tau I + \boldsymbol{x}_{P_i}^T J_2 \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T J_2\right\} \boldsymbol{x}_i\right]^{-\left(\frac{\delta+a_i+n}{2}\right)}, \end{aligned}$$

and, when $P_i = \emptyset$

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) = f(\boldsymbol{x}_i) = \pi^{-\frac{n}{2}} \tau^{-\frac{n}{2}} \frac{\Gamma\left(\frac{\delta+n}{2}\right)}{\Gamma\left(\frac{\delta}{2}\right)} \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} \left(1 + \frac{1}{\tau} \boldsymbol{x}_i^T J_2 \boldsymbol{x}_i\right)^{-\left(\frac{\delta+n}{2}\right)}.$$

Note first that $\upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}}$ occurs in every term of the product in S_2, so that S_2 may be written as

$$S_2(B|d) = \upsilon^{\frac{pm}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{p}{2}} \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^{p} a_i} \prod_{i=1}^{p} g(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}),$$

where $f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) = \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} g(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ and $f(\boldsymbol{x}_i) = \upsilon^{\frac{m}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{1}{2}} g(\boldsymbol{x}_i)$. Since $\upsilon^{\frac{pm}{2}} \left|\upsilon I + Q^T Q\right|^{-\frac{p}{2}}$ is the same for all Bayesian networks for the genes, this term may be ignored. Hence, hereafter, the score metric S_2 is given by

$$S_2(B|d) = \left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^{p} a_i} \prod_{i=1}^{p} g(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}).$$

Consider now the function $g(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$. Recall that

$$J_2 = I - Q \left(\upsilon I + Q^T Q\right)^{-1} Q^T,$$

which is a symmetric, positive-definite matrix. Hence, the Cholesky decomposition of J_2 may be obtained, and J_2 may be written as $J_2 = L^T L$ for some $n \times n$ matrix L. Hence, $g(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})$ may be written as

$$\begin{aligned} g(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}) = & \pi^{-\frac{n}{2}} \tau^{\frac{a_i-n}{2}} \frac{\Gamma\left(\frac{\delta+a_i+n}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \left|\tau I + \left(L\boldsymbol{x}_{P_i}\right)^T L\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} \\ & \times \left[1 + \frac{1}{\tau} \left(L\boldsymbol{x}_i\right)^T L\boldsymbol{x}_i - \left(L\boldsymbol{x}_i\right)^T L\boldsymbol{x}_{P_i} \left\{\tau I + \left(L\boldsymbol{x}_{P_i}\right)^T L\boldsymbol{x}_{P_i}\right\}^{-1} \left(L\boldsymbol{x}_{P_i}\right)^T L\boldsymbol{x}_i\right]^{-\left(\frac{\delta+a_i+n}{2}\right)}, \end{aligned}$$

and

$$g(\boldsymbol{x}_i) = \pi^{-\frac{n}{2}} \tau^{-\frac{n}{2}} \frac{\Gamma\left(\frac{\delta+n}{2}\right)}{\Gamma\left(\frac{\delta}{2}\right)} \left\{1 + \frac{1}{\tau}\left(L\boldsymbol{x}_i\right)^T L\boldsymbol{x}_i\right\}^{-\left(\frac{\delta+n}{2}\right)}.$$

These functions have the same form as the marginal model likelihoods obtained under the assumption of independent and identically distributed samples, given in Equations (3.17) and (3.21). Hence, to incorporate S_2 into an S_0-based method for the estimation of Bayesian networks, no extra code needs to be written, and no additional inputs are required. However, the data matrix d must be augmented, so that Ld, where $J_2 = L^T L$, is the input for the program.

Given the Bayesian networks estimated through the use of any score-based method, posterior estimates of $\boldsymbol{\gamma}_i$, ψ_i, $\boldsymbol{b}_i$ and ϕ_i may be obtained for the desired values of $i \in \{1, 2, \ldots, p\}$. These estimates may then be used in the estimation of the joint covariance matrix of genes of interest, as described in Section 4.3.

In this thesis, the score metrics S_1, S_2 and S_3 are used in conjunction with the High-dimensional Bayesian Covariance Selection algorithm, as described in Section 3.3. The performance of these score metrics when used in conjunction with this algorithm are analysed and discussed in Chapter 7.

Chapter 5

Generalisation of the Distribution of the Random Effects

The random effects formulation of High-dimensional Bayesian Covariance Selection, given by

$$
\begin{aligned}
\boldsymbol{x}_i | \boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i &\sim N_n(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q\boldsymbol{b}_i, \psi_i I), \\
\boldsymbol{\gamma}_i | \psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I), \\
\psi_i &\sim \text{Inv Gamma}\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right), \\
\boldsymbol{b}_i | \phi_i &\sim N_m(\mathbf{0}, \phi_i I), \\
\phi_i = \upsilon^{-1}\psi_i \text{ or } \phi_i^{\frac{1}{2}} &\sim \text{ Uniform}(0, \kappa),
\end{aligned}
\tag{5.1}
$$

assumes that the random effects are independent and identically distributed for each gene, with some variance ϕ_i. This covariance structure is not particularly satisfactory, as it does not allow for dependence amongst the random effects of a particular gene, nor does it allow the random effects for each gene to have unequal variance parameters. Such modifications to the prior distribution of the random effects are explored in Section 5.1. In Section 5.2, an improper prior for the random effects is proposed.

5.1 Exploring the Covariance Structure of the Random Effects

In the development of the random effects formulation of High-dimensional Bayesian Covariance Selection, it was assumed that the random effects, b_{ik}, $k \in \{1, 2, \ldots, m\}$, of gene i were independent and identically distributed, for all values of i. This assumption is questionable, as random effects may have different variances, and may not be independent.

5.1.1 Assuming $\boldsymbol{b}_i|\phi_i \sim N_m(\mathbf{0}, \phi_i V)$, V known

As a first step in the generalisation of the covariance structure of the random effects, suppose there is some known symmetric positive-definite $m \times m$ matrix V such that

$$\boldsymbol{b}_i|\phi_i \sim N_m(\mathbf{0}, \phi_i V).$$

When the random effects are assumed to vary as the regression coefficients, this prior distribution becomes

$$\boldsymbol{b}_i|\psi_i \sim N_m(\mathbf{0}, \psi_i V).$$

Note that in this case, the υ that was present in the random effects formulation, given in Equation (5.1), has been subsumed into the V matrix.

The modified score metric under the assumption that $\phi_i^{\frac{1}{2}} \sim \text{Uniform}(0, \kappa)$ is then

$$S_3^V(B|d) = \prod_{i=1}^{p} \left(\frac{\beta}{1-\beta}\right)^{a_i} \sum_{s=1}^{q_1} \sum_{h=1}^{q_2} w_s v_h e^{u_s} f(\boldsymbol{x}_i, u_s, t_h|\boldsymbol{x}_{P_i}), \tag{5.2}$$

where

$$\begin{aligned}
f(\boldsymbol{x}_i,&\psi_i, t|\boldsymbol{x}_{P_i}) \\
=&2^{-\frac{3}{2}}(2\pi)^{-\frac{n}{2}} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \left|V + \frac{\kappa^2(1+t)}{2\psi_i} Q^T Q\right|^{-\frac{1}{2}} \left|\tau I + \boldsymbol{x}_{P_i}^T J_V^{\psi_i,t} \boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}} \frac{\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}}{(1+t)^{\frac{1}{2}}} \\
&\times \exp\left(-\frac{1}{2\psi_i}\left[\tau + \boldsymbol{x}_i^T \left\{J_V^{\psi_i,t} - J_V^{\psi_i,t} \boldsymbol{x}_{P_i} \left(\tau I + \boldsymbol{x}_{P_i}^T J_V^{\psi_i,t} \boldsymbol{x}_{P_i}\right)^{-1} \boldsymbol{x}_{P_i}^T J_V^{\psi_i,t}\right\} \boldsymbol{x}_i\right]\right), \\
J_V^{\psi_i,t} &= I - Q\left(\frac{2\psi_i}{\kappa^2(1+t)} V + Q^T Q\right)^{-1} Q^T,
\end{aligned}$$

and when gene i has no parents,

$$\begin{aligned} f(\boldsymbol{x}_i,\psi_i,t) =& 2^{-\frac{3}{2}}(2\pi)^{-\frac{n}{2}}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta}{2}}}{\Gamma\left(\frac{\delta}{2}\right)}\left|V+\frac{\kappa^2(1+t)}{2\psi_i}Q^TQ\right|^{-\frac{1}{2}}(1+t)^{-\frac{1}{2}} \\ & \times \psi_i^{-\left(\frac{n+\delta}{2}+1\right)}\exp\left(-\frac{\tau}{2\psi_i}-\frac{1}{2\psi_i}\boldsymbol{x}_i^TJ_V^{\psi_i,t}\boldsymbol{x}_i\right), \end{aligned}$$

Similarly, the modified score metric under the assumption that the random effects vary as the regression coefficients is given by

$$S_2^V(B|d)=\left(\frac{\beta}{1-\beta}\right)^{\sum_{i=1}^p a_i}\prod_{i=1}^p f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i}),$$

where $\boldsymbol{x}_i|\boldsymbol{x}_{P_i}$ has a multivariate t-distribution, $\boldsymbol{x}_i|\boldsymbol{x}_{P_i}\sim t_{\delta+a_i}\left(\mathbf{0},\Sigma^V_{\boldsymbol{x}_i|\boldsymbol{x}_{P_i}}\right)$,

$$\begin{aligned} \Sigma^V_{\boldsymbol{x}_i|\boldsymbol{x}_{P_i}} =& \frac{\tau}{\delta+a_i}\left\{J_2^V-J_2^V\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_2^V\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_2^V\right\}^{-1}, \\ J_2^V =& I-Q\left(V+Q^TQ\right)^{-1}Q^T, \end{aligned}$$

and when gene i has no parents, $\boldsymbol{x}_i\sim t_{\delta+a_i}\left(\mathbf{0},\Sigma^V_{\boldsymbol{x}_i}\right)$,

$$\Sigma^V_{\boldsymbol{x}_i}=\frac{\tau}{\delta}J_2^{V-1}=\frac{\tau}{\delta}\left\{I-Q\left(V+Q^TQ\right)^{-1}Q^T\right\}^{-1}.$$

Hence, when the matrix V is known, it may be included in the analysis at no extra computational cost. The inclusion of such a matrix will not be considered further.

5.1.2 A different variance parameter for each random effect

Instead of assuming the existence of a known positive definite matrix V controlling the relationships of the random effects, it could be assumed that each random effect has its own variance parameter. That is, the prior distribution on the random effects becomes

$$\boldsymbol{b}_i|\phi_i\sim N_m\left(\mathbf{0},\Phi_i=diag(\phi_{i1},\phi_{i2},\ldots,\phi_{im})\right).$$

This assumption cannot apply when it is assumed that the random effects have variance proportional to that of the regression coefficients.

If it is assumed that $\phi_{ik}^{\frac{1}{2}}\sim$ Uniform$(0,\kappa)$, $\forall i\in\{1,2,\ldots,p\}$, $\forall k\in\{1,2,\ldots,m\}$, then the marginal model likelihood for each gene i can be calculated:

$$f(\boldsymbol{x}_i|\boldsymbol{x}_{P_i})=\int f(\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i,\phi_{i1},\phi_{i2},\ldots,\phi_{im}|\boldsymbol{x}_{P_i})d\boldsymbol{\gamma}_id\psi_id\boldsymbol{b}_id\phi_{i1}d\phi_{i2}\cdots d\phi_{im},$$

$$
\begin{aligned}
f(&\boldsymbol{x}_i,\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i,\phi_{i1},\phi_{i2},\ldots,\phi_{im}|\boldsymbol{x}_{P_i})\\
=&(2\pi)^{-\left(\frac{n+a_i+m}{2}\right)}(2\kappa)^{-m}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\psi_i^{-\left(\frac{n+2a_i+\delta}{2}+1\right)}\left(\prod_{k=1}^{m}\phi_{ik}^{-1}\right)\\
&\times\exp\left\{-\frac{\tau}{2\psi_i}-\frac{1}{2\phi_i}\boldsymbol{b}_i^T\Phi_i\boldsymbol{b}_i-\frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i-\frac{1}{2\psi_i}\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i-\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i-Q\boldsymbol{b}_i\right)\right\}.
\end{aligned}
$$

After integrating over $\boldsymbol{b}_i$ and $\boldsymbol{\gamma}_i$, the joint distribution of the data and the variance parameters for gene i is

$$
\begin{aligned}
f(&\boldsymbol{x}_i,\psi_i,\phi_{i1},\phi_{i2},\ldots,\phi_{im}|\boldsymbol{x}_{P_i})\\
=&(2\pi)^{-\frac{n}{2}}(2\kappa)^{-m}\frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)}\tau^{\frac{a_i}{2}}\left|\Phi_i+\frac{1}{\psi_i}Q^TQ\right|^{-\frac{1}{2}}\left|\tau I+\boldsymbol{x}_{P_i}^TJ_{\Phi_i}\boldsymbol{x}_{P_i}\right|^{-\frac{1}{2}}\left(\prod_{k=1}^{m}\phi_{ik}^{-1}\right)\psi_i^{-\left(\frac{n+a_i+\delta}{2}+1\right)}\\
&\times\exp\left[-\frac{\tau}{2\psi_i}-\frac{1}{2\psi_i}\boldsymbol{x}_i^T\left\{J_{\Phi_i}-J_{\Phi_i}\boldsymbol{x}_{P_i}\left(\tau I+\boldsymbol{x}_{P_i}^TJ_{\Phi_i}\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^TJ_{\Phi_i}\right\}\boldsymbol{x}_i\right], \qquad (5.3)
\end{aligned}
$$

where

$$
J_{\Phi_i}=I-Q\left(\psi_i\Phi_i+Q^TQ\right)^{-1}Q^T.
$$

Numerical integration over $m+1$ dimensions is then required to obtain the score metric under this assumption. Such numerical integration is not feasible; given data on p genes, and q-point numerical integration over each of $\phi_{i1},\phi_{i2},\ldots,\phi_{im}$ and ψ_i, pq^{m+1} evaluations of Equation (5.3) would be required to obtain the score of one Bayesian network. More concretely, supposing 1000 genes, 2 random effects and 10-point numerical integration over each variance parameter, one million evaluations of Equation (5.3) would be required to obtain the score of a single Bayesian network. Hence, while the assumption of a separate variance parameter for each random effect may be more realistic than the assumption of independent and identically distributed random effects, such an assumption is impracticable.

Similarly, assuming a wholly unknown random effects covariance matrix results in marginal model likelihoods requiring $\frac{m^2+m+2}{2}$-dimensional numerical integration; which is again infeasible.

5.2 An Uninformative Random Effects Prior

Often, there will be very little prior information about the random effects. Hence, it is tempting to place an improper prior on the random effects,

$$f(\boldsymbol{b}_i) \propto 1,$$

and derive the score metric induced by such an assumption. Note that such a prior distribution on the random effects satisfies the requirements on prior distributions for the parameters of graphical models presented in [35].

Assuming that $f(\boldsymbol{b}_i) \propto 1$ implies that

$$\begin{aligned} f(\boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i | \boldsymbol{x}_{P_i}) \propto & (2\pi)^{-\left(\frac{n+a_i}{2}\right)} \frac{\left(\frac{\tau}{2}\right)^{\frac{\delta+a_i}{2}}}{\Gamma\left(\frac{\delta+a_i}{2}\right)} \tau^{\frac{a_i}{2}} \psi_i^{-\left(\frac{n+2a_i+\delta}{2}+1\right)} \\ & \times \exp\left\{-\frac{\tau}{2\psi_i} - \frac{\tau}{2\psi_i}\boldsymbol{\gamma}_i^T\boldsymbol{\gamma}_i - \frac{1}{2\psi_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)^T\left(\boldsymbol{x}_i - \boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i - Q\boldsymbol{b}_i\right)\right\}. \end{aligned}$$

Given the Normal-Inverse Gamma priors on the regression parameters $\boldsymbol{\gamma}_i$ and ψ_i, the marginal model likelihood then has the form of a multivariate t-distribution:

$$\boldsymbol{x}_i | \boldsymbol{x}_{P_i} \sim t_{a_i+\delta-m}\left(\mathbf{0}, \frac{\tau}{a_i+\delta-m}\left\{\hat{H}_Q - \hat{H}_Q\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T\hat{H}_Q\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T\hat{H}_Q\right\}^{-1}\right), \quad (5.4)$$

where

$$\hat{H}_Q = I - Q\left(Q^TQ\right)^{-1}Q^T.$$

Note that this distribution is improper for $a_i + \delta - m \leq 0$, implying that for the score of a Bayesian network to exist, $a_i + \delta > m$, $\forall i \in \{1, 2, \ldots, p\}$. All Bayesian networks contain at least one vertex with no parents, hence, δ must be selected so as to be strictly greater than m. Recall that δ, which must be positive, can be thought of as, in part, controlling the sparsity of the networks generated: the smaller the choice of δ, the sparser the generated networks. Hence, if δ must be larger than some number m, the sparsity of the estimated networks may be compromised.

When gene i has no parents,

$$\boldsymbol{x}_i \sim t_{\delta-m}\left(\mathbf{0}, \frac{\tau}{\delta-m}\left\{I - Q\left(Q^TQ\right)^{-1}Q^T\right\}^{-1}\right). \quad (5.5)$$

This distribution is improper, since $I - Q\left(Q^TQ\right)^{-1}Q^T$ is singular. In fact, $f(\boldsymbol{x}_i) = 0$ $\forall \boldsymbol{x}_i \in \mathbb{R}^n$.

This implies that even if $\delta > m$, when an improper prior distribution on the random effects is assumed, the score of any Bayesian network will be zero. Hence, the score metric induced by (5.4) and (5.5) is useless in the estimation of Bayesian networks for a given data set.

It is for this reason that an improper prior on the random effects is not recommended. However, such a prior could be approximated by assuming a normal prior distribution with a large variance for the random effects. That is, in an approximation to a noninformative prior on the random effects, the score metric S_1, given in Equation (4.10), could be used, where ϕ_i is taken to be large, $\forall i \in \{1, 2, \ldots, p\}$.

Chapter 6

Removal of Random Effects Through Analysis of Residuals

In many situations, the random effects associated with a gene expression data set may be considered to be nothing more than a nuisance, complicating the estimation of Bayesian networks for the given gene expression levels. It may be desirable to ignore the possible influences of such effects on gene expression levels, and on the relationships between genes. Of course, as discussed in Section 4.1, such an approach is not recommended. Instead, an approach which takes account of random effects that are present, without making assumptions on the form of the distribution of these random effects, is developed for use in such situations.

This approach, instead of directly using the gene expression data, is based upon the use of linear combinations of residuals left over when the data is regressed upon the random effects. This approach to the estimation of Bayesian networks for gene expression data, hereafter termed the "residual approach", was motivated by the restricted maximum likelihood procedure used in inference for mixed linear models; see for example [16, 62].

This approach is useful for a number of reasons, primarily for the reason that it makes no assumptions about the distributional form of the random effects of interest. Since no assumptions on the form of this distribution are made, the approach is correct no matter what the true distribution of the random effects may be. Another reason for the usefulness of the approach is that it results in a score metric which allows for a complex mean structure that is as fast to compute as S_0.

The drawback of the residual approach is that estimates of the random effects $\boldsymbol{b}_i$ are not admitted when this approach is taken. However, in cases when such effects are not of particular interest, but still need to be accounted for in some way, the residual approach is recommended. It is conjectured that the loss of information occurring through the use of this approach, as opposed to the use of the approaches outlined in Chapter 4, is minimal.

Consider the regression model for the expression values of gene i when random effects are included:

$$\boldsymbol{x}_i|\boldsymbol{x}_{P_i},\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i,\phi_i \sim N_n(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q\boldsymbol{b}_i,\psi_i I),\ \ \boldsymbol{b}_i|\phi_i \sim N_m\left(\mathbf{0},\phi_i I\right),$$

where ϕ_i has some distribution function, $f(\phi_i)$. The residual approach to the derivation of a score metric for use in conjunction with any score-based method for the estimation of Bayesian networks requires an $n \times (n-m)$ matrix P such that

$$\begin{aligned} &P^TQ = 0,\\ &P^TP = I_{n-m},\\ &PP^T = I_n - Q(Q^TQ)^{-1}Q^T. \end{aligned} \tag{6.1}$$

Consider the $(n-m)\times 1$ random variable $\boldsymbol{y}_i = P^T\boldsymbol{x}_i$. This random variable is a vector of linear combinations of the n gene expression values given in $\boldsymbol{x}_i$. Let $\boldsymbol{y}_{P_i} = P^T\boldsymbol{x}_{P_i}$, an $(n-m)\times a_i$ matrix whose j^{th} column is a vector of linear combinations of the expression values of the j^{th} predictor of $\boldsymbol{x}_i$. Then

$$\begin{aligned} E[\boldsymbol{y}_i] = E[P^T\boldsymbol{x}_i] &= P^TE[\boldsymbol{x}_i]\\ &= P^T\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q\boldsymbol{b}_i\right)\\ &= P^T\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + P^TQ\boldsymbol{b}_i\\ &= P^T\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i\\ &= \boldsymbol{y}_{P_i}\boldsymbol{\gamma}_i,\\ \text{var}(\boldsymbol{y}_i) = \text{var}(P^T\boldsymbol{x}_i) &= P^T\text{var}(\boldsymbol{x}_i)P\\ &= \psi_iP^TP\\ &= \psi_iI_{n-m}, \end{aligned}$$

and

$$\boldsymbol{y}_i|\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{y}_{P_i} \sim N_{n-m}(\boldsymbol{y}_{P_i}\boldsymbol{\gamma}_i, \psi_i I). \tag{6.2}$$

The form of this distribution is the same as that assumed for $\boldsymbol{x}_i|\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{x}_{P_i}$ given in Equation (3.14). Hence, the High-dimensional Bayesian Covariance Selection algorithm, or any other S_0-based algorithm, may be applied directly to the reduced data set $\{\boldsymbol{y}_i = P^T\boldsymbol{x}_i\}_{i\in\{1,2,\dots,p\}}$. Note that when the reduced data is used, the algorithm takes as input $n-m$ linear combinations of the n expression levels of each gene, instead of the n original gene expression values for each gene.

Implementation of the residual approach to the estimation of Bayesian networks is therefore simple: after selection of an appropriate matrix P and computation of $\boldsymbol{y}_i = P^T\boldsymbol{x}_i$, for all $i \in \{1, 2, \dots, p\}$, $\{\boldsymbol{y}_i = P^T\boldsymbol{x}_i\}_{i\in\{1,2,\dots,p\}}$ is used as input for the algorithm, and S_0 is used.

It will now be shown that the score metric is not dependent upon the (somewhat arbitrary) choice of P. Consider the marginal model likelihood of $\boldsymbol{y}_i$, from Equation (3.18):

$$\boldsymbol{y}_i|\boldsymbol{y}_{P_i} \sim t_{\delta+a_i}\left(\boldsymbol{0}, \frac{\tau}{\delta+a_i}\left\{I_{n-m} - \boldsymbol{y}_{P_i}\left(\tau I + \boldsymbol{y}_{P_i}^T\boldsymbol{y}_{P_i}\right)^{-1}\boldsymbol{y}_{P_i}^T\right\}^{-1}\right).$$

This implies that

$$\begin{aligned}
f(\boldsymbol{y}_i|\boldsymbol{y}_{P_i}) =& \frac{\Gamma\left(\frac{\delta+a_i+n-m}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)(\pi\tau)^{\frac{n-m}{2}}}\left|I - \boldsymbol{y}_{P_i}\left(\tau I + \boldsymbol{y}_{P_i}^T\boldsymbol{y}_{P_i}\right)^{-1}\boldsymbol{y}_{P_i}^T\right|^{\frac{1}{2}} \\
&\times\left[1 + \frac{1}{\tau}\boldsymbol{y}_i^T\left\{I - \boldsymbol{y}_{P_i}\left(\tau I + \boldsymbol{y}_{P_i}^T\boldsymbol{y}_{P_i}\right)^{-1}\boldsymbol{y}_{P_i}^T\right\}\boldsymbol{y}_i\right]^{-\left(\frac{\delta+a_i+n-m}{2}\right)} \\
=& \frac{\Gamma\left(\frac{\delta+a_i+n-m}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)(\pi\tau)^{\frac{n-m}{2}}}\left|I - P^T\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T P\right|^{\frac{1}{2}} \\
&\times\left[1 + \frac{1}{\tau}\boldsymbol{x}_i^T P\left\{I - P^T\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T P\right\}P^T\boldsymbol{x}_i\right]^{-\left(\frac{\delta+a_i+n-m}{2}\right)} \\
=& \frac{\Gamma\left(\frac{\delta+a_i+n-m}{2}\right)}{\Gamma\left(\frac{\delta+a_i}{2}\right)(\pi\tau)^{\frac{n-m}{2}}}\left|I - P^T\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T P\right|^{\frac{1}{2}} \\
&\times\left[1 + \frac{1}{\tau}\boldsymbol{x}_i^T\left\{PP^T - PP^T\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T PP^T\right\}\boldsymbol{x}_i\right]^{-\left(\frac{\delta+a_i+n-m}{2}\right)}.
\end{aligned}$$

Using the identity

$$\left|I - P^T\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T P\right|^{\frac{1}{2}} = \tau^{\frac{a_i}{2}}\left|\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\right|^{\frac{1}{2}}, \tag{6.3}$$

it can be seen that P only appears in the marginal model likelihood as $PP^T = I_n - Q(Q^TQ)^{-1}Q^T$. Similarly, from Equation (3.19), when gene i has no parents,

$$\boldsymbol{y}_i \sim t_\delta \left(\boldsymbol{0}, \frac{\tau}{\delta} I_{n-m}\right),$$

so that

$$\begin{aligned} f(\boldsymbol{y}_i) =& \frac{\Gamma\left(\frac{\delta+n-m}{2}\right)}{\Gamma\left(\frac{\delta}{2}\right)(\pi\tau)^{\frac{n-m}{2}}} \left(1 + \frac{1}{\tau}\boldsymbol{y}_i^T\boldsymbol{y}_i\right)^{-\left(\frac{\delta+n-m}{2}\right)} \\ =& \frac{\Gamma\left(\frac{\delta+n-m}{2}\right)}{\Gamma\left(\frac{\delta}{2}\right)(\pi\tau)^{\frac{n-m}{2}}} \left(1 + \frac{1}{\tau}\boldsymbol{x}_i^T PP^T \boldsymbol{x}_i\right)^{-\left(\frac{\delta+n-m}{2}\right)}. \end{aligned}$$

Hence, the score metric under this residual approach to the removal of random effects is independent of the choice of P.

The posterior distributions of the parameters $\boldsymbol{\gamma}_i$ and ψ_i are also independent of P. Equations (3.22) and (3.23) imply that

$$\begin{aligned} \boldsymbol{\gamma}_i|\boldsymbol{y}_i, \psi_i, \boldsymbol{y}_{P_i} &\sim N_{a_i}\left(\boldsymbol{\mu}_{\boldsymbol{\gamma}_i}, \Sigma_{\gamma_i}\right), \\ \boldsymbol{\mu}_{\boldsymbol{\gamma}_i} &= \left(\tau I + \boldsymbol{y}_{P_i}^T\boldsymbol{y}_{P_i}\right)^{-1}\boldsymbol{y}_{P_i}^T\boldsymbol{y}_i, \\ &= \left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_i, \\ \Sigma_{\gamma_i} =& \psi_i\left(\tau I + \boldsymbol{y}_{P_i}^T\boldsymbol{y}_{P_i}\right)^{-1}, \\ =& \psi_i\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1} \\ \psi_i|\boldsymbol{y}_i, \boldsymbol{y}_{P_i} &\sim \text{Inv Gamma}(\alpha, \beta), \\ \alpha =& \frac{\delta + a_i + n - m}{2} \\ \beta =& \frac{\tau}{2} + \frac{1}{2}\boldsymbol{y}_i^T\left\{I - \boldsymbol{y}_{P_i}\left(\tau I + \boldsymbol{y}_{P_i}^T\boldsymbol{y}_{P_i}\right)^{-1}\boldsymbol{y}_{P_i}^T\right\}\boldsymbol{y}_i, \\ =& \frac{\tau}{2} + \frac{1}{2}\boldsymbol{x}_i^T\left\{PP^T - PP^T\boldsymbol{x}_{P_i}\left(\tau I + \boldsymbol{x}_{P_i}^T PP^T\boldsymbol{x}_{P_i}\right)^{-1}\boldsymbol{x}_{P_i}^T PP^T\right\}\boldsymbol{x}_i, \end{aligned}$$

and when gene i has no parents,

$$\psi_i|\boldsymbol{y}_i \sim \text{Inverse Gamma}\left(\frac{\delta + n - m}{2}, \frac{\tau}{2} + \frac{1}{2}\boldsymbol{y}_i^T\boldsymbol{y}_i\right),$$

where $\boldsymbol{y}_i^T \boldsymbol{y}_i = \boldsymbol{x}_i^T P P^T \boldsymbol{x}_i$.

Hence, the residual approach to the removal of random effects is independent of the choice of P. In Chapter 7, the performance of the residual approach is assessed and compared to the performance of the original approach, and to the performance of the score metrics S_1, S_2 and S_3.

Chapter 7

The Use of Score Metrics That Take Account of Complex Mean Structure

In this chapter, the necessity and utility of score metrics that take account of a complex mean structure in the estimation of graphical models is demonstrated. In particular, the score metrics S_1, S_2 and S_3 and the residual approach are implemented within the High-dimensional Bayesian Covariance Selection algorithm [23, 25], and applied to example data sets with known network structure. Note that while High-dimensional Bayesian Covariance Selection is the algorithm used here, the score metrics considered could be used within any other score-based method for the estimation of Bayesian networks, such as those described in Section 3.1.1.

Recall that S_1, given in Equation (4.10), arises from placing a prior distribution with a fixed variance ϕ_i on the random effects for gene i. In our evaluation of this score metric, we make the simplifying assumption, as discussed in Section 4.5, that $\phi_i = \phi$ for all i. S_2, given in Equation (4.13), arises from the assumption that the random effects have a variance $\phi_i = \upsilon^{-1}\psi_i$, where ψ_i is the true variance of the log-scale expression levels for gene i. The third score metric considered, S_3, given in Equation (4.19), arises from the assumption that the standard deviation of the random effects for gene i, $\phi_i^{1/2}$, is uniformly distributed over the range $(0, \kappa)$. Note that S_3 is only appropriate when more than 5 random effects are being considered.

In Section 7.1, the necessity of score metrics that take account of complex mean structure is demonstrated. In particular, it is shown that in the estimation of Bayesian networks, failure to account for heterogeneity can lead to significant bias.

In Section 7.2, the effect of misspecification of the parameters associated with S_1 and S_2 on the estimation of Bayesian networks is investigated.

In Section 7.3, the consequences of misspecifying the distribution of ϕ_i on the estimation of Bayesian networks is investigated. Also considered are the consequences of such misspecification on the posterior estimation of random effects.

7.1 The Necessity of Taking Account of Complex Mean Structure

To demonstrate the need to account for complex mean structure in the estimation of Bayesian networks, S_0, given in Equation (3.20), is applied to data sets generated from known systems of linear recursive equations. First, S_0 is applied to data sets consisting of independent and identically distributed samples. S_0 is then applied to more complex data sets with a non-zero mean structure, allowing a demonstration of results typical when the presence of the mean structure is ignored. In order to compare these results to the Bayesian networks obtained when random effects are taken into account, the residual approach is applied to these same data sets.

Comparison of the results of these analyses demonstrate the necessity of score metrics that can take account of a complex mean structure, and the utility of the residual approach in the estimation of Bayesian networks for data sets with such a complex mean structure. The application of the residual approach to data sets that consist of independent and identically distributed samples demonstrates what can happen when a non-zero mean structure is wrongly assumed for the data.

The data sets analysed in this section are generated according to the systems of linear recursive equations outlined in Examples 7.1–7.6. Note that data sets generated according to Examples 7.1–7.3 consist of independent and identically distributed samples, while those data sets generated according to Examples 7.4–7.6 contain random effects. For each example, 10 data sets are generated.

Example 7.1. *For this example, ten data sets are simulated according to the following system of linear recursive equations:*

$$X_{ik} = \epsilon_{ik}, \;\; \epsilon_{ik} \sim N(0, \psi_i), \;\; i = 1, 2, \ldots, 10, k = 1, 2, \ldots, 10,$$

where

$$\Psi = (\psi_1, \psi_2, \ldots, \psi_{10}) = (0.555, 0.713, 1.38, 0.942, 3.513, 1.234, 2.256, 1.739, 0.552, 39.251)\,.$$

Note that the values of the ψ_i were obtained by sampling from an Inverse Gamma $(1, 1/2)$ distribution, and were constant for each of the samples generated according to this system of linear recursive equations.

Each of the ten data sets generated consist of ten independent and identically distributed samples of ten variables. Note that the Bayesian network associated with this system of linear recursive equations is the empty network.

Example 7.2. *This example is a larger version of Example 7.1. For this example, ten data sets are simulated according to*

$$X_{ik} = \epsilon_{ik}, \;\; \epsilon_{ik} \sim N(0, \psi_i), \;\; i = 1, 2, \ldots, 100, k = 1, 2, \ldots, 100,$$

where the parameters ψ_i, $i = 1, 2, \ldots, 100$, were again obtained by sampling from an Inverse Gamma $(1, 1/2)$ distribution, and are constant across the data sets sampled from this system of linear recursive equations.

Each of the ten data sets generated consist of 100 independent and identically distributed samples of 100 variables. As with the previous example, the Bayesian network associated with this system of linear recursive equations is the empty network.

Example 7.3. *The system of linear recursive equations governing the data sets simulated for this example is*

$$\begin{aligned}
X_{ik} &= \epsilon_{ik}, \;\; i = 1, 2, \ldots, 18 \\
X_{19,k} &= \gamma_{19,1} X_{1k} + \gamma_{19,2} X_{2k} + \epsilon_{19,k}, \\
X_{20,k} &= \gamma_{20,19} X_{19,k} + \epsilon_{20,k}, \\
\epsilon_{ik} &\sim N(0, \psi_i), \;\; i = 1, 2, \ldots, 20, \;\; k = 1, 2, \ldots, 10.
\end{aligned}$$

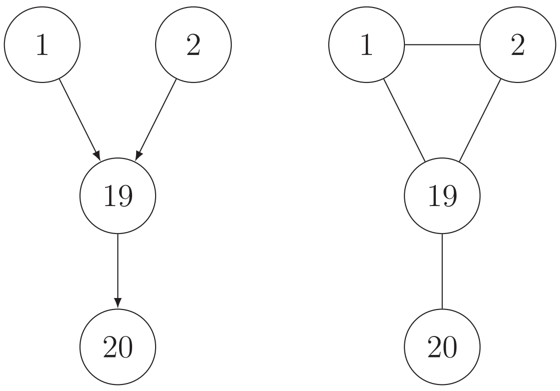


Figure 7.1: The connected component of the directed acyclic graph of the Bayesian network in Example 7.3, and the moralised version of that connected component.

The parameters ψ_i, $i = 1, 2, \dots, 20$, $\boldsymbol{\gamma}_{19} = (\gamma_{19,1}, \gamma_{19,2})^T$ and $\gamma_{20,19}$ are constant across samples, and were obtained by sampling from the following distributions:

$$\psi_i \sim \text{Inv Gamma}\left(\frac{2+a_i}{2}, \frac{1}{2}\right), \; a_i = 0, \; i = 1, 2, \dots, 18, \; a_{19} = 2, \; a_{20} = 1,$$
$$\boldsymbol{\gamma}_{19} \sim N_2\left(\mathbf{0}, \psi_{19} I_2\right), \gamma_{20,19} \sim N(0, \psi_{20}).$$

The Bayesian network associated with this system of linear recursive equations then has 3 edges: edges from vertices 1 and 2 to 19, and an edge from vertex 19 to 20. See Figure 7.1 for the connected component of the directed acyclic graph associated with this Bayesian network, and the corresponding moralised component.

Each of the ten data sets generated for this example consist of ten independent and identically distributed samples of 20 variables.

Example 7.4. *The system of linear recursive equations governing this example is the same as Example 7.1, but a non-zero mean structure corresponding to two groups is included. The system of linear recursive equations is given by*

$$X_{ijk} = b_{ij} + \epsilon_{ijk}, \; \epsilon_i \sim N(0, \psi_i), \; i = 1, 2, \dots, 10 \; j = 1, 2, \; k = 1, 2, \dots, 5 \qquad (7.1)$$

where values of the parameters ψ_i are the same as those given in Example 7.1, and $\boldsymbol{b}_i = (b_{i1}, b_{i2})^T$ are fixed across samples, obtained by sampling from:

$$b_{ij} \sim N(0, 5), \; i = 1, 2, \dots, 10, \; j = 1, 2.$$

The Bayesian network associated with this system of linear equations has no edges, and the true model for each variable may be written as

$$\boldsymbol{x}_i|\psi_i, \boldsymbol{b}_i \sim N_{10}\left(Q\boldsymbol{b}_i, \psi_i I_{10}\right), \tag{7.2}$$

where

$$\boldsymbol{x}_i = \begin{pmatrix} x_{i11} \\ x_{i12} \\ x_{i13} \\ x_{i14} \\ x_{i15} \\ x_{i21} \\ x_{i22} \\ x_{i23} \\ x_{i24} \\ x_{i25} \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix},$$

$i = 1, 2, \ldots, 10$. *Here,* $p = n = 10$ *and* $m = 2$.

Ten data sets are generated for this example.

Example 7.5. *The system of linear recursive equations governing this example is the same as Example 7.2, but a non-zero mean structure corresponding to two groups is included:*

$$X_{ijk} = b_{ij} + \epsilon_{ijk}, \ \epsilon_i \sim N(0, \psi_i), \ i = 1, 2, \ldots, 100, \ j = 1, 2, \ k = 1, 2, \ldots, 50. \tag{7.3}$$

Values of the parameters ψ_i *are the same as those for Example 7.2, and* $\boldsymbol{b}_i = (b_{i1}, b_{i2})^T$, $i = 1, 2, \ldots, 100$, *are fixed across data sets, obtained by sampling from*

$$b_{ij} \sim N(0, 5), \ i = 1, 2, \ldots, 100, \ j = 1, 2.$$

Example 7.6. *The system of linear recursive equations governing this example is the same as Example 7.3, but with a non-zero mean structure included:*

$$\begin{aligned} X_{ik} &= q_{1k}b_{i1} + q_{2k}b_{i2} + q_{3k}b_{i3} + \epsilon_{ik}, \ i = 1, 2, \ldots, 18 \\ X_{19,k} &= q_{1k}b_{19,1} + q_{2k}b_{19,2} + q_{3k}b_{19,3} + \gamma_{19,1}X_{1k} + \gamma_{19,2}X_{2k} + \epsilon_{19,k}, \\ X_{20,k} &= q_{1k}b_{20,1} + q_{2k}b_{20,2} + q_{3k}b_{20,3} + \gamma_{20,19}X_{19,k} + \epsilon_{20,k}, \\ \epsilon_{ik} &\sim N(0, \psi_i), \ i = 1, 2, \ldots, 20, \ k = 1, 2, \ldots, 10, \end{aligned}$$

where values of the parameters ψ_i, $\boldsymbol{\gamma}_{19} = (\gamma_{19,1}, \gamma_{19,2})^T$ *and* $\gamma_{20,19}$ *are the same as those obtained for the data sets generated from Example 7.3. The random effects* $\boldsymbol{b}_i = (b_{i1}, b_{i2}, b_{i3})^T$, $i = 1, 2, \ldots, 20$, *constant across the 10 data sets generated, are obtained by sampling from*

$$b_{ij} \sim N(0, 4), \;\; i = 1, 2, \ldots, 20, \;\; j = 1, 2, 3.$$

The Bayesian network associated with this system of linear recursive equations is the same as that associated with Example 7.3, shown in Figure 7.1.

The true model for each variable may be written as

$$\begin{aligned} \boldsymbol{x}_i | \psi_i, \boldsymbol{b}_i \sim & N_{10}\left(Q\boldsymbol{b}_i, \psi_i I_{10}\right), \;\; i = 1, 2, \ldots, 18 \\ \boldsymbol{x}_{19} | \boldsymbol{\gamma}_{19}, \psi_{19}, \boldsymbol{b}_{19} \sim & N_{10}\left(\boldsymbol{x}_{P_{19}}\boldsymbol{\gamma}_{19} + Q\boldsymbol{b}_{19}, \psi_{19} I_{10}\right), \\ \boldsymbol{x}_{20} | \boldsymbol{\gamma}_{20}, \psi_{20}, \boldsymbol{b}_{20} \sim & N_{10}\left(\boldsymbol{x}_{P_{20}}\gamma_{20} + Q\boldsymbol{b}_{20}, \psi_{20} I_{10}\right), \end{aligned}$$

where

$$\boldsymbol{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{i10} \end{pmatrix}, \;\; \boldsymbol{x}_{P_{19}} = \begin{pmatrix} \boldsymbol{x}_1, \boldsymbol{x}_2 \end{pmatrix}, \;\; \boldsymbol{x}_{P_{20}} = \boldsymbol{x}_{19}$$

and

$$Q = \begin{pmatrix} q_{11} & q_{21} & q_{31} \\ q_{12} & q_{22} & q_{32} \\ \vdots & \vdots & \vdots \\ q_{1,10} & q_{2,10} & q_{3,10} \end{pmatrix} = \begin{pmatrix} -1.3222537 & 0.83255032 & -1.73607746 \\ 0.2200045 & -1.36996699 & 0.54524960 \\ 0.3711498 & 0.60683906 & 0.60280101 \\ -1.5345847 & 1.51883764 & 0.82072612 \\ -0.7342315 & -0.01382055 & 0.92755141 \\ 0.9225403 & 0.86877640 & -0.09072802 \\ 1.0222735 & -0.44125852 & -0.04272580 \\ 0.2694415 & -0.59209217 & 0.11438749 \\ -0.6392365 & 0.20427984 & -0.21395314 \\ -0.1542683 & 0.47523095 & -0.12045658 \end{pmatrix}.$$

Note that in this case, elements of the Q matrix consist of random samples from the standard normal distribution, but are treated as known constants in the analysis.

7.1.1 Analysis of the data sets

For each of the six examples described above, ten data sets are generated. Bayesian networks are estimated for each of these data sets using two methods: first S_0 is used in conjunction with the High-dimensional Bayesian Covariance Selection algorithm, then the residual approach is taken in the estimation of Bayesian networks for the data sets.

7.1.2 Using S_0 in the Estimation of Bayesian Networks

The number of edges in the highest-scoring networks found when S_0 is used in the estimation of Bayesian networks for data sets generated according to Examples 7.1–7.6 are summarised in Table 7.1. Note that data sets drawn from Examples 7.1–7.3 consist of independent and identically distributed samples, while data sets drawn from Examples 7.4–7.6 contain a non-constant mean structure. Recall that implicit in the use of S_0 is the assumption of the presence of a data set that consists of independent and identically distributed samples.

Note that since we are primarily interested in the conditional independence relationships present in the estimated graphs, only the number of spurious and correct edges present in the estimated graphs is considered.

From the top half of Table 7.1, it can be seen that when S_0 is applied to the simulated data sets consisting of independent and identically distributed samples, the highest-scoring Bayesian networks obtained are, in general, reasonably close to the true networks that were used to generate the data sets. For each example, few spurious edges are included in the highest-scoring networks found. Conversely, the bottom half of this table shows that when S_0 is used to analyse data sets with a non-constant mean structure, the highest-scoring networks obtained generally contain many more edges than are present in the true network.

Comparison of the top and bottom halves of of Table 7.1 demonstrates the consequences of ignoring the presence of a complex mean structure in the estimation of Bayesian networks for data sets. These results show clearly that if S_0 is used in the analysis of a data set that has a complex mean structure, the high-scoring Bayesian networks found generally have many more spurious edges than those networks obtained when data sets consisting of independent and identically distributed samples are analysed. This is to be expected, as correlations

Table 7.1: Mean and standard deviation of the number of spurious and correct edges in the highest-scoring Bayesian networks obtained when S_0 is applied to data sets simulated according to Examples 7.1–7.6.

Example	True Edges	Spurious Edges	Correct Edges
7.1	0	0.2(0.42)	-
7.2	0	0.7(0.95)	-
7.3	3	0.8(0.63)	1.7(0.48)
7.4	0	2.2(0.63)	-
7.5	0	107.7(5.42)	-
7.6	3	5.3(1.42)	1.8(0.42)

between samples are not being correctly controlled for when the complex mean structure is ignored, and hence spurious correlations between variables are induced. Such results have been discussed by Schäfer and Bühlman [70], and by Teng *et al* [77].

Other Approaches Based on S_0

Other approaches based on the use of S_0, such as those briefly discussed in Chapter 4, could be used in the analysis of data sets that contain a complex mean structure. These approaches do not explicitly provide for the presence of a complex mean structure in the models of each of the variables, instead allowing for the presence of such effects in a less rigorous way.

The first such method involves considering the sub-samples in each data set that are independent and identically distributed, and using S_0 in the estimation of Bayesian networks given these independent and identically distributed samples. For example, the data sets generated according to Example 7.4 consist of 10 samples of each variable. The first five of these samples can be viewed as independent and identically distributed samples, as can the last five of these samples. Hence, S_0 may be used in the estimation of Bayesian networks for each half of the data sets generated according to Example 7.4. Note that this approach is also applicable to the data sets generated according to Example 7.5.

Application of this approach to the data sets generated according to Example 7.4 results in the estimation of Bayesian networks for 20 data sets consisting of 5 independent and identically distributed samples of each of 10 variables. For each of these 20 data sets, the highest-scoring network obtained is the true network, with no edges.

Table 7.2: Mean and standard deviation of the number of edges in the highest-scoring network when data sets from Example 7.5 are analysed in halves.

	Mean (sd)
First half	1.6 (0.97)
Second half	2.4 (1.78)

When this method is applied to the data sets generated according to Example 7.5, Bayesian networks are estimated for 20 data sets each consisting of 50 independent and identically distributed samples of 100 variables. The number of edges in the highest-scoring network found in each of these analyses is summarised in Table 7.2. Examination of these results shows that the highest-scoring networks found through the use of this method are quite close to the true networks.

Note that while this method appears to work quite well in the estimation of Bayesian networks for Examples 7.4 and 7.5, the method is only applicable when the data set being analysed contains some independent and identically distributed samples. Often this method will not be appropriate. For example, it is not applicable to data sets generated according to Example 7.6.

The second method considered is to include the variables that define the mean structure as vertices. If the data relating to these variables consists of samples from some normal distribution, these variables may be directly included as vertices in the estimation of a Bayesian network for the data. However, if the data relating to these variables is discrete, as it is in Examples 7.4 and 7.5, then hybrid Bayesian networks, that is, networks on both continuous and discrete random variables, need to be considered. Such networks are beyond the scope of this work, but the interested reader is referred to Lauritzen [50] for a discussion on graphical models on both continuous and discrete variables.

As the data relating to the variables that define the mean structure in Example 7.6 are in fact samples from the standard normal distribution, data generated from this example may be analysed in this way. Hence, instead of estimating a Bayesian network on 20 variables, a Bayesian network on 23 variables is estimated. The true directed acyclic graph underlying this Bayesian network will have 63 edges: an edge from each of the vertices representing the variables involved in the mean structure to each of the original vertices, and the three edges that were present in the original directed acyclic graph, as shown in Figure 7.1.

Each of the ten data sets generated according to Example 7.6 are analysed in this way, and the number of edges in the highest-scoring networks found are recorded. The number of these edges that are associated with the nodes that represent the mean structure covariates, as well as the number of edges that are present in the true network, are also recorded. The results are displayed in Table 7.3.

Table 7.3: Mean and standard deviation of the number of edges in the highest-scoring networks when covariates are included as vertices in the analysis of Example 7.6.

Edges	Mean (sd)
Total	10.5 (1.35)
Covariate	4.0 (0.67)
Correct	1.6 (0.52)
Spurious	4.9 (1.66)

The first thing to note is that the number of edges in the highest-scoring networks increase upon the inclusion of mean structure covariates as nodes in the network, as can be seen from the comparison of the first line of Table 7.3 and the last line of Table 7.1. Note also that when mean structure covariates are included as nodes, many of the edges in the networks found are connected to the nodes representing these covariates.

The connected components of the highest scoring Bayesian network obtained when the first data set is analysed in this way are shown in Figure 7.2, where the vertices corresponding to the covariates are labelled as C_i, $i = 1, 2, 3$. Note that the direction of the edges do not imply any notions of causality: the edges only provide information about conditional independence relationships between the variables. The network shown is typical of the networks found in the analysis of the other data sets. While several of the edges found in these graphs are connected to the covariates, these graphs are quite different to the true graph. As mentioned above, in the true graph there are 60 edges connected to the covariates; on average, the graphs obtained via this method have 4 edges connected to the covariates.

From the table and the figure, it can be seen that the nodes representing the random effects are the most highly connected vertices in the networks obtained. However, these nodes are not as highly connected as may be hoped, as it is known that each random effect affects each variable in the network.

A possible explanation for these results may lie in the sparsity constraint of the High-dimensional Bayesian Covariance Selection algorithm. This constraint is thought to reflect the sparse nature of genetic networks, and is useful in the estimation of large joint covariance matrices. However, in this case, the network of interest cannot be thought of as sparse, since each of the three covariate vertices is the parent of 20 vertices. Recall that the degree of sparsity of the networks found by the High-dimensional Bayesian Covariance Selection algorithm is controlled through the value of β, which is the probability of any variable having any other variable in its parent set. Hence, an increase in β may result in the estimation of a network with more edges involving the random effect vertices.

Table 7.4: Mean and standard deviation of the number of edges in the highest-scoring networks when covariates are included as vertices in the analysis of Example 7.6, $\beta = 0.9$.

Edges	Mean (sd)
Total	66.2 (4.89)
Covariate	21.7 (3.56)
Correct	2.2 (0.42)
Spurious	42.3 (7.07)

The results of increasing β to 0.9 are summarised in Table 7.4. It can be seen that, on average, the highest scoring Bayesian network found has 66.2 edges, and that 21.7 of these involve the covariates. Although more edges associated with the covariate vertices are present in the highest-scoring network, so are more spurious edges.

Another approach could be to control the sparsity of the covariate vertices separately, allow-

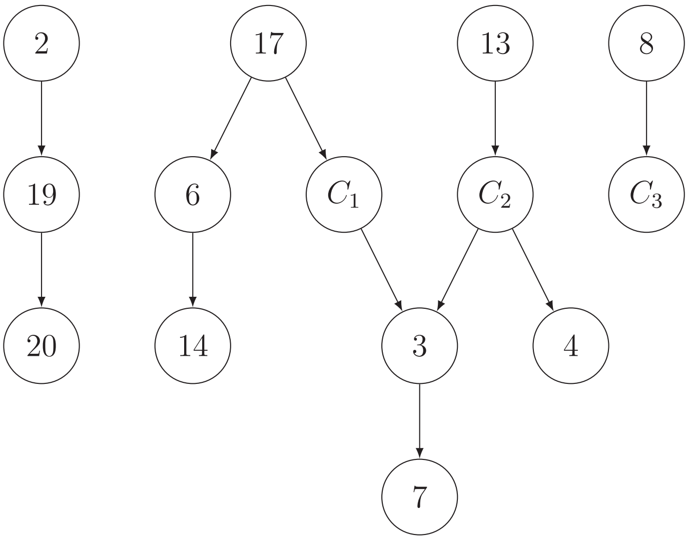


Figure 7.2: The connected components of the Bayesian network for Example 7.6, taking the covariates as vertices in the network. Vertices corresponding to these random effects are C_1, C_2 and C_3.

ing these vertices to be highly connected, while still ensuring that the rest of the structure remains quite sparse. However, this approach is beyond the scope of this thesis.

Alternatively, the possible orderings of variables could be constrained, allowing the covariates associated with the exogenous variables to be associated with more of the original variables. Again, this approach is beyond the scope of this thesis.

In the light of the preceding examples, it appears that simple approaches to account for the presence of a complex mean structure are not effective.

7.1.3 The Residual Approach to the Estimation of Bayesian Networks

Here the residual approach to the estimation of Bayesian networks is applied to the data sets simulated according to Examples 7.1–7.6. First this approach will be applied to the data sets with a complex mean structure, generated according to Examples 7.4–7.6. Then the consequences of mistakenly taking account of a complex mean structure will be explored

Table 7.5: Mean and standard deviation of the number of spurious and correct edges in the highest-scoring Bayesian networks obtained when the residual approach is applied to data sets simulated according to Examples 7.4–7.6.

Example	True Edges	Spurious Edges	Correct Edges
7.4	0	0.0(0.00)	-
7.5	0	1.3(1.64)	-
7.6	3	0.3(0.48)	1.7(0.82)

Table 7.6: Mean and standard deviation of the number of spurious and correct edges in the highest-scoring Bayesian networks obtained when the residual approach is applied to data sets simulated according to Examples 7.1–7.3.

Example	True Edges	Spurious Edges	Correct Edges
7.1	0	0.0(0.00)	-
7.2	0	0.4(0.52)	-
7.3	3	0.5(0.53)	1.4(0.52)

through the application of the residual approach to data sets generated according to Examples 7.1–7.3.

Table 7.5 summarises the results obtained when complex mean structure is correctly accounted for in the estimation of Bayesian networks for the data sets generated according to Examples 7.4–7.6. The results show that few spurious edges are included in the highest-scoring Bayesian networks found when this approach is taken. These results must be compared with those presented in the bottom half of Table 7.1, obtained when S_0 is used in the analysis of these data sets. Upon such comparison, it can be seen that a score metric that takes account of the presence of a complex mean structure performs much better than in a score metric that does not take account of such complexity.

The consequences of the assumption of a more complex mean structure than is actually present in a data set is investigated through the application of the residual approach to the data sets generated according to Examples 7.1–7.3. When the residual approach is applied to Example 7.1, the mean structure of Example 7.4 is assumed, in the analysis of data sets generated according to Example 7.2, the mean structure of Example 7.5 is assumed, and in the analysis of data sets generated according to Example 7.3, the mean structure of Example 7.6 is assumed. The results so obtained are summarised in Table 7.6. Examination of the results in this table indicates that wrongly supposing a complex mean structure in the analysis of a data set does little to affect the networks obtained.

Hence, it can be concluded that the residual approach to the estimation of Bayesian networks performs quite well when the mean structure is correctly specified. Further, when the mean structure is supposed to be more complex than is actually the case, the residual approach performs similarly well, as was to be expected.

7.2 The Use of S_1 and S_2 in the Estimation of Bayesian Networks

In the previous section, it was shown that the residual approach performed quite well in the estimation of Bayesian networks for data sets with a complex mean structure. Here, the performance of the score metrics S_1 and S_2 are evaluated. After initial assessments of the performance of these score metrics under ideal conditions, the sensitivity of the results obtained to the choice of parameters relating to these score metrics is assessed. If the networks obtained are very sensitive to changes in these parameters, then the utility of these score metrics will be limited to cases when the true values of the parameters are known, or can be estimated with high precision. If, on the other hand, the estimated Bayesian networks are not so sensitive to changes in these parameters, the random effects score metrics will be more useful.

Investigation into this is conducted through the application of S_1 and S_2, in conjunction with the High-dimensional Bayesian Covariance Selection algorithm, to data sets generated from Examples 7.4, 7.5 and 7.6. Data sets are generated from the systems of linear recursive equations in these examples for fixed parameter values, and are analysed for a range of these parameter values. Initially, analysis of these data sets is conducted given the true parameter values. Then, when S_1 is used, the sensitivity of the algorithm to changes in the size of the quadrature used to calculate S_1 and to the mis-specification of ϕ are investigated. The sensitivity of the algorithm to the mis-specification of υ, the parameter associated with S_2, is similarly examined. Note that due to the computational speed of S_1, S_1 is not used in the analysis of data sets generated according to Example 7.5.

Note that in what follows, ϕ^* denotes the specified value of ϕ, where ϕ is the true variance of the random effects used in the generation of the data. It is this value, ϕ^*, that is used in the calculation of S_1. Similarly, υ^* is the specified value of υ: υ is the parameter used in the generation of the data, υ^* is used in the calculation of S_2.

It is here that the results of Section 4.2.5 should be brought to mind. In that section it was shown that as $\frac{\phi^*}{\psi^*} \to 0$, both of the score metrics S_1 and S_2 approach the score metric obtained when no random effects are present. Hence, as smaller values of ϕ^* are selected in the use of S_1, and as larger values of υ^* are selected in the use of S_2, the results obtained should approach those obtained when the presence of random effects is ignored, and the original score metric is used. This implies that the performance of S_1 for decreasing values of ϕ^* and of S_2 for increasing values of υ^* for a particular example will be largely dependent upon the performance of the original score metric in the analysis of that example.

7.2.1 The Use of S_1

The use of S_1 in the analysis of data sets generated according to Examples 7.4 and 7.6 is investigated in this section. First, the true value of ϕ and quadrature of size 50 are used in the analysis of these data sets. Then, the sensitivity of the Bayesian networks found through the application of S_1 to the misspecification of ϕ and the use of smaller quadratures is examined.

Table 7.7 summarises the results of using S_1 to analyse the data sets generated according to Examples 7.4 and 7.6 under ideal conditions. In these analyses, the true value of ϕ was used, and a quadrature of size 50, the largest quadrature size considered here, was used. Recall that for Example 7.4, $\phi = 5$, and in Example 7.6, $\phi = 4$. The results shown in the table indicate that the highest-scoring Bayesian networks obtained when S_1 is used in ideal

Table 7.7: Mean and standard deviation of the number of spurious and correct edges in the highest-scoring Bayesian networks obtained when S_1 is applied to data sets simulated according to Examples 7.4 and 7.6, using the true value of ϕ and quadrature of size 50.

Example	ϕ	True Edges	Spurious Edges	Correct Edges
7.4	5	0	0.1(0.32)	-
7.6	4	3	0.1(0.32)	2.3(0.68)

Table 7.8: Mean and standard deviation of the number of edges found in the analysis of data sets from Example 7.4 using S_1, (standard deviation in brackets).

	ϕ^*				
Quadrature Size	0.005	0.05	0.5	1	2
10	0.1(0.32)	0.1(0.32)	0.2(0.42)	0.1(0.32)	0.1(0.32)
25	0.1(0.32)	0.1(0.32)	0.2(0.42)	0.1(0.32)	0.1(0.32)
50	0.1(0.32)	0.1(0.32)	0.2(0.42)	0.1(0.32)	0.1(0.32)
	ϕ^*				
Quadrature Size	4	5	50	500	
10	0.1(0.32)	0.1(0.32)	0.1(0.32)	0(0)	
25	0.1(0.32)	0.1(0.32)	0.1(0.32)	0(0)	
50	0.1(0.32)	0.1(0.32)	0.1(0.32)	0(0)	

conditions are, on average, quite close to the true networks.

To assess the sensitivity of the high-scoring Bayesian networks found by the algorithm to changes in S_1, data sets are analysed for differing quadrature sizes and for varying values of ϕ^*. Given that ϕ will, in general, need to be estimated, how differing values of ϕ^* affect the estimation of Bayesian networks requires consideration. The results obtained for differing quadrature sizes are also of interest. Given a Bayesian network on a set of p variables, the calculation of the S_1 score of that network requires $p \times q$ function evaluations, where q is the size of the Gauss-Laguerre quadrature used in the estimation of the marginal model likelihood for each variable. Since p will in general be large, it is useful to know to what degree quadrature size q affects the results obtained by the High-dimensional Bayesian Covariance

Table 7.9: Mean and standard deviation of the number of spurious and correct edges found in the analysis of data sets from Example 7.6 using S_1, (standard deviation in brackets).

		ϕ^*								
Quadrature Size	Edges	0.005	0.05	0.5	1	2	4	5	50	500
10	Spurious	0.5(0.71)	0.5(0.71)	0.1(0.32)	0.1(0.32)	0.4(0.70)	0.5(0.71)	0.3(0.48)	0.1(0.32)	0.1(0.32)
	Correct	1.6(0.52)	1.7(0.68)	2.1(0.57)	2.2(0.42)	2.0(0.82)	1.8(0.79)	2.0(0.67)	2.0(0.82)	1.9(0.88)
25	Spurious	0.7(0.68)	0.4(0.70)	0.2(0.42)	0.3(0.67)	0.2(0.63)	0.1(0.32)	0.1(0.32)	0.3(0.48)	0.2(0.42)
	Correct	1.4(0.52)	1.6(0.70)	2.1(0.57)	2.0(0.67)	2.2(0.63)	2.3(0.68)	2.3(0.68)	1.6(0.70)	1.6(0.70)
50	Spurious	0.6(0.52)	0.5(0.53)	0.2(0.42)	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.2(0.42)	0.2(0.42)
	Correct	1.5(0.53)	1.5(0.71)	2.0(0.67)	2.3(0.48)	2.4(0.52)	2.3(0.68)	2.3(0.68)	1.8(0.79)	1.6(0.70)

Selection algorithm. The smaller q is, the less time required to run the algorithm, and the larger the data set S_1 may be used to analyse.

Analyses of the data generated according to Examples 7.4 and 7.6 using the S_1 score metric are then conducted for each value of $\phi^* \in \{0.005, 0.05, 0.5, 1, 2, 4, 5, 50, 500\}$ for quadratures of sizes 10, 25 and 50.

The number of edges in the highest-scoring Bayesian networks obtained when data sets generated according to Example 7.4 are analysed in this way is summarised in Table 7.8. This table gives the mean number of edges and the standard deviation of the number of edges in the highest-scoring Bayesian network found in the analysis of the data sets for each combination of ϕ^* and quadrature size. For example, when the data sets are analysed for $\phi^* = 0.005$, with a quadrature of size 25, the mean number of edges in the highest-scoring graphs obtained when the 10 data sets are analysed in this way is 0.1 edges, with a standard deviation of 0.95 edges.

Examination of Table 7.8 shows that, for these data sets, changes in quadrature size do not affect the number of edges in the highest-scoring networks found. Also shown by this table is the robustness of the highest-scoring networks to changes in ϕ^*.

The number of edges in the highest-scoring Bayesian networks obtained when data sets generated according to Example 7.6 are analysed in this way is summarised in Table 7.9. As was the case in Table 7.8, the mean number of edges and the standard deviation of the number of edges in the highest-scoring network found in the analysis of the data sets for each combination of ϕ^* and quadrature size is recorded.

Examination of this table shows that, for this example, the number of edges in the highest-scoring networks obtained are relatively constant for different quadrature sizes. It can also be seen that this quantity is also quite constant for changing values of ϕ^*, although it does appear that slightly fewer edges are found when ϕ^* is taken to be very large. Hence, the variability in the number of edges in the highest-scoring networks found for various combinations of ϕ^* and quadrature size appears to be primarily due to changes in ϕ^*.

From the two small examples considered here, it appears that when S_1 is used in conjunction with the High-dimensional Bayesian Covariance Selection algorithm in the analysis of data sets containing a small number of variables, the highest-scoring Bayesian networks visited

Table 7.10: Mean and standard deviation of the number of edges in the highest scoring networks found when the data sets generated by taking $\phi_i = \psi_i$ are analysed using S_0, (standard deviation in brackets).

Example	True Edges	Spurious Edges	Correct Edges
7.4	0	0.1(0.32)	-
7.5	0	123.6(2.99)	-
7.6	3	1.9(1.37)	2.2(0.79)

by the algorithm are quite robust to changes in quadrature size and ϕ^*. Due to the computational speed of the S_1 score metric, sensitivity analyses for larger examples were not conducted.

7.2.2 The Use of S_2

Here, the use of S_2 in the analysis of data sets generated according to Examples 7.4–7.6 is investigated. In order for the score metric under consideration to be consistent with the data under analysis, in the generation of the data sets considered in this section, ϕ_i is taken to be equal to ψ_i, so that $\boldsymbol{b}_i \sim N(\mathbf{0}, \psi_i I)$, implying that $\upsilon = 1$. Given the values of ψ_i previously generated, the values of $\boldsymbol{b}_i$ are generated, then treated as constants in the simulation of the data sets for each of the examples.

To allow a baseline for comparison, the data sets so generated are first analysed using S_0. The results of these analyses are summarised in Table 7.10. These data sets are then analysed using S_2, taking $\upsilon^* = \upsilon = 1$. These results are summarised in Table 7.11, and show that when S_2 is used given the true value of υ, the highest-scoring Bayesian networks found are quite close to the true networks.

Typically, the true value of υ will not be known, and some estimate of υ, here denoted υ^*, will be used in the calculation of S_2. To analyse the sensitivity of the algorithm to the misspecification of this parameter, the data sets generated taking $\upsilon = 1$ are analysed for

Table 7.11: Mean and standard deviation of the number of edges in the highest scoring networks found when the data sets generated by taking $\phi_i = \psi_i$ are analysed using S_2, $\upsilon^* = \upsilon = 1$, (standard deviation in brackets).

Example	True Edges	Spurious Edges	Correct Edges
7.4	0	0.1(0.32)	-
7.5	0	1.0(0.82)	-
7.6	3	0.4(0.70)	2.2(0.92)

each $\upsilon^* \in \{0.0001, 0.001, 0.01, 0.1, 0.5, 1, 2, 5, 10, 20, 50, 100, 500, 1000\}$.

The number of edges in the highest-scoring Bayesian networks obtained when data sets generated according to Examples 7.4–7.6 are analysed in this way is summarised in Table 7.12. For Examples 7.4 and 7.5, for each value of υ^*, the mean number of edges and the standard deviation of the number of spurious edges in the highest-scoring network is recorded. For Example 7.6, the mean number of edges and the standard deviation of the number of spurious and correct edges are recorded. For example, in the analysis of the data sets generated according to Example 7.6, when $\upsilon^* = \frac{1}{2}$, the mean number of spurious edges in the highest-scoring Bayesian networks for these 10 data sets is 0.6 edges, with a standard deviation of 0.84 edges. On average, 2.1 correct edges are found, with a standard deviation of 0.74 edges.

From Table 7.12, it can be seen that for values of υ^* smaller than or of the same magnitude as the true value of $\upsilon = 1$, changes in υ^* do not result in very substantial changes to the average number of spurious edges in the highest-scoring network obtained through the use of S_2. It can also be seen that as υ^* increases, the average number of edges in the highest-scoring networks approaches the average number of edges in the highest-scoring networks found when random effects are ignored in the estimation of Bayesian networks for the data. This is particularly apparent in the analysis of the data sets from Examples 7.5 and 7.6.

An explanation of this behaviour is as follows. Recall that as υ increases, the variance of the random effects decreases: $\phi_i = \upsilon^{-1}\psi_i$. Hence, the larger the value of υ used in the estimation of Bayesian networks, the less of the variation in the data is expected to be accounted for by the presence of a complex mean structure. As the true value of υ is smaller than the value

Table 7.12: Mean and standard deviation of the number of edges in the highest-scoring graphs found through the application of S_2 for varying values of υ^*, (standard deviation in brackets).

	υ^*						
Example	$\frac{1}{10000}$	$\frac{1}{1000}$	$\frac{1}{100}$	$\frac{1}{10}$	$\frac{1}{2}$	1	2
7.4 - Spurious Edges	0.2(0.42)	0.2(0.42)	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.1(0.32)
7.5 - Spurious Edges	1.4(0.97)	1.4(0.97)	1.4(0.97)	1.3(0.82)	1.0(0.94)	1.0(0.82)	1.4(1.07)
7.6 - Spurious Edges	0.9(1.10)	0.8(0.92)	1.0(0.82)	0.9(0.99)	0.6(0.84)	0.4(0.70)	0.6(0.70)
7.6 - Correct Edges	1.9(0.57)	2.2(0.63)	2.2(0.63)	1.9(0.74)	2.1(0.74)	2.2(0.92)	2.1(0.74)
	υ^*						
Example	5	10	20	50	100	500	1000
7.4 - Spurious Edges	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.1(0.32)	0.1(0.32)
7.5 - Spurious Edges	3.8(2.39)	15.8(3.46)	45.6(5.76)	84.6(4.17)	101.7(4.42)	118.4(4.09)	119.4(2.88)
7.6 - Spurious Edges	0.4(0.52)	1.1(0.88)	1.3(1.16)	2.1(0.88)	2.3(1.25)	2.2(1.48)	2.2(1.48)
7.6 - Correct Edges	2.5(0.53)	2.1(0.74)	2.2(0.79)	2.0(0.82)	1.8(0.79)	2.0(0.82)	2.0(0.82)

of υ^* used in the analysis, the complex mean structure actually accounts for more variation than indicated by this larger value of υ^*. This extra variation is accounted for through the inclusion of edges in the Bayesian network for the data.

This behaviour is consistent with the result obtained in Section 4.2.5. In that section it was found that as the variance of the exogenous variables that make up the complex mean structure becomes much smaller than the variance of the variables being considered, S_2 tends towards the score metric obtained when there is no complex mean structure. Hence, the high-scoring Bayesian networks obtained when υ^* is large tend towards those obtained when S_0 is used.

From the analysis of data sets from these three examples, it appears that the networks obtained through the use of High-dimensional Bayesian Covariance Selection algorithm are reasonably robust to the mis-estimation of υ. For values of υ^* orders of magnitude larger than the true value υ, the results of Section 4.2.5 imply that the extent of this robustness depends on the graphs obtained when a complex mean structure is ignored in the analysis. For selected values of υ^* of the same or smaller order of magnitude as the true value υ, high-scoring Bayesian networks obtained are very similar to those obtained when $\upsilon^* = \upsilon$ is used.

7.3 Consequences of Misspecification of the Distribution of ϕ_i

In Section 7.2, it was shown that the highest-scoring Bayesian networks found through the application of S_1 or S_2 in conjunction with the High-dimensional Bayesian Covariance Selection algorithm are reasonably robust to the misspecification of parameters associated with the distribution of the variance of the random effects. Typically, not only will the parameters of this distribution be misspecified, but the distribution itself may be misspecified. In this section, the robustness of the estimation of Bayesian networks, and the robustness of posterior inference on the random effects, to such model misspecification is discussed and investigated. The investigation proceeds through the analysis of data sets generated from the following example.

Example 7.7. *For this example, data sets are generated according to the following system of linear recursive equations:*

$$\begin{aligned} X_{ijk} =& b_{ij} + \epsilon_{ijk}, \\ \epsilon_{ijk} \sim& N(0, \psi_i), \;\; i = 1, 2, \ldots, 20, j = 1, 2, \ldots, 6, k = 1, 2, 3. \end{aligned}$$

The values of ψ_i are constant across the data sets that are generated, and are obtained by taking a sample of size 20 from an Inverse Gamma $(1, 1/2)$ distribution.

For each of the data sets generated, values of b_{ij} are obtained by sampling from a $N(0, \phi_i)$ distribution. Ten data sets are generated for each of the following assumptions on ϕ_i:

$$\begin{aligned} &M_1 : \phi_i = \phi = 4, \\ &M_2 : \phi_i = \psi_i, \\ &M_3 : \phi_i^{\frac{1}{2}} \sim \textit{Uniform}\,(0, 2)\,. \end{aligned}$$

For the ten data sets generated given model M_1, the values of b_{ij} are held constant, having been sampled from a $N(0, 4)$ distribution. For the ten data sets generated given model M_2, the values of b_{ij} are again held constant, obtained by sampling from $N(0, \psi_i)$ distributions. For the ten data sets generated given M_3, the values of ϕ_i are constant across data sets, with $\phi_i^{\frac{1}{2}}$ sampled from Uniform$(0, 2)$ distributions. The values of b_{ij} are also constant across data sets, sampled from $N(0, \phi_i)$ distributions.

The Bayesian network associated with this system of linear equations has no edges, and the true model for each variable may be written as

$$\boldsymbol{x}_i | \psi_i, \boldsymbol{b}_i \sim N_{18}\left(Q\boldsymbol{b}_i, \psi_i I_{18}\right), \tag{7.4}$$

where

$$\boldsymbol{x}_i = \begin{pmatrix} x_{i11} \\ x_{i12} \\ x_{i13} \\ x_{i21} \\ x_{i22} \\ x_{i23} \\ x_{i31} \\ x_{i32} \\ x_{i33} \\ x_{i41} \\ x_{i42} \\ x_{i43} \\ x_{i51} \\ x_{i52} \\ x_{i53} \\ x_{i61} \\ x_{i62} \\ x_{i63} \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{b}_i = \begin{pmatrix} b_{i1} \\ b_{i2} \\ b_{i3} \\ b_{i4} \\ b_{i5} \\ b_{i6} \end{pmatrix}.$$

Note that since there are 6 random effects in this model, S_3 may be used in the analysis of data generated from the model.

Thirty data sets are generated; ten for each assumption on the distribution of the variance of the random effects. Each of these thirty data sets is then analysed using each of the score metrics S_0, S_1, S_2, S_3 and the residual approach. The number of edges in the highest-scoring Bayesian network for each assumption/score metric combination are summarised in Table 7.13.

The tabulated results show that as the variance of the b_{ij} increases, the number of spurious edges in the highest-scoring networks found through the application of S_0 increases. This is to be expected, because as ϕ_i increases, the more variation due to the presence of the b_{ij} there will be in the data sets generated.

Table 7.13: Mean and standard deviation of the number of spurious edges in the highest-scoring Bayesian networks for data sets generated from Example 7.7, (standard deviation in brackets).

	Score Metric				
Distribution of ϕ	S_1 ($\phi^* = 4$)	S_2 ($\upsilon^* = 1$)	S_3	Residual Approach	S_0
$\phi = 4$	1.4(0.84)	0.4(0.52)	2.2(1.62)	0(0)	9.1(2.13)
$\phi_i = \psi_i$	1.1(0.88)	0.3(0.48)	2.7(1.16)	0.1(0.32)	0.7(0.82)
$\phi_i^{\frac{1}{2}} \sim \text{Uniform}(0, 2)$	1.0(0.47)	0(0)	2.4(1.58)	0(0)	2.1(1.10)

Consider now the results obtained when one of S_1, S_2, S_3 or the residual approach is used in the analysis of the data sets. The tabulated results indicate that while the number of edges in the highest-scoring Bayesian network found appears to depend upon the score metric used in the analysis of the data, there does not appear to be much dependence upon the distribution of ϕ_i used in the generation of the data. Consider, for example, the results obtained when S_1 is used in the estimation of Bayesian networks for the data sets. Recall that implicit in the use of S_1 is the assumption that the variance of the b_{ij} is fixed. It may be expected that the results obtained when S_1 is used in the analysis of data sets generated when $\phi = \psi_i$ or $\phi_i^{\frac{1}{2}} \sim \text{Uniform}(0, 2)$ would be different to the results obtained when S_1 is used in the analysis of data sets generated when ϕ is fixed. However, the table shows that no matter what distribution for the variance of the b_{ij} was used in the generation of data sets, similar results are obtained when S_1 is used in the analysis of those data sets. Similar results are shown for S_2, S_3 and the residual approach.

Further theoretical research into the consequences of the misspecification of the model on the high-scoring Bayesian networks found by the algorithm is required. Although it appears for the small example considered here that each score metric that takes account of a complex mean structure performs similarly no matter what the true distribution of ϕ_i is, this may not be the case for a larger example with a more complex Bayesian network.

7.3.1 The Effect of Model Misspecification on Posterior Estimation

The consequences of the misspecification of the prior distribution of ϕ_i on posterior inference for the random effects are now investigated. Posterior inference for the data sets generated as described in Example 7.7 is conducted, with posterior samples for each data set generated assuming each of the following models for the variance of the random effects:

$$\begin{aligned}M_1 &: \phi_i = 4;\\ M_2 &: \phi_i = \psi_i;\\ M_3 &: \phi_i^{\frac{1}{2}} \sim \text{Uniform}\,(0, 2)\,,\\ &\quad i = 1, 2, \ldots, 20.\end{aligned}$$

Recall from Section 4.4 that to obtain samples from the joint posterior distribution of $\boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i | \boldsymbol{x}_{P_i}$ via Gibbs sampling, the required conditional posterior distributions are

$$\begin{aligned}&\boldsymbol{\gamma}_i | \boldsymbol{x}_i, \psi_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i},\\ &\psi_i | \boldsymbol{x}_i, \boldsymbol{\gamma}_i, \boldsymbol{b}_i, \boldsymbol{x}_{P_i},\\ &\boldsymbol{b}_i | \boldsymbol{x}_i, \boldsymbol{\gamma}_i, \psi_i, \phi_i, \boldsymbol{x}_{P_i}\\ &\quad \text{and } \phi_i | \boldsymbol{b}_i.\end{aligned}$$

We will assume that the true directed acyclic graph underlying the Bayesian network from which the data sets were generated is known. In this case, the true directed acyclic graph is the graph with no edges. The required conditional posterior distributions are then

$$\begin{aligned}&\psi_i | \boldsymbol{x}_i, \boldsymbol{b}_i\\ &\boldsymbol{b}_i | \boldsymbol{x}_i, \psi_i, \phi_i\\ &\text{and } \phi_i | \boldsymbol{b}_i.\end{aligned}$$

For any of the three assumptions M_1, M_2 or M_3,

$$\boldsymbol{b}_i | \boldsymbol{x}_i, \psi_i, \phi_i \sim N_6 \left(\left(\frac{\psi_i}{\phi_i} I + Q^T Q \right)^{-1} Q^T \boldsymbol{x}_i, \psi_i \left(\frac{\psi_i}{\phi_i} I + Q^T Q \right)^{-1} \right).$$

Note that for Q as given in Example 7.7, $Q^TQ = 3I_6$, so that

$$\boldsymbol{b}_i|\boldsymbol{x}_i, \psi_i, \phi_i \sim N_6 \left(\frac{\phi_i}{3\phi_i + \psi_i} Q^T \boldsymbol{x}_i, \frac{\phi_i \psi_i}{3\phi_i + \psi_i} I \right),$$

and

$$b_{ij}|\boldsymbol{x}_i, \psi_i, \phi_i \sim N \left(\frac{\phi_i}{3\phi_i + \psi_i} \sum_{k=1}^{3} x_{ijk}, \frac{\phi_i \psi_i}{3\phi_i + \psi_i} \right). \tag{7.5}$$

Under assumption M_2, $\psi_i = \phi_i$, implying that this distribution may be written as

$$b_{ij}|\boldsymbol{x}_i, \psi_i \sim N \left(\frac{1}{4} \sum_{k=1}^{3} x_{ijk}, \frac{\psi_i}{4} \right). \tag{7.6}$$

Under the assumptions M_1 and M_3,

$$\begin{aligned}
\psi_i|\boldsymbol{x}_i, \boldsymbol{b}_i \sim & \text{Inv-Gamma}\left(\alpha, \beta\right); \\
\alpha = & \frac{n+\delta}{2}, \\
\beta = & \frac{\tau}{2} + \frac{1}{2} \left(\boldsymbol{x}_i - Q\boldsymbol{b}_i\right)^T \left(\boldsymbol{x}_i - Q\boldsymbol{b}_i\right).
\end{aligned}$$

On the other hand, under M_2,

$$\begin{aligned}
\psi_i|\boldsymbol{x}_i, \boldsymbol{b}_i \sim & \text{Inv-Gamma}\left(\alpha^*, \beta^*\right); \\
\alpha^* = & \alpha + \frac{m}{2}, \\
\beta^* = & \beta + \frac{1}{2} \boldsymbol{b}_i^T \boldsymbol{b}_i.
\end{aligned}$$

The final conditional posterior distribution only exists for $\phi_i^{\frac{1}{2}} \sim \text{Uniform}\left(0, 2\right)$:

$$\phi_i|\boldsymbol{b}_i \sim \text{Inv Gamma} \left(\frac{m-1}{2}, \frac{1}{2} \boldsymbol{b}_i^T \boldsymbol{b}_i \right).$$

Note that for the data sets generated according to Example 7.7,

$$n = 18, m = 6, \delta = 2, \tau = 1.$$

For each of the 30 data sets generated according to Example 7.7, posterior samples are obtained assuming each of M_1, M_2 and M_3. In this way, the effect of the misspecification of the prior distribution of ϕ_i on posterior inference may be assessed.

The sampling proceeds as described in Section 4.4.5, given the conditional posterior distributions presented above and using the R code given in Appendix B.3. Under each of the three prior assumptions on ϕ_i, for each of the thirty data sets, 2500 samples are simulated from the joint posterior distribution, with the first 500 samples discarded as burn-in.

For each of these 90 simulations, the posterior samples from the marginal posterior distributions

$$\begin{aligned}&\phi_i\\&\psi_i|\boldsymbol{x}_i \text{ and}\\&b_{ij}|\boldsymbol{x}_i \;\; i=1,2,\ldots,18,\;\; j=1,2,\ldots,6.\end{aligned}$$

are of the most interest. Samples from these marginal posterior distributions are obtained by looking at the samples from the joint posterior distribution $f\left(\psi_i, \boldsymbol{b}_i, \phi_i|\boldsymbol{x}_i, \boldsymbol{x}_{P_i}\right)$, and, for example, in the case of $\boldsymbol{b}_i$, ignoring the sampled values of ψ_i and ϕ_i.

In this analysis of the effect of prior misspecification on posterior inference for Example 7.7, only the parameters associated with a few variables will be considered. In particular, for the data sets generated under assumption M_1, parameters associated with X_1 and X_7 are considered. For the data sets generated under assumption M_2, parameters associated with X_5 and X_{11} are considered, and for data sets generated under M_3, parameters associated with X_6 and X_{17} are considered. It should be noted that the posterior samples of the parameters of the variables selected for analysis are not extreme in any way, but are representative of the general trends that may be observed across the posterior samples for all variables.

For each of the thirty data sets generated, posterior samples assuming each model M_1, M_2 and M_3 are simulated. For the data sets generated under each model, the posterior samples when the correct model is used for posterior simulation are discussed, then the consequences of misspecifying the prior distribution of ϕ_i on posterior inference is dicussed.

When the correct model is assumed in the generation of posterior samples, histograms of the samples from selected marginal posterior distributions given the first data set generated under that model are displayed. For example, Figure 7.3 displays histograms of the posterior samples from the marginal posterior distributions of b_{11}, ϕ_1, b_{71} and ϕ_7 given the first data set generated under M_1, when the correct prior distribution for ϕ_i is assumed.

For each of the thirty data sets generated, the posterior samples from selected posterior marginal distributions given these data sets are summarised using the sample median and the 90% posterior interval. For example, Figure 7.4 summarises the posterior samples of $b_{11}|\boldsymbol{x}_1$, $\psi_1|\boldsymbol{x}_1$ and ϕ_1 when the data $\boldsymbol{x}_1$ is generated under assumption M_1. The red bars on these plots indicate the true values of the parameters that were used in the generation of the data.

These figures allow investigation into the consequences of the misspecification of the distribution of ϕ_i on posterior inference, through the comparison of the posterior intervals obtained for each of the data sets under each of the assumptions M_1, M_2 and M_3.

Analysis of the data generated under $M_1 : \phi_i = 4$

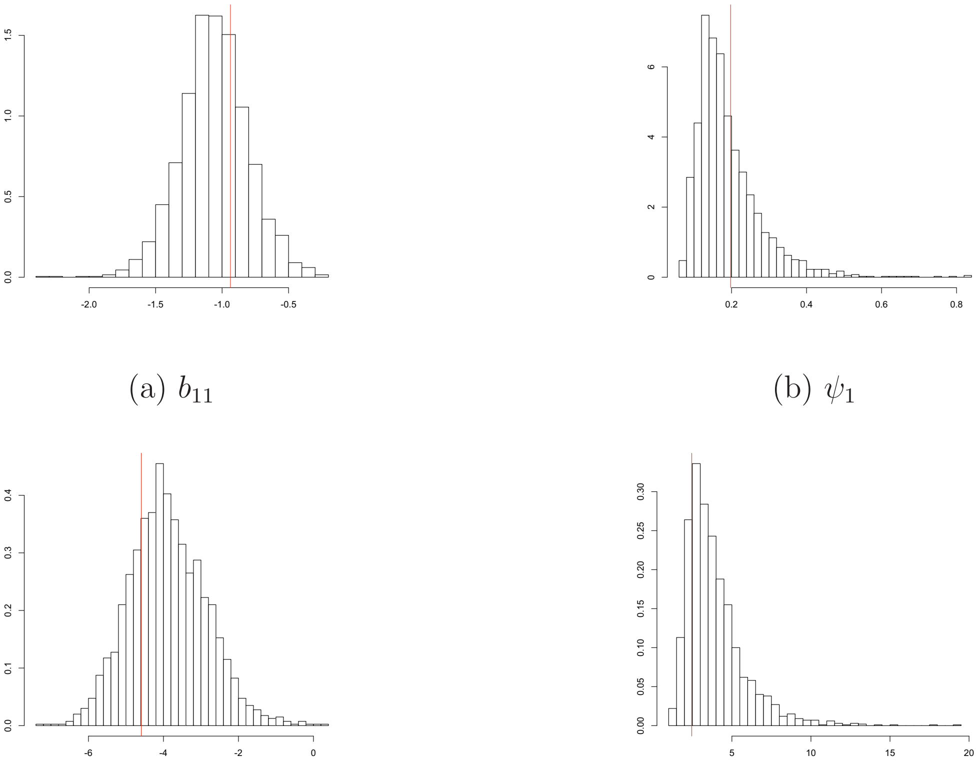
Here, the posterior analysis of the ten data sets generated under M_1 is discussed. Initially, the posterior analysis when the correct model is assumed is presented, then the consequences of the misspecification of the prior distribution of ϕ_i is discussed.

Figure 7.3 displays the histograms of the simulations from the marginal distributions of b_{11}, ψ_1, b_{71} and ψ_7 given the first data set generated under M_1. Note that the histograms given the other data sets generated under M_1, and those for other parameters, are similar. These histograms show that given the true prior distribution of ϕ_i, the marginal posterior distributions for the b_{ij} are centered quite closely to the value of b_{ij} used in the generation of the data. The marginal posterior distributions for the ψ_i behave similarly.

Figure 7.4 summarises the marginal posterior samples of b_{11}, ψ_1 and ϕ_1, and Figure 7.5 summarises the marginal posterior samples of b_{71}, ψ_7 and ϕ_7 given each of the data sets generated under M_1, assuming each of M_1, M_2 and M_3 in turn.

These figures show that for each of the data sets generated taking $\phi_i = 4$, the 90% posterior intervals for b_{11}, ψ_1, b_{71} and ψ_7, are similar when either M_1 or M_3 is assumed.

The posterior intervals obtained when M_2 is assumed in the simulation of posterior samples of b_{11} and ψ_1 are similar to those obtained when M_1 or M_3 are assumed. However, when M_2 is assumed in the simulation of posterior samples for b_{71} and ψ_7, the posterior intervals are



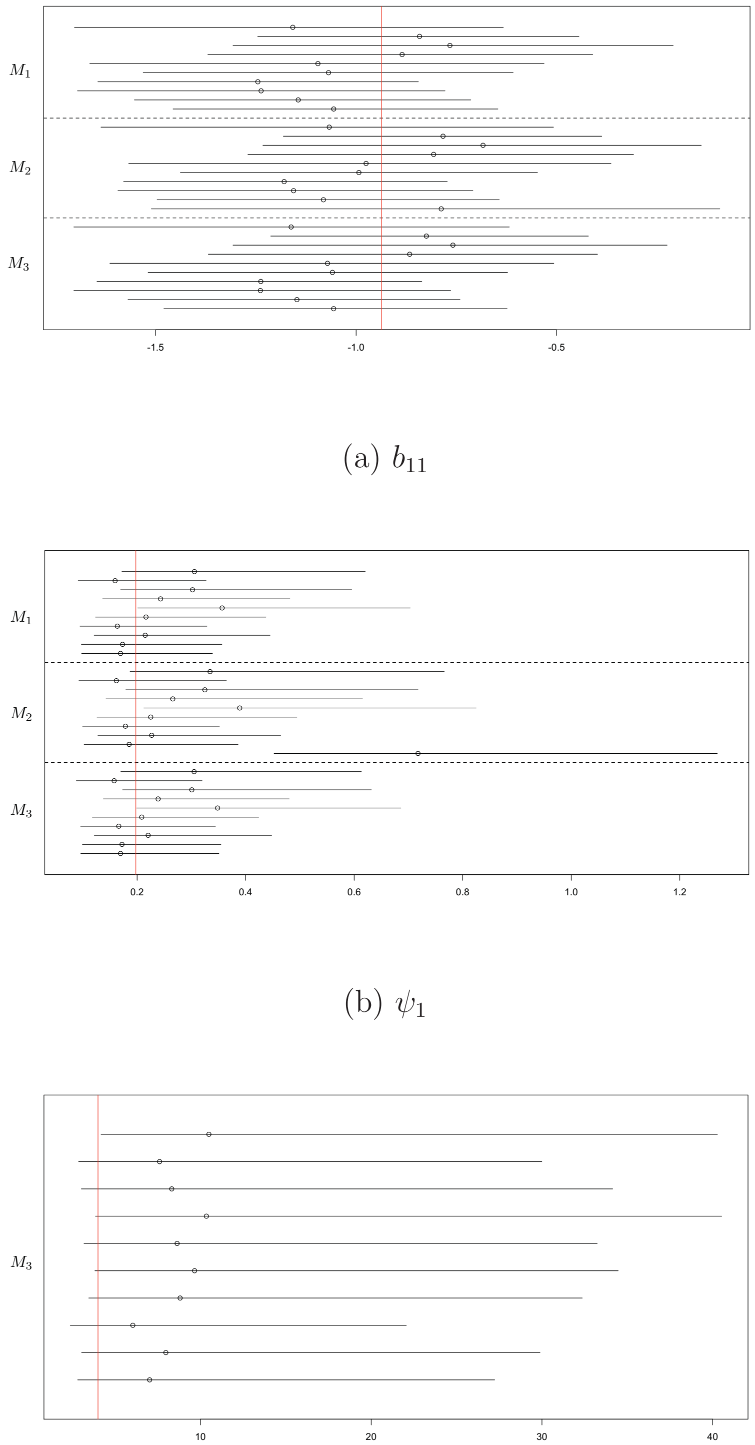
(a) b_{11}

(b) ψ_1

(c) b_{71}

(d) ψ_7

Figure 7.3: Histograms of the samples from the marginal posterior distribution of (a) b_{11}, (b) ψ_1, (c) b_{71} and (d) ψ_7, for the first data set generated under M_1, assuming M_1. The true values are marked in red.

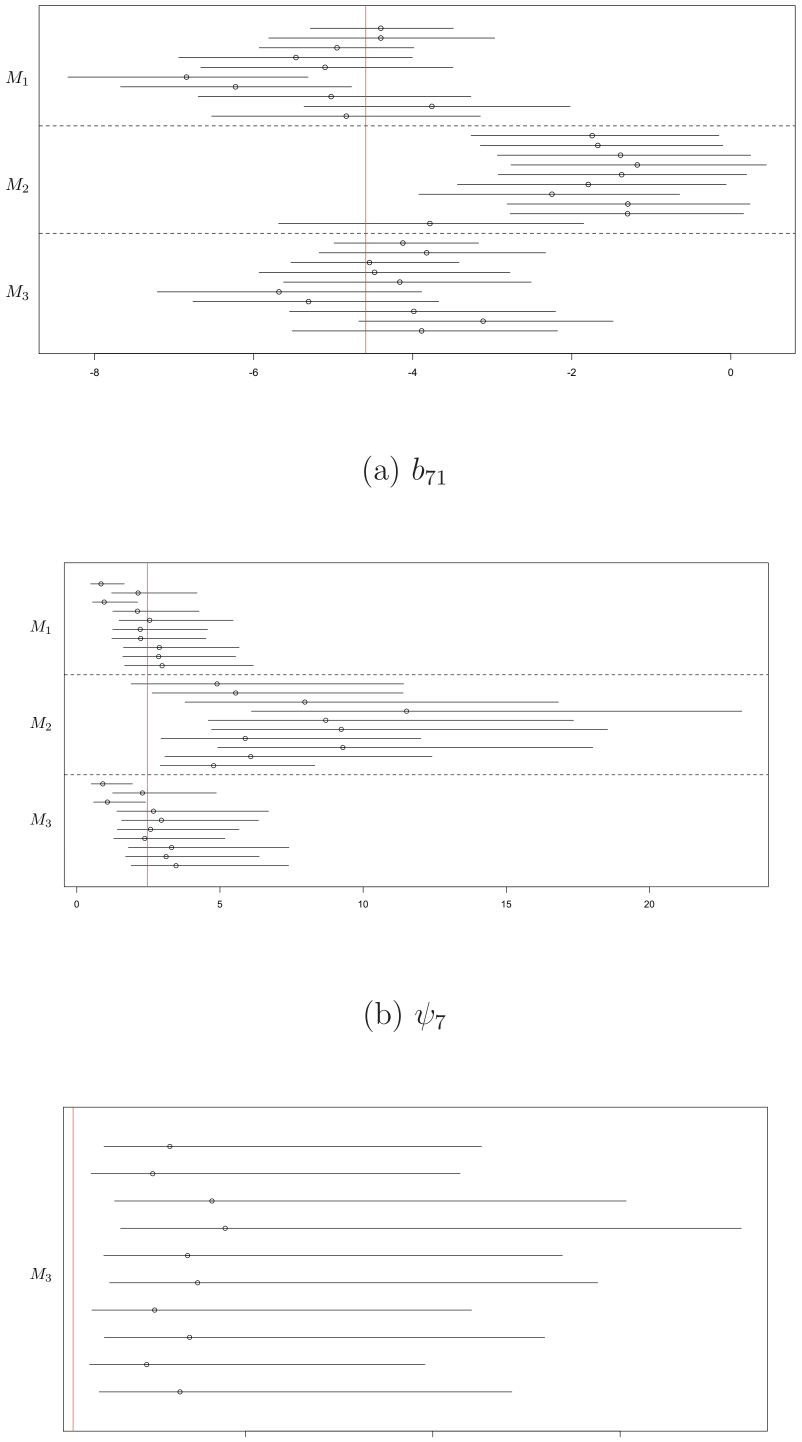


(a) b_{11}

(b) ψ_1

(c) ϕ_1

Figure 7.4: Medians and 90% posterior intervals for $b_{11}|\boldsymbol{x}_1$, $\psi_1|\boldsymbol{x}_1$, and ϕ_1 when $\boldsymbol{x}_1$ is generated under M_1. The red vertical bars indicate the true values.



(a) b_{71}

(b) ψ_7

(c) ϕ_7

Figure 7.5: Medians and 90% posterior intervals for $b_{71}|\boldsymbol{x}_7$, $\psi_7|\boldsymbol{x}_7$, and ϕ_7 when $\boldsymbol{x}_7$ is generated under M_1. The red vertical bars indicate the true values.

quite different, and these parameters tend to be overestimated.

Note that posterior estimates of ϕ_i are only available when M_3 is assumed. Figures 7.4(c) and 7.5(c) display the tendency of posterior samples to overestimate ϕ_i.

Analysis of the data generated under $M_2 : \phi_i = \psi_i$

Figure 7.6 displays the histograms of the simulations from the marginal distributions of b_{51}, ψ_5, $b_{11,1}$ and ψ_{11} given the first data set generated under M_2. The histograms given the other data sets generated under M_2 are similar. These histograms indicate that posterior estimates of the parameters when M_2 is correctly assumed are, in general, reasonably close to the true values of the parameters.

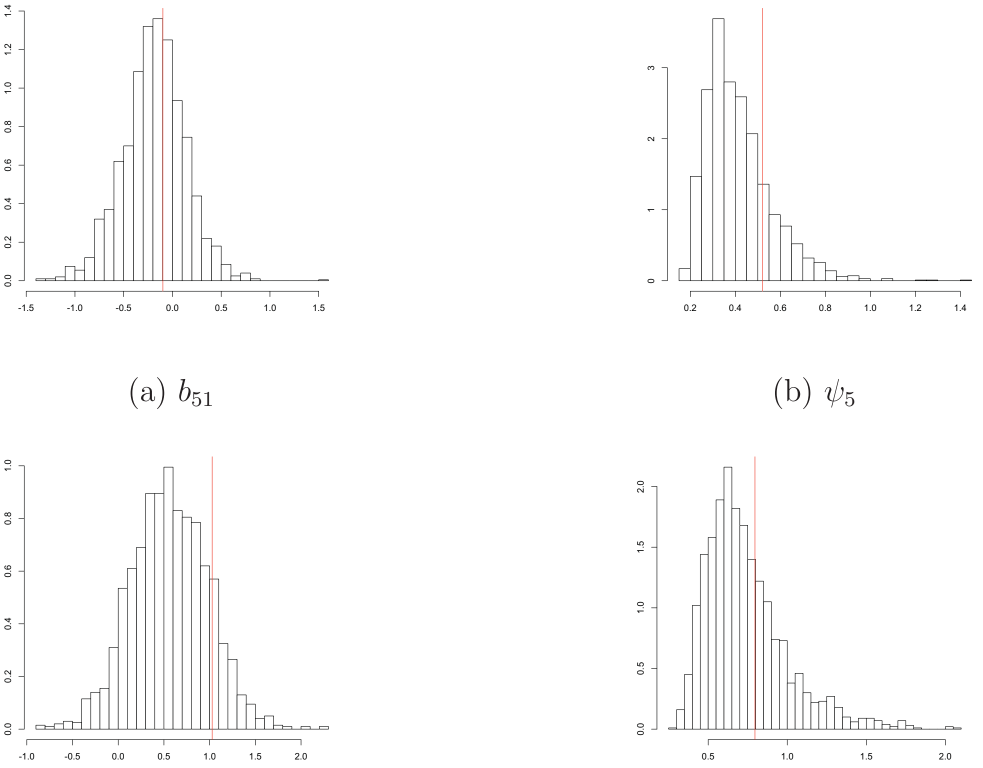
Figures 7.7 and 7.8 summarise the posterior simulations assuming each of M_1, M_2 and M_3 given each of the ten data sets generated under M_2. Figures 7.7(a) and 7.8(a) indicate that the posterior simulations of b_{ij} behave similarly for each of the three models assumed. Figures 7.7(b) and 7.8(b) show that when M_1 or M_3 are incorrectly assumed, the posterior intervals for ψ_i may be slightly wider than when M_2 is correctly assumed. Apart from this, posterior inference for ψ_i seems quite unaffected by incorrect specification of the prior distribution of ϕ_i.

Again, Figures 7.7(c) and 7.8(c) show that posterior estimates of ϕ_i, obtained when M_3 is assumed, tend to overestimate the true value of the parameter quite dramatically.

Analysis of the data generated under $M_3 : \phi_i^{\frac{1}{2}} \sim$ Uniform$(0, 2)$

Figure 7.9 displays the histograms of the simulations from the marginal distributions of b_{61}, ψ_6, ϕ_6, $b_{17,1}$, ψ_{17} and ϕ_{17} given the first data set generated under M_3. Again, the histograms given the other data sets generated under M_3 are similar to those displayed.

It can be seen that posterior estimation for b_{ij} and ψ_i when M_3 is correctly assumed performs similarly to when M_1 or M_2 are correctly assumed. Posterior estimation of ϕ_i performs quite



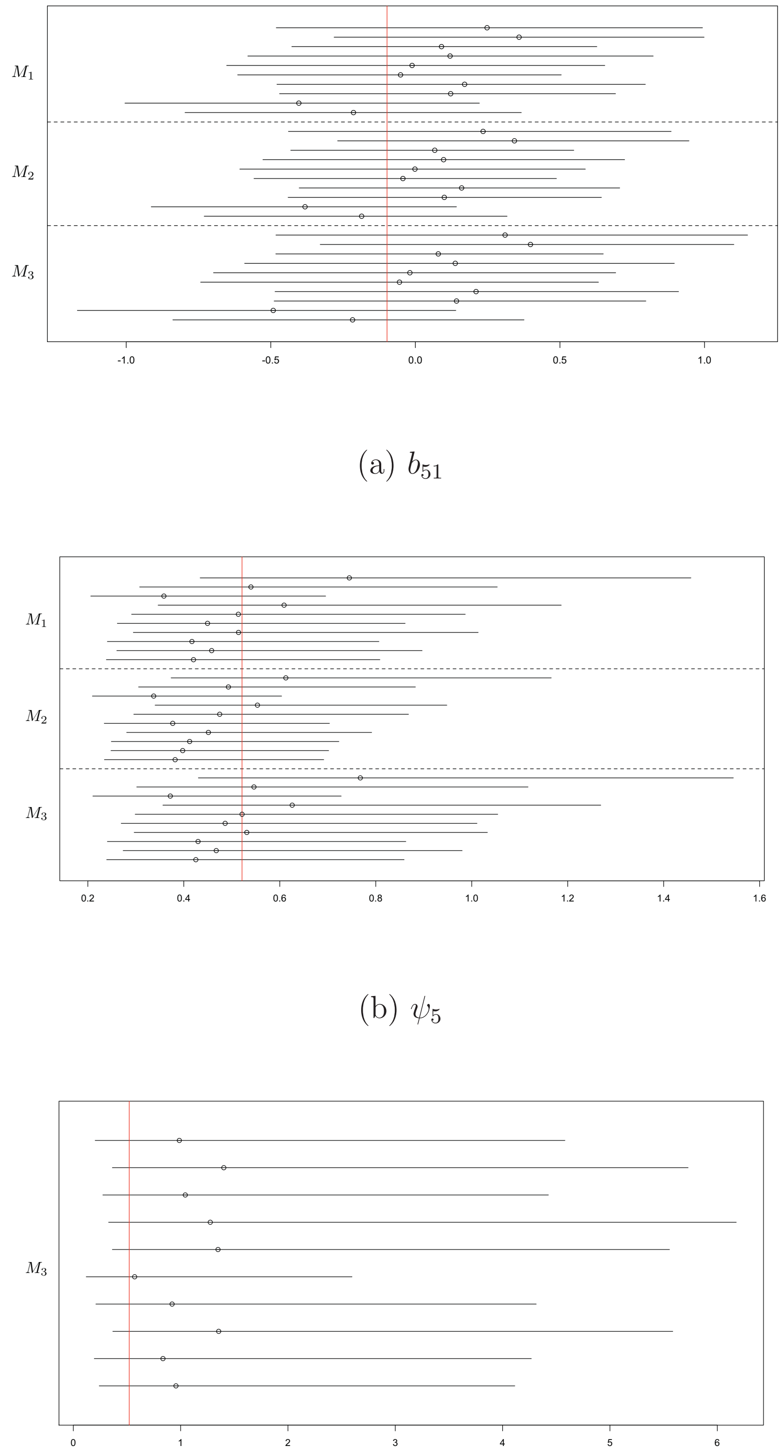
(a) b_{51}

(b) ψ_5

(c) $b_{11,1}$

(d) ψ_{11}

Figure 7.6: Histograms of the samples from the marginal posterior distribution of (a) b_{51}, (b) ψ_5, (c) $b_{11,1}$ and (d) ψ_{11}, for the first data set generated under M_2, assuming M_2. The true values are marked in red.

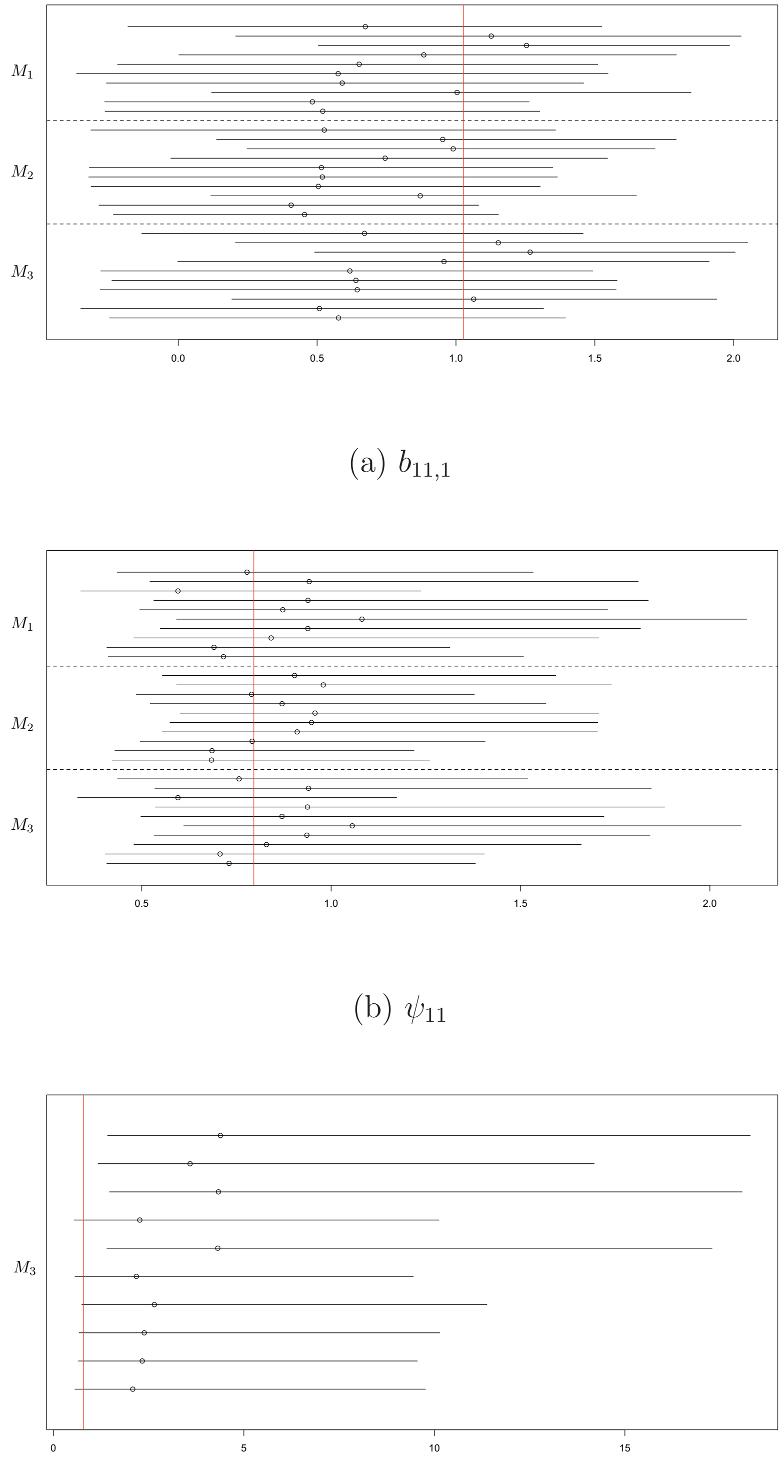


(a) b_{51}

(b) ψ_5

(c) ϕ_5

Figure 7.7: Medians and 90% posterior intervals for $b_{51}|\boldsymbol{x}_5$, $\psi_5|\boldsymbol{x}_5$, and ϕ_5 when $\boldsymbol{x}_5$ is generated under M_2. The red vertical bars indicate the true values.

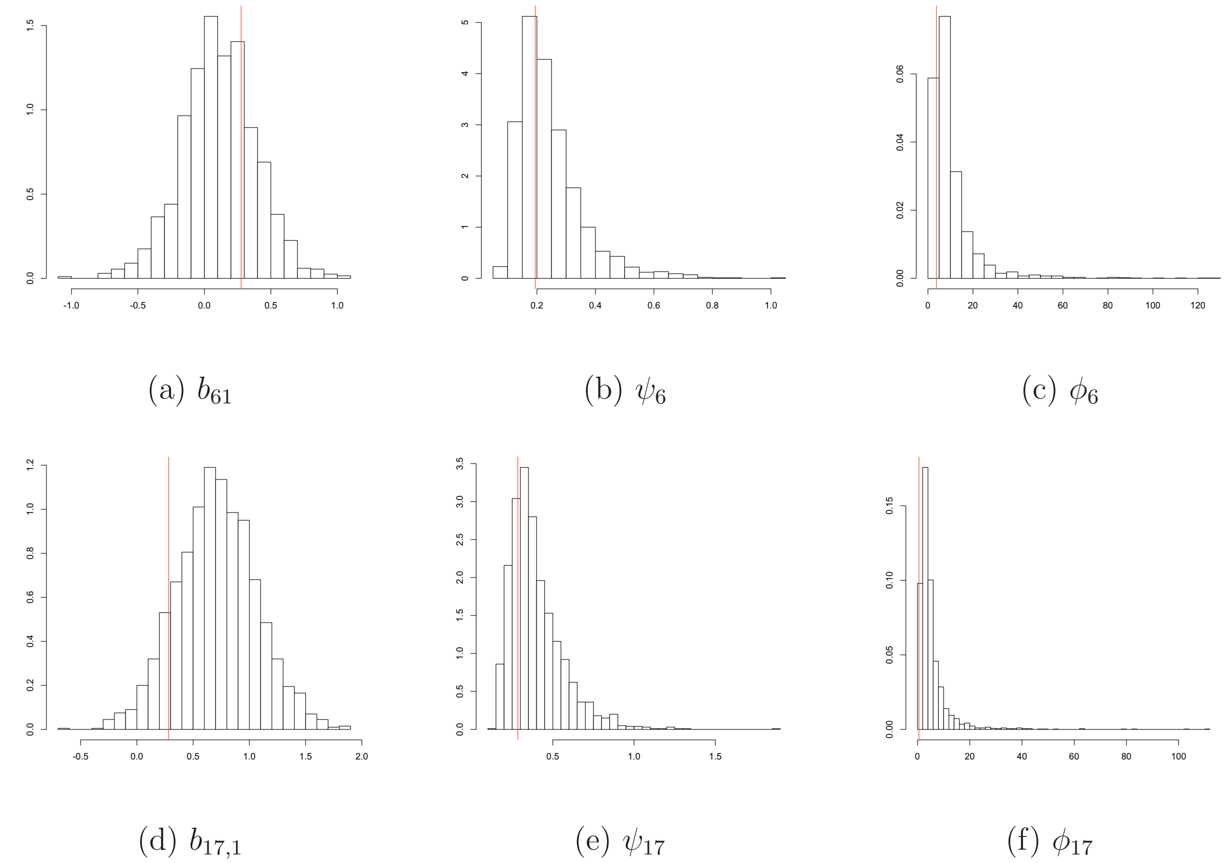


(a) $b_{11,1}$

(b) ψ_{11}

(c) ϕ_{11}

Figure 7.8: Medians and 90% posterior intervals for $b_{11,1}|\boldsymbol{x}_{11}$, $\psi_{11}|\boldsymbol{x}_{11}$, and ϕ_{11} when $\boldsymbol{x}_{11}$ is generated under M_2. The red vertical bars indicate the true values.



(a) b_{61}

(b) ψ_6

(c) ϕ_6

(d) $b_{17,1}$

(e) ψ_{17}

(f) ϕ_{17}

Figure 7.9: Histograms of the samples from the marginal posterior distribution of (a) b_{61}, (b) ψ_6, (c) ϕ_6, (d) $b_{17,1}$, (e) ψ_{17} and ϕ_{17} for the first data set generated under M_3, assuming M_3. The true values are marked in red.

poorly, due to the paucity of information available for the estimation of this parameter.

Figures 7.10 and 7.11 show that when M_3 is the correct model, the posterior samples for b_{ij} and ψ_i are quite similar when M_1 or M_3 is assumed.

When M_2 is assumed, the posterior medians of the samples from the marginal posterior distribution of ψ_i tend to be much larger than the value of ψ_i used in the generation of the data, and the posterior intervals tend to be wider than when either M_1 or M_3 is assumed. The posterior intervals for b_{61} obtained assuming M_2 are much wider than those obtained under M_1 or M_3. The posterior intervals for $b_{17,1}$ are similar under each of the three assumptions.

Figures 7.10(c) and 7.11(c) indicate that even when M_3 is the correct model, posterior estimates of ϕ_i tend to be much larger than the value of ϕ_i used in the generation of the data.

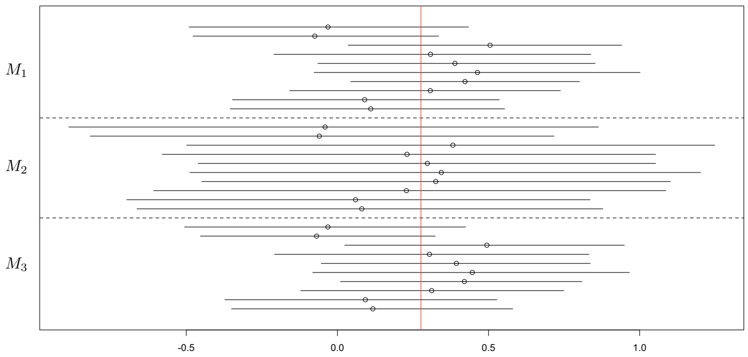
Conclusions

Several conclusions may be drawn from the analysis of Section 7.3.1. The first conclusion is that no matter what the true prior distribution of ϕ_i, posterior estimates of ϕ_i, necessarily obtained when M_3 is assumed, are quite poor. Given the small number of random effects in this example, such poor estimation is to be expected.

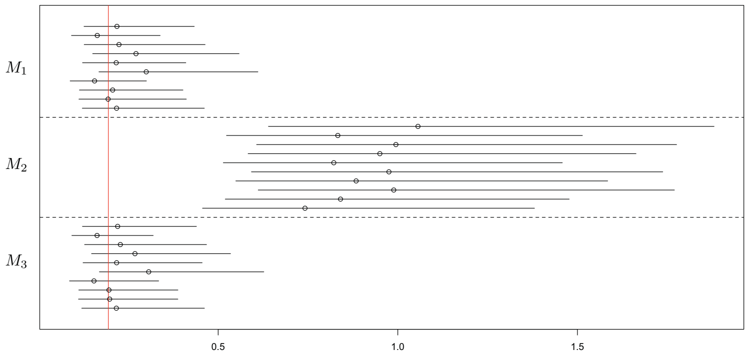
Second, it seems that the posterior samples of b_{ij} and ψ_i obtained when M_1 or M_3 are assumed behave similarly, no matter which distribution of ϕ_i was used in the generation of the data sets.

When M_2 is assumed in the analysis of data generated under M_1 or M_3, it was observed that the posterior samples obtained were, on occasion, quite different to those obtained when M_1 or M_3 were assumed. These observed differences are probably due to the difference between the prior magnitude of the ϕ_i assumed when M_2 is assumed, and the true magnitude of the ϕ_i.

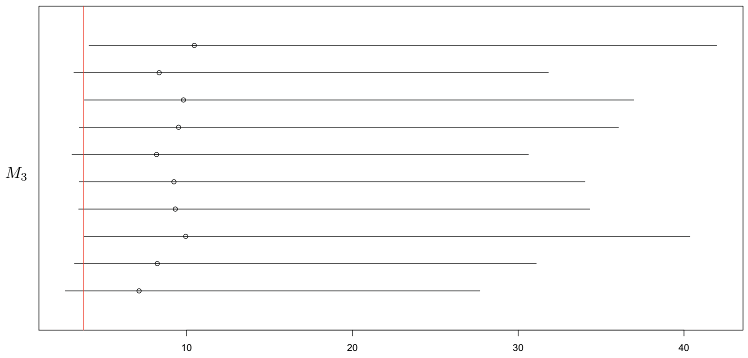
From the above, it can be concluded that, for this example, misspecification of the prior distribution of ϕ_i does not have a substantive effect on the posterior estimation of $\boldsymbol{b}_i$, provided the magnitude of the ϕ_i is apriori estimated accurately. However, it seems that posterior



(a) b_{61}

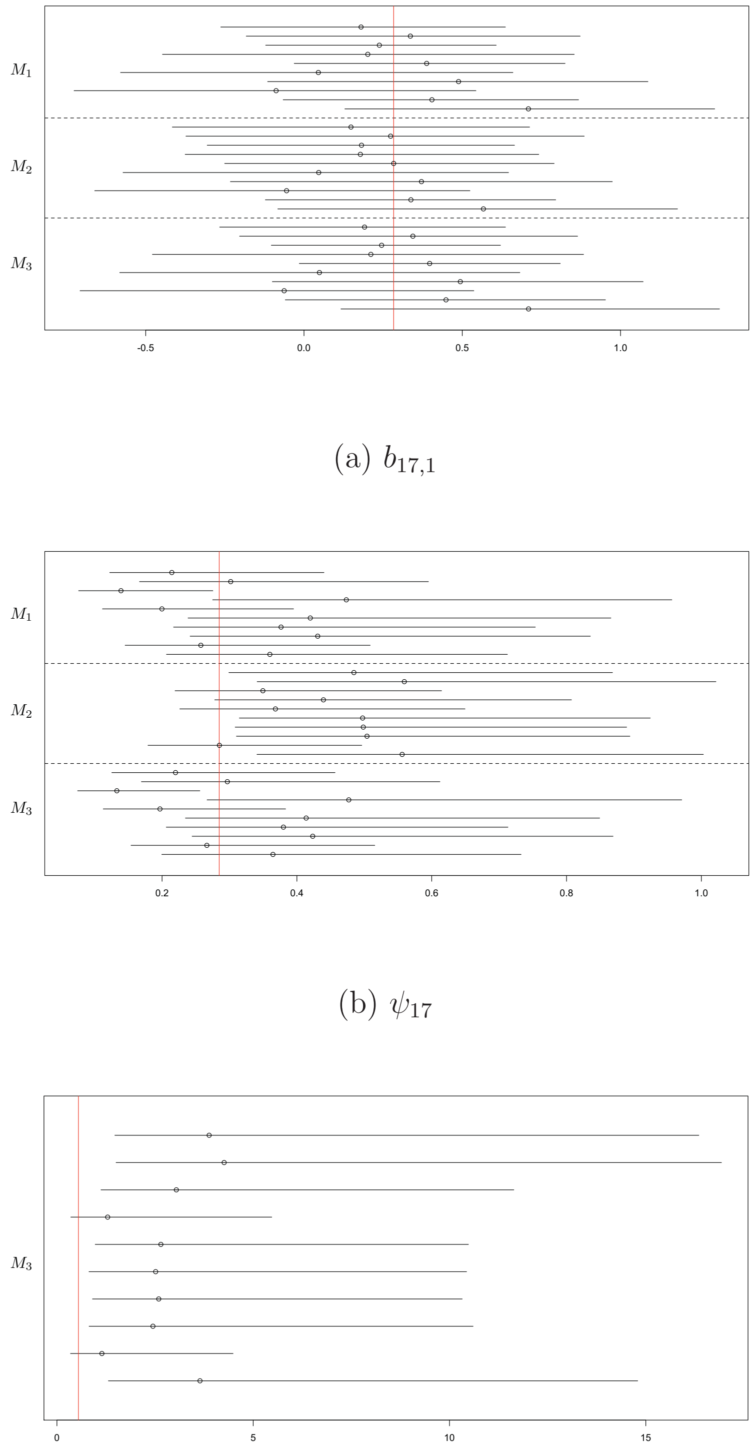


(b) ψ_6



(c) ϕ_6

Figure 7.10: Medians and 90% posterior intervals for $b_{61}|\boldsymbol{x}_6$, $\psi_6|\boldsymbol{x}_6$, and ϕ_6 when $\boldsymbol{x}_6$ is generated under M_3. The red vertical bars indicate the true values.



(a) $b_{17,1}$

(b) ψ_{17}

(c) ϕ_{17}

Figure 7.11: Medians and 90% posterior intervals for $b_{17,1}|\boldsymbol{x}_{17}$, $\psi_{17}|\boldsymbol{x}_{17}$, and ϕ_{17} when $\boldsymbol{x}_{17}$ is generated under M_3. The red vertical bars indicate the true values.

estimates of ψ_i are sensitive to the prior distribution used. It also seems that posterior estimation of ϕ_i will typically be poor, due to the small number of exogenous variables generally included in analyses.

7.4 Conclusions and Recommendations

In this chapter the score metrics S_1, S_2, S_3 and the residual approach to the estimation of Bayesian networks, in conjunction with the High-dimensional Bayesian Covariance Selection algorithm, were used to estimate Bayesian networks for example data sets with known network structure.

The necessity of such score metrics was shown in Section 7.1. It was shown that ignoring a more complex mean structure may result in the generation, by the High-dimensional Bayesian Covariance Selection algorithm, of high-scoring Bayesian networks with many spurious edges. The use of score metrics that take account of the presence of a complex mean structure was shown to produce high-scoring Bayesian networks with structures much closer to the true structure.

In Section 7.2, the sensitivity of the high-scoring Bayesian networks obtained through the use of S_1 and S_2 in conjunction with the High-dimensional Bayesian Covariance Selection algorithm to changes in ϕ and υ was assessed for some examples. It was found that provided the selected values of ϕ and υ are within an order of magnitude or two of the true parameter values, the high-scoring networks are not particularly sensitive to changes in these values.

The sensitivity of the high-scoring Bayesian networks obtained to the misspecification of the prior distribution of the random effects was investigated in Section 7.3. For the example considered, it was demonstrated that such prior misspecification did not affect the high-scoring Bayesian networks obtained to a great degree. It is conjectured that no matter what prior distribution on ϕ_i is assumed, the number of spurious edges in the highest-scoring Bayesian network obtained will be limited by the number obtained when random effects are ignored.

In Section 7.3.1, the effect of the misspecification of the prior distribution on posterior inference was investigated. The forms of the conditional posterior distributions obtained under each assumption on the form of the prior distribution of ϕ_i were first considered. It was then noted that due to the similarity of these distributions, prior misspecification should not have a great effect on posterior inference. Such a result was demonstrated for an example.

A more thorough theoretical treatment of the effects of the misspecification of the prior distribution of the random effects on estimated Bayesian networks and on posterior inference would be of interest but is beyond the scope of this thesis.

When analysing a data set with a complex mean structure, it is recommended, in the first instance, to appeal to the residual approach to the estimation of Bayesian networks, as no assumptions on the form of the prior distribution of the variance of the random effects are made. Then, if posterior estimates of the parameters that make up this complex mean structure are desired, the use of S_2 is recommended. The reasons for the recommendation of S_2 over either S_1 or S_3 is the computational speed of S_2 compared to the computational speeds of S_1 and S_3, and the results of this chapter that imply that results obtained should be quite robust to the misspecification of the prior distribution of ϕ_i.

Chapter 8

Analysis of the Grape Gene Data

This chapter considers the estimation of Bayesian networks for a gene expression data set that contains a complex mean structure. In the initial analysis of the data set, the samples of gene expression values are supposed to be independent and identically distributed. Given this assumption, the High-dimensional Bayesian Covariance Selection algorithm is used to estimate a Bayesian network for the gene expression data. The complex mean structure of the data is then considered, and the residual approach is used to control for such complexity in the estimation of a Bayesian network for the data. The Bayesian networks so obtained are then compared, and their biological plausibility discussed.

The gene expression data set considered in this chapter, discussed in more detail in Section 8.1, consists of fifty samples of gene expression levels for 26 grape genes. The grape berry tissue samples from which the gene expression levels were derived were taken from three different vineyards, in three different regions of South Australia. Note also that for each of the samples of gene expression, the temperatures at each vineyard in the hours leading up to the picking of the grape berries were recorded.

Table 8.1 provides information about the 26 genes considered here. The first column of Table 8.1 gives the reference number of each gene, used to refer to the genes in the following analysis. The second column of the table gives the Affymetrix probe set references. The third column gives the NCBI name of the gene, where Vvi stands for Vitis vinifera. Note that several of the genes with distinct Affymetrix probe set references have the same NCBI name. This is because these probe sets were later found to be targeting the same gene; different primers that bound to different regions of the same gene were used in the Affymetrix microarrays. Note that the following discussions are framed in terms of expression levels for the 26 genes, instead of referring to the expression levels of the 26 probes.

Descriptions of the proteins these genes code for are provided in the fourth column of Table 8.1. These 26 grape genes are known to code for heat shock proteins, [86], hereafter termed Hsps. These Hsps are responsible for protecting grapes against heat-induced stress, [86], hence, it is known that the expression levels of these genes are related to temperature, [87]. For a more detailed description of the functions of these proteins, see [86] and [87].

Given the known functions of the genes considered here and the climatic and geographic disparities between the regions where grape berries were sampled from, it would be remiss to ignore the effects of vineyard and temperature in the estimation of a Bayesian network for the grape genes. If the expression levels of these genes are strongly influenced by temperature, it follows that if the variation in the expression levels due to temperature is accounted for in the estimation of a Bayesian network, the resulting network should be much sparser than that obtained when such variation was not accounted for.

After a detailed description of the grape gene data set in Section 8.1, the effects of vineyard and temperature on the expression levels of the genes are ignored, and in Section 8.2, the High-dimensional Bayesian Covariance Selection algorithm is used to estimate a Bayesian network for the grape genes.

The effects of vineyard and temperature on the expression levels of the genes are then explored in Section 8.3. In Section 8.3.2, the residual approach is used to take account of these effects in the estimation of a Bayesian network for the genes. In Section 8.3.3, the S_2 score metric is used, and in Section 8.3.4, a combination of the residual approach and S_2 is used in the estimation of Bayesian networks for the grape genes. The networks so obtained are then compared to the one obtained when the effects of vineyard and temperature were ignored, and the biological plausibility of these networks is discussed.

Posterior estimation of the effects of different vineyards on the expression levels of the genes is considered in Section 8.5.

Note that the data set considered in this chapter consists of gene expression data for a very limited set of genes. As such, in the analysis of this data set, while it is hoped that the associations discovered will provide some insight into relationships between the genes, given the data set used, the methods applied could not be expected to find actual genetic regulatory pathways. The data available could not support any such conclusions. Instead, the focus is on the investigation of the conditional independence structure of these genes. Importantly, this chapter demonstrates the utility of the approaches presented in this thesis in the analysis of a real data set with an unknown network structure.

8.1 The Grape Gene Data

The grape gene data set considered in this chapter consists of expression values for each of the 26 grape genes, taken from 50 grape berry tissue samples. The grape berries from which gene expression levels were derived were all of the same variety and grown in three different vineyards: a vineyard in Clare, a Wingara vineyard, and a vineyard in Willunga, all located in South Australia. Twenty of the grape berry tissue samples were taken from the Clare vineyard, 20 were taken from the Wingara vineyard, and 10 were taken from the vineyard in Willunga. Note that all of the analysis was performed at the same laboratory. At each vineyard, temperatures at various times leading up to the picking of the grapes were recorded.

This data set is actually a subset of a larger set of data that consists of gene expression values obtained from 174 grape berry tissue samples, where 68 of the samples were taken from the Clare vineyard, 68 from the Wingara vineyard, and 38 from the Wingara vineyard. At the Clare and Wingara vineyards, four grape berry tissue samples were selected each week for 17 weeks. At the Willunga vineyard, 2 grape berry tissue samples were selected each week for 19 weeks. At each of the vineyards, the first samples were taken at fruit set, when the fertilised grape flowers began to form grape berries. Samples were then taken each week for a pre-specified number of weeks. In this way, gene expression levels were measured over the course of the development of the grape berries.

The data set from the Willunga vineyard consists of 34 grape tissue samples, with 2 samples taken on each of 19 sampling dates, roughly a week apart, occurring between November 20, 2003 and March 25, 2004. The times at which the samples were taken was recorded, with

Table 8.1: The grape heat shock genes considered here. The first column gives the reference number of the gene, the second column gives the Affymetrix reference numbers, and the third column gives the National Center for Biotechnology Information reference numbers. The fourth column provides a description of the function of the genes. This table is taken from [14].

Reference #	Affymetrix #	NCBI #	Protein Identity
15	1616246_at	Vvi.9142	Heat shock protein 70, ATP binding
1	1607002_at	Vvi.4801	Heat shock protein 70 ATP binding, luminal binding protein, glucose regulated
7	1610684_at	Vvi.2869	chloroplast HSP 70-1, ATP binding
11	1611740_at	Vvi.295	unknown
23	1620985_at	Vvi.4530	HSP21 chloroplast (could also be mitochondrial located)
17	1616995_at	Vvi.23518	Ubiquitin conjugating enzyme 4, ubiquitin-protein ligase
13	1614132_at	Vvi.863	Ubiquitin conjugating enzyme 4, ubiquitin-protein ligase
19	1618265_at	Vvi.15427	Ubiquitin conjugating enzyme 4, ubiquitin-protein ligase
2	1608052_s_at	Vvi.9085	HSP81(early response to dehydration) also similar to Hsp90-2, ATP binding
18	1618009_at	Vvi.9085	HSP81(early response to dehydration) also similar to Hsp90-2, ATP binding
20	1619931_s_at	Vvi.7394	HSP81(early response to dehydration) also similar to Hsp90-2, ATP binding
4	1608701_at	Vvi.2083	10 kDa chaperonin (Protein CPN10) (Protein groES)
3	1608164_at	Vvi.6787	Cytosolic class II 17.6 HSP
9	1611052_at	Vvi.6787	Cytosolic class II 17.6 HSP
10	1611192_at	Vvi.6787	Cytosolic class II 17.6 HSP
6	1610032_at	Vvi.6787	Cytosolic class II 17.6 HSP
14	1614330_at	Vvi.6787	Cytosolic class II 17.6 HSP
21	1620956_at	Vvi.3921	17.6 kDa class I small HSP
16	1616538_at	Vvi.7869	17.6 kDa class I small HSP
5	1609554_at	Vvi.7044	17.6 kDa class I small HSP
22	1620960_a_at	Vvi.7044	17.6kDa class I small HSP
24	1621652_at	Vvi.4464	17.6kDa class I small HSP
25	1622165_at	Vvi.6156	17.4kDa class I small HSP
12	1612385_at	Vvi.4422	17.4KDa class I small HSP
26	1622628_at	Vvi.5040	17.4kDa class III small HSP
8	1610700_at	Vvi.2537	23.6K mitochondrial small HSP

the samples on the first sampling date being taken at midday. On each of the remaining sampling dates the samples were taken at 11am. Air temperatures on each of the sampling dates were recorded 5.5 hours before the grapes were picked, and then every hour up until half an hour before the picking of the grapes. On the sampling day in the fourteenth week of the observation, the temperature at 2.5 hours before the grapes were picked was not recorded, and on the sampling date of the sixteenth week, temperatures at 3.5, 2.5 and 1.5 hours before picking were not recorded. This leaves 34 samples of gene expression levels taken from the Willunga vineyard with complete air temperature records.

The data set from the Clare vineyard consists of 68 grape tissue samples, 4 samples of which were taken on each of 17 sampling dates, roughly a week apart, occurring between November 23, 2004 and March 16, 2005. Two of each of these 4 samples were picked using spur pruning, while the other two were picked using machine pruning. However, it is not thought that the method of pruning affects the expression levels of any gene. In any case, for the purposes of the analysis that will be conducted here, any difference between the different methods of pruning used is inconsequential.

For the Clare data set, for most sampling days, the times of the picking of the grapes were recorded, as were air temperatures in degrees Celsius every 10 minutes from 6 hours before the grapes were picked to the time of picking. To avoid missing data problems, only temperatures recorded 5.5, 4.5, 3.5, 2.5, 1.5 and 0.5 hours before the picking of the grapes will be considered. No temperatures were recorded for the first sampling day, and on the second sampling day, very few temperatures were recorded, leaving 60 samples with complete temperature records.

Like the sample from Clare, the sample from the Wingara vineyard consists of 68 grape tissue samples, 4 samples of which were taken on each of 17 sampling dates. These sampling dates occurred roughly a week apart from November 10, 2004 to March 2, 2005. Two of each of these 4 samples were control samples, while the other 3 had a treatment known as "partial deficit". It is not thought that this treatment should affect gene expression levels. Again, note that any difference in expression levels due to these different treatments has no bearing upon the results obtained in the current chapter.

For the Wingara data set, on the first sampling day, all samples were collected at 12.30pm, while on all remaining sampling days, samples were collected at 11.30am. For all sampling

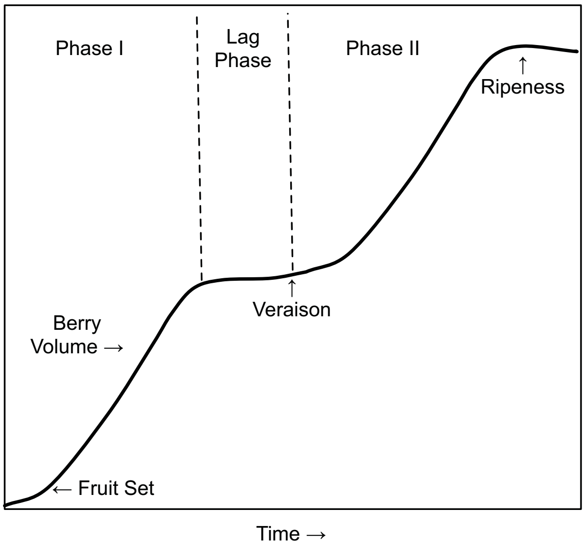


Figure 8.1: A schematic representation of the development of grape berries.

days, temperatures were recorded every ten minutes from 6 hours before the grapes were picked until the time of picking. Again, only the temperatures recorded 5.5, 4.5, 3.5, 2.5, 1.5 and 0.5 hours before the picking of the grapes will be considered in this analysis of the data.

The reduced data set considered here, consisting of 50 gene expression samples for each gene, consists of the samples from each vineyard taken in the third to seventh weeks of sampling, inclusive. Note that each of these samples have complete temperature records. The reason for the use of only these samples relates to the developmental cycle of grape berries. As discussed above, the expression levels of the grape genes in Table 8.1 were measured over the course of the development of the grape berries. Grape berries follow a double sigmoidal pattern of growth, that consists of two distinct growth phases with a lag period between these phases, [11, 68]. Figure 8.1 provides a schematic representation of grape berry development, displaying this double sigmoidal pattern.

Figure 8.1 shows that the second phase of grape berry growth commences upon the occurrence of an event known as "veraison". Veraison is the term used to describe the commencement of the colour change of grape berries. Robinson and Davies, [68], suggest that at veraison and during ripening, there are many changes in the expression levels of many different genes in grape berries. The relationships between the expression levels of genes during these changes are not well understood. It is for this reason that only data gathered in weeks 3 to 7 is considered. This time period occurs after fruit set, but well before veraison. It is thought that the relationships between expression levels of genes are quite stable during this time period. Hence, the assumption of identically distributed samples of gene expression values is more likely to be valid if only the data from this time slice is considered.

mRNA expression for each of these grape tissue samples was measured using Affymetrix Vitis vinifera oligonucleotide arrays. Background subtraction and normalisation was carried out using robust microarray analysis (RMA), as described in Irizarry *et al* [43].

8.2 Initial Analysis of the Grape Gene Data

In the initial estimation of a Bayesian network for the grape gene data, no attempt is made to account for the effects of vineyard and temperature on the expression levels of the genes. That is, if $\boldsymbol{x}_i$ is the 50-vector of the log-scale expression levels for grape gene i, it is assumed that

$$\begin{aligned} \boldsymbol{x}_i | \boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i &\sim N_{50}\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i, \psi_i I_{50}\right), \\ \boldsymbol{\gamma}_i | \psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I_{a_i}), \\ \psi_i^{-1} &\sim Ga\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right). \end{aligned} \tag{8.1}$$

where $a_i = |P_i|$ and $\boldsymbol{x}_{P_i}$ is a $50 \times a_i$ matrix. The columns of this matrix consist of the expression levels of the grape genes in the data set that the expression level of gene i is dependent upon and $\boldsymbol{\gamma}_i = (\gamma_{ij})_{j \in P_i}$. γ_{ij} is the effect of the expression level of gene j on the expression level of gene i. Following Dobra *et al*'s analysis of Affymetrix gene expression data, [23, 25], $\tau = 1$ and $\delta = 2$.

Note that in what follows, references to expression levels of the grape genes should be taken to mean the log-scale expression levels of the genes.

The application of High-dimensional Bayesian Covariance Selection to the grape gene data set results in the estimation of a Bayesian network for the grape genes in Table 8.1. Given that the grape gene expression levels available here are observational, causal interpretations should not be applied to the directed edges present in any network estimated given this

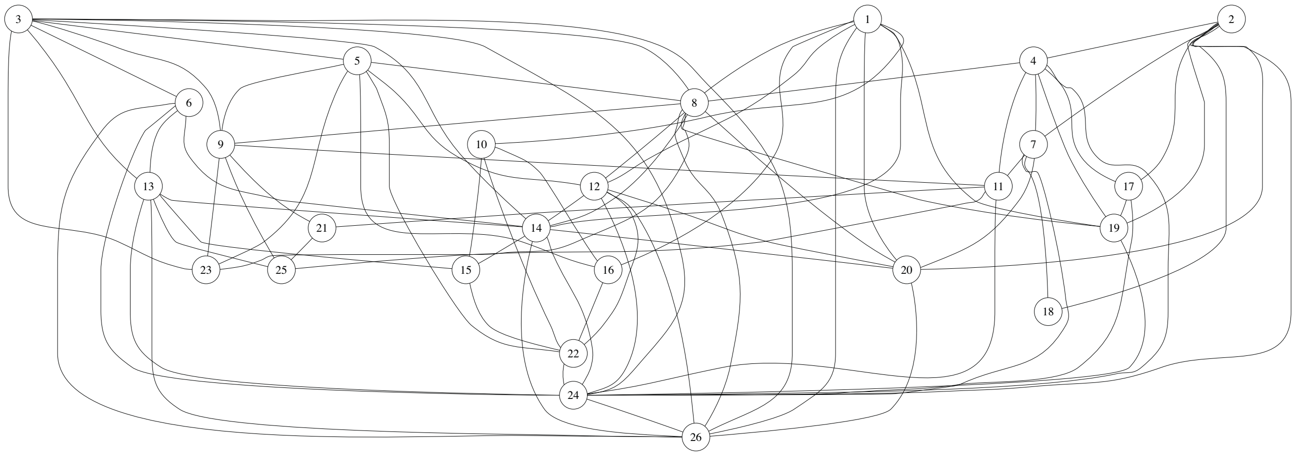


Figure 8.2: The moral version of the highest-scoring graph obtained for the grape genes, when vineyard and temperature are not accounted for.

data. Hence, the moralised version of the directed acyclic graph is used to summarise the conditional independence relationships of the grape genes.

The highest scoring Bayesian network found through the application of High-dimensional Bayesian Covariance Selection has 43 edges. The moralised version of the directed acyclic graph has 84 edges, and is displayed in Figure 8.2. Note that the placement of the vertices is arbitrary, and that the edges alone provide information about the conditional independence relationships of the genes.

Graphs were also estimated separately for the data from each vineyard. The highest scoring graph found for the Clare data had 22 edges, the highest scoring graph for the Wingara data had 23 edges, and the highest scoring graph for the Willunga data had 17 edges. These highest-scoring graphs were all quite different: only two edges were common to all three graphs (the edges between genes 21 and 25 and between genes 22 and 24), with the Clare and Wingara graphs sharing 8 edges, the Willunga graph having three edges in common with both the Wingara graph and the Clare graph. Given the paucity of the data, and the complexity of the models being analysed, the lack of concordance between the graphs obtained separately for each vineyard is not surprising.

When these vineyard-specific graphs are compared to the graph of Figure 8.2, however, it is seen that many of the edges in each of these individual graphs are contained in the graph obtained through the use of all of the data. The graph in Figure 8.2 has 18 edges in common with the Clare graph, 16 edges in common with the Wingara graph, and 9 edges in common with the Willunga graph. It seems that the sparsity constraints of the HdBCS algorithm will prevent more of the edges in the individual vineyard graphs being contained in the graph based on all of the data. In an effort to make the most efficient use of the data, in what follows, we will concentrate on models that incorporate data from all three vineyards.

In the estimation of the graph for the full data set, no attempt to account for the effects of vineyard and temperature was made. However, it is reasonable to assume that the expression levels of different genes will be differently affected by conditions such as soil type, solar radiation, rainfall, etc., that are present at each of the different vineyards. Also, since the genes in Table 8.1 are heat shock genes, it is known that the expression levels of these genes will be affected by changes in temperature. Hence, the effect of temperature and vineyard should be accounted for in the estimation of Bayesian networks for the grape gene data.

In Chapter 7, it was demonstrated that when the presence of a complex mean structure is ignored in the estimation of Bayesian networks for a data set, the highest-scoring networks obtained through the application of High-dimensional Bayesian Covariance Selection may differ quite remarkably from the true network from which the data set was drawn. Hence, it can reasonably be assumed that ignoring the effects of vineyard and temperature on the grape gene expression levels may result in estimated Bayesian networks that do not encode the same conditional independence relationships as are present in the true inverse covariance matrix of the genes. As a result of this, the graph obtained above may not provide an accurate representation of the conditional independence structure of the grape genes.

In Section 8.3, how to appropriately include temperature and vineyard effects in the model for the grape genes is discussed. Such effects are then included in the model, and Bayesian networks given this model are generated. In Section 8.4, the Bayesian networks so obtained are compared with the network obtained above.

8.3 Taking Account of Vineyard and Temperature Effects in the Analysis of the Grape Gene Data

The dependence of the grape gene expression values on changes in vineyard and temperature is now considered. As demonstrated in Section 7.1, if a data set has a complex mean structure, such a structure should be accounted for in the estimation of a Bayesian network for the data, lest spurious edges be included in the highest-scoring network. Hence, if it is found that the expression levels of the grape genes differ significantly from vineyard to vineyard, or that there is a relationship between expression level and temperatures at times before the picking of the grapes, such effects should be accounted for in the estimation of a Bayesian network for the genes.

Since the genes being considered here are known heat shock genes, the expression levels of these genes are expected to be dependent, in some way, upon temperature. That is, some of the variation in the gene expression values sampled is expected be due to the different temperatures on the different sampling days. For this reason, the relationship of each gene to temperature must be taken into account in the estimation of a Bayesian network for the grape genes. It also seems reasonable to assume that the mean expression levels of genes will differ from vineyard to vineyard. For visual summaries of how the expression levels of each gene differ from vineyard to vineyard, see the boxplots in Appendix C.1. Note also that the temperature profiles at each of the vineyards for each of the samples are displayed in Figure 8.3.

The question of how best to include temperature and vineyard effects in the model for gene expression is investigated using linear regression models with forward and backward selection.

It could be argued that instead of including each of the temperatures for each sample, average temperatures, or the average of the first three temperatures and the last three temperatures, could be used instead. However, the sharp increases in temperature displayed for some samples in Figure 8.3 would not be captured by such data reduction. Hence, from this point, only models containing the raw temperatures are considered.

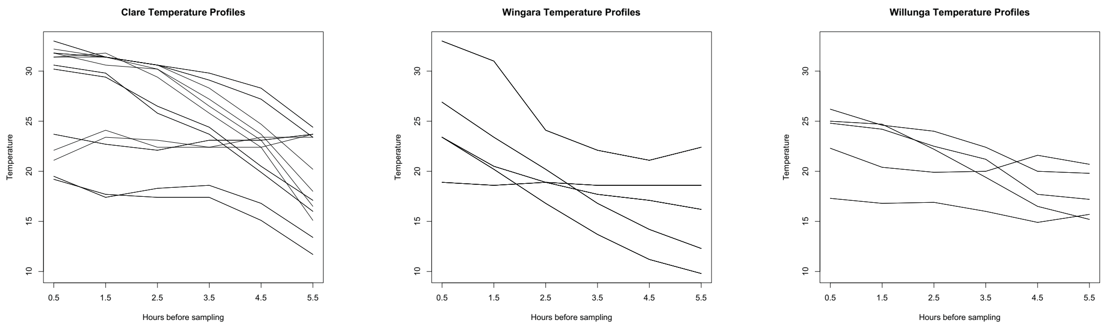


Figure 8.3: The temperatures at each vineyard at the times leading up to the picking of the grapes.

The largest model for each gene that will be considered contains separate intercepts for the data from each vineyard, terms for each of the temperatures recorded $\{30, 90, 150, 210, 270, 330\}$ minutes before the grapes were picked, and all 2-way interactions of these temperatures, excepting the interaction between the temperatures 270 and 330 minutes before picking. This interaction is excluded as the resulting vector happens to be linearly dependent upon the set of the remaining vectors. This model contains 23 covariates: 3 vineyard effects, 6 main temperature effects, and the 14 second-order temperature interactions.

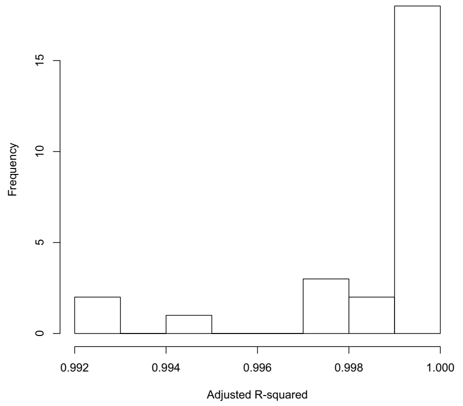


Figure 8.4: Histogram of the adjusted r^2s obtained after regressing the gene expression values on vineyard, temperature, and all 2-way interactions of the temperatures.

For each of the 26 genes, a step-wise approach is taken to finding the best regression model.

The step-wise procedure used is the `stepAIC` function from the `MASS` package, [84], in R, [76]. Examination of the results of this step-wise procedure indicate that the model containing all of the terms should be used. That is, there is not a single backwards elimination step that would apply to all genes. Summaries of the regressions of the expression levels of each gene on the vineyard and temperature covariates are shown in Appendix C.3. The adjusted r^2s for these regressions are all very high, and are summarised in Figure 8.4.

The selected model, containing 3 vineyard effects, 6 main temperature effects and 14 second-order temperature interactions is required because it is the simplest model that fits every gene. Of course, as a consequence, overfitting does occur for some genes, as can be seen in the regression summaries in Appendix C.3. For example, for gene 2, a highly significant F-statistic is observed, but none of the regression coefficients are individually significantly different from zero. While this indicates that separate models for each gene should be considered, such work is beyond the scope of this thesis.

In Figure 8.5, histograms of the marginal standard deviations of the expression data for each gene and the residual standard errors from the regressions containing vineyard, temperature and the two-way temperature interactions are shown. The histogram of the marginal standard deviations shows how variable the expression levels of each of the genes were before temperature or vineyard effects were taken into account. The histogram of the residual standard errors shows how much variation remains in the data after the relationships with temperature and vineyard have been accounted for in this way. The histograms show that for the genes considered here, changes in temperature and vineyard account for most of the variation in the expression levels.

Hence, it is clear that both vineyard and temperature need to be considered in the estimation of a Bayesian network for the grape genes. Given the amount of variation in the expression levels accounted for by these covariates, accounting for them in the estimation of a Bayesian network for the grape genes should result in a high-scoring Bayesian network with fewer edges than the one obtained in Section 8.2, the moral graph if which is shown in Figure 8.2.

Note that in order to estimate a Bayesian network for these gene expression levels under the assumption of a joint Gaussian distribution, the scatter plots of the residuals after fitting the above model, with vineyard, main temperature and two-way temperature interactions

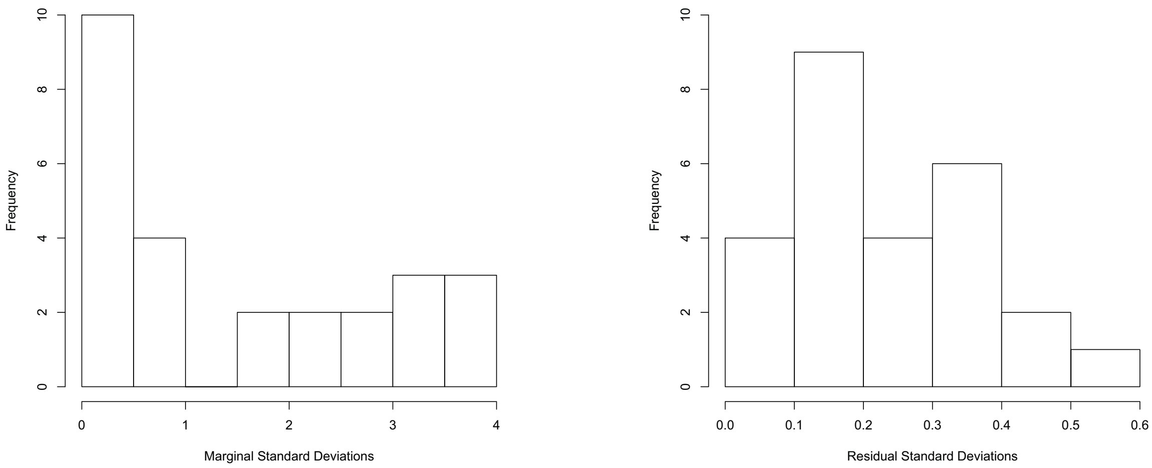


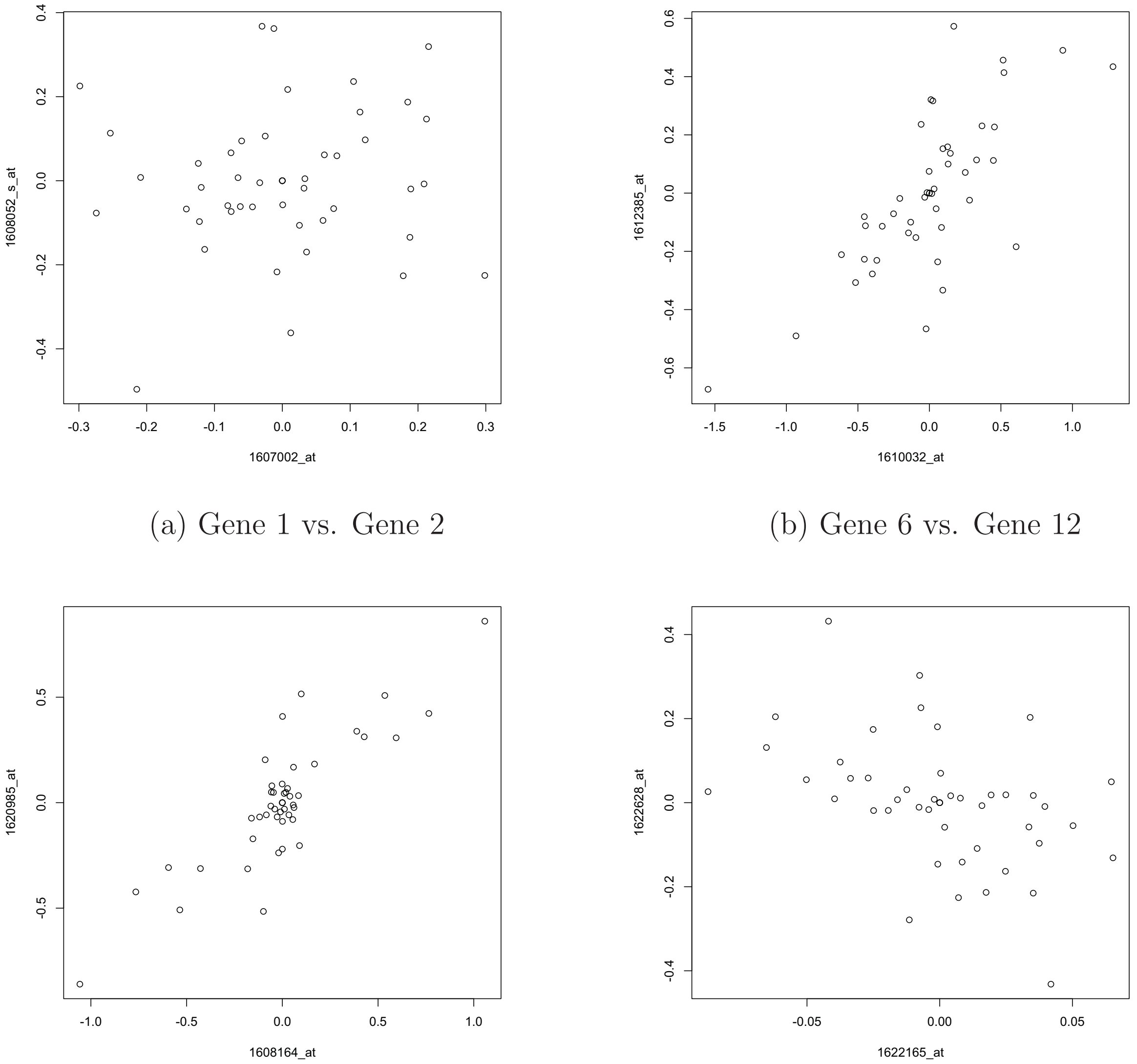
Figure 8.5: Histograms of the marginal standard deviations of the grape gene expression levels and the residual standard errors after regressing the expression levels on temperature and vineyard.

should show roughly linear relationships. Four such plots are displayed in Figure 8.6, and it can be seen that, for these pairs of genes at least, this requirement is satisfied.

8.3.1 Inclusion of the Effects of Vineyard and Temperature in the Model

The question is now how to include the effects of vineyard and temperature in the estimation of a Bayesian network for the grape genes. When the residual approach to the estimation of Bayesian networks is taken, as it is in Section 8.3.2, no assumptions on the form of the prior distribution of the effects of vineyard and temperature are made. However, if posterior estimates of such effects are desired, prior distributions on these effects need to be assumed.

In Bayesian analyses, there is little distinction between fixed and random effects, as prior distributions are placed on all unknown parameters. However, it is still worth considering whether it makes sense to treat vineyard and temperature effects as random effects in the analysis, and it is necessary to discuss which prior distributions are appropriate.



(a) Gene 1 vs. Gene 2

(b) Gene 6 vs. Gene 12

(c) Gene 3 vs. Gene 23

(d) Gene 25 vs. Gene 26

Figure 8.6: Scatterplots of the residuals after fitting the above model, with vineyard, main temperature and two-way temperature interaction effects for some pairs of genes.

Consider first the effects of different vineyards on the expression levels of the various genes. The vineyards from which grapes were sampled for this data set can be thought of as a random sample from the population of all vineyards. Alternatively, the effect of a particular vineyard upon the expression level of a gene could be thought of as a combination of the effects of soil type, solar radiation, and other such variables on the expression level of that gene. Either way, the effect of a particular vineyard upon the expression level of a particular gene can be thought of as a random effect. We make the further simplifying assumption that for each gene, the variance of the vineyard effects is related to the conditional variance of the expression levels of that gene in the same way. That is, the prior distribution of the effect of the vineyards on the expression levels of gene i is assumed to be

$$\begin{pmatrix} b_{i1}^V \\ b_{i2}^V \\ b_{i3}^V \end{pmatrix} \sim N_3(\mathbf{0}, \upsilon^{-1}\psi_i I_3), \tag{8.2}$$

where b_{ij}^V is the effect of vineyard j on the expression level of gene i, $j = 1 =$ Clare, $j = 2 =$ Wingara, $j = 3 =$ Willunga, υ is constant from gene to gene, and ψ_i is as in Equation (8.1). These vineyards can be considered to be blocking factors, and are not of intrinsic interest.

The inclusion of temperature effects then needs to be considered. If analysis was being conducted in a non-Bayesian setting, the effects of temperature on gene expression levels would best be included as fixed effects, as these parameters are meaningful, and the true values of these parameters are of interest. However, given that analysis is being conducted within a Bayesian framework, the effects of temperature are necessarily treated as random.

The simplest way to include the effects of temperature is to assume that, for each gene, these effects are independent and distributed identically to the vineyard effects of that gene. That is, if $\boldsymbol{b}_i$ is the vector of vineyard and temperature effects for gene i it is assumed that

$$\boldsymbol{b}_i | \psi_i \sim N_{24}\left(\mathbf{0}, \upsilon^{-1}\psi_i I_{24}\right).$$

Under this assumption, the appropriate score metric to use in the estimation of Bayesian networks for the grape genes is the S_2 score metric.

This assumption, while certainly convenient, is probably not valid. It seems unlikely that temperature effects are distributed identically to vineyard effects, and the effects of vineyard and temperature may not be independent. Additionally, if temperature effects are considered in isolation, it is unlikely that the effect of temperatures at different times are independent.

Alternatively, the temperature vectors could be replaced with an orthonormal basis, and the effects of these orthonormal basis vectors assumed independent and identically distributed. This approach is more reasonable than assuming the effects of the temperatures themselves are independent and identically distributed, but is still not entirely realistic. Such an approach is beyond the scope of this thesis.

Assuming a more complex covariance structure for the random effects would be more realistic. The assumption of such a covariance structure and the implications for the calculation of score metrics was discussed in Section 5.1. In this case, while more realistic, the score metric arising from such an assumption would be very computationally intensive to compute. For example, if a wholly unknown covariance matrix for the random effects was assumed, 301-dimensional numerical integration would be required to compute the associated score metric.

An alternative approach is available, combining the residual approach and the S_2 score metric. The residual approach could be used to remove the effects of temperature, and the prior distribution in Equation (8.2) could be placed on the vineyard effects. S_2 could then be used in conjunction with the High-dimensional Bayesian Covariance Selection algorithm to estimate Bayesian networks for the grape gene data. In this way, the only assumption made about the variance structure of the temperature effects is that these effects are independent of the vineyard effects. This approach allows the use of the S_2 score metric, given a model more realistic than assuming vineyard and temperature effects are independent and identically distributed.

8.3.2 Using the Residual Approach to Estimate a Bayesian Network for the Grape Genes

Using the residual approach, the association between the expression levels of the grape genes and vineyard and temperature is accounted for in the estimation of Bayesian networks for the genes. With this approach, no assumptions about the distribution of the effects of vineyard and temperature on gene expression need be made.

Four distinct models are assumed in the estimation of Bayesian networks for the grape genes. In the first model considered, only vineyard effects are accounted for. In the second model, only temperature effects are considered. In the third model, vineyard and main temperature effects are included, while in the fourth model, vineyard, main temperature effects and two-way temperature interactions are included. The resulting moral graphs are discussed in Section 8.4.

Vineyard Effects Only

When only vineyard effects are considered, the following model for the expression levels of each gene is assumed:

$$\begin{aligned}\boldsymbol{x}_i|\boldsymbol{x}_{P_i},\boldsymbol{\gamma}_i,\psi_i,\boldsymbol{b}_i^V &\sim N_{50}\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i+Q_V\boldsymbol{b}_i^V,\psi_i I_{50}\right),\\ \boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0},\tau^{-1}\psi_i I_{a_i}),\\ \psi_i^{-1} &\sim Ga\left(\frac{\delta+a_i}{2},\frac{\tau}{2}\right).\end{aligned}\tag{8.3}$$

where

$$Q_V=\begin{pmatrix}1&0&0\\ \vdots&\vdots&\vdots\\ 1&0&0\\ 0&1&0\\ \vdots&\vdots&\vdots\\ 0&1&0\\ 0&0&1\\ \vdots&\vdots&\vdots\\ 0&0&1\end{pmatrix}\text{ and }\boldsymbol{b}_i^V=\begin{pmatrix}b_{i1}^V\\ b_{i2}^V\\ b_{i3}^V\end{pmatrix}.\tag{8.4}$$

Note that $\boldsymbol{x}_i$ is the vector of length 50 consisting of the expression levels for gene i: first the 20 gene expression levels from samples taken at Clare, then 20 from Wingara and 10 from Willunga. Here, b_{i1}^V is the effect of the Clare vineyard on the expression level of gene i, b_{i2}^V is the effect of the Wingara vineyard on the expression level of gene i, and b_{i3}^V is the effect of the Willunga vineyard on gene i.

The highest-scoring Bayesian network obtained when the residual approach given this model is applied to the grape gene data set has 37 edges, and the corresponding moral graph has 68 edges. The moral graph is shown in Figure 8.7(a).

Temperature Effects Only

When only the effects of temperature on gene expression are considered, it is assumed that

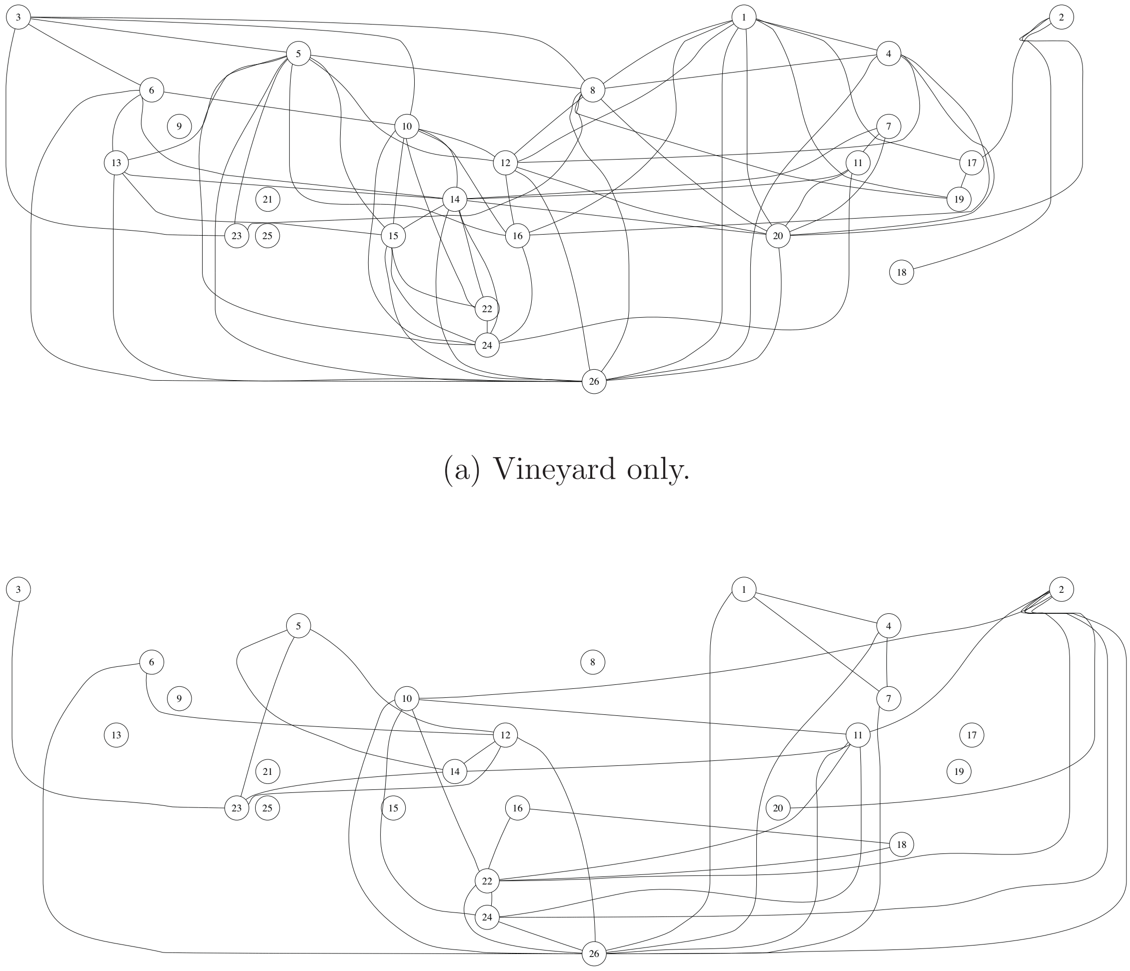
$$\begin{aligned} \boldsymbol{x}_i | \boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i^{\text{Temp}} &\sim N_{50}\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q_{\text{Temp}}\boldsymbol{b}_i^{\text{Temp}}, \psi_i I_{50} \right), \\ \boldsymbol{\gamma}_i | \psi_i &\sim N_{a_i}(\boldsymbol{0}, \tau^{-1}\psi_i I_{a_i}), \\ \psi_i^{-1} &\sim Ga\left(\frac{\delta + a_i}{2}, \frac{\tau}{2} \right). \end{aligned} \tag{8.5}$$

where

$$Q_{\text{Temp}} = \begin{pmatrix} \boldsymbol{q}_{1,1}^T \\ \vdots \\ \boldsymbol{q}_{1,20}^T \\ \boldsymbol{q}_{2,1}^T \\ \vdots \\ \boldsymbol{q}_{2,20}^T \\ \boldsymbol{q}_{3,1}^T \\ \vdots \\ \boldsymbol{q}_{3,10}^T \end{pmatrix} \text{ and } \boldsymbol{b}_i^{\text{Temp}} = \begin{pmatrix} b_{i1}^{\text{Temp}} \\ b_{i2}^{\text{Temp}} \\ \vdots \\ b_{i,23}^{\text{Temp}} \end{pmatrix}. \tag{8.6}$$

$\boldsymbol{q}_{kl}$ is the vector of length 20 of temperatures associated with sample l from vineyard k, where $k = 1$ codes for Clare, $k = 2$ for Wingara, and $k = 3$ for Willunga. Let q_{kl}^t be the temperature t minutes before the picking of sample l from vineyard k, $t \in \{30, 90, 150, 210, 270, 330\}$. Then $\boldsymbol{q}_{kl}$ is the following vector:

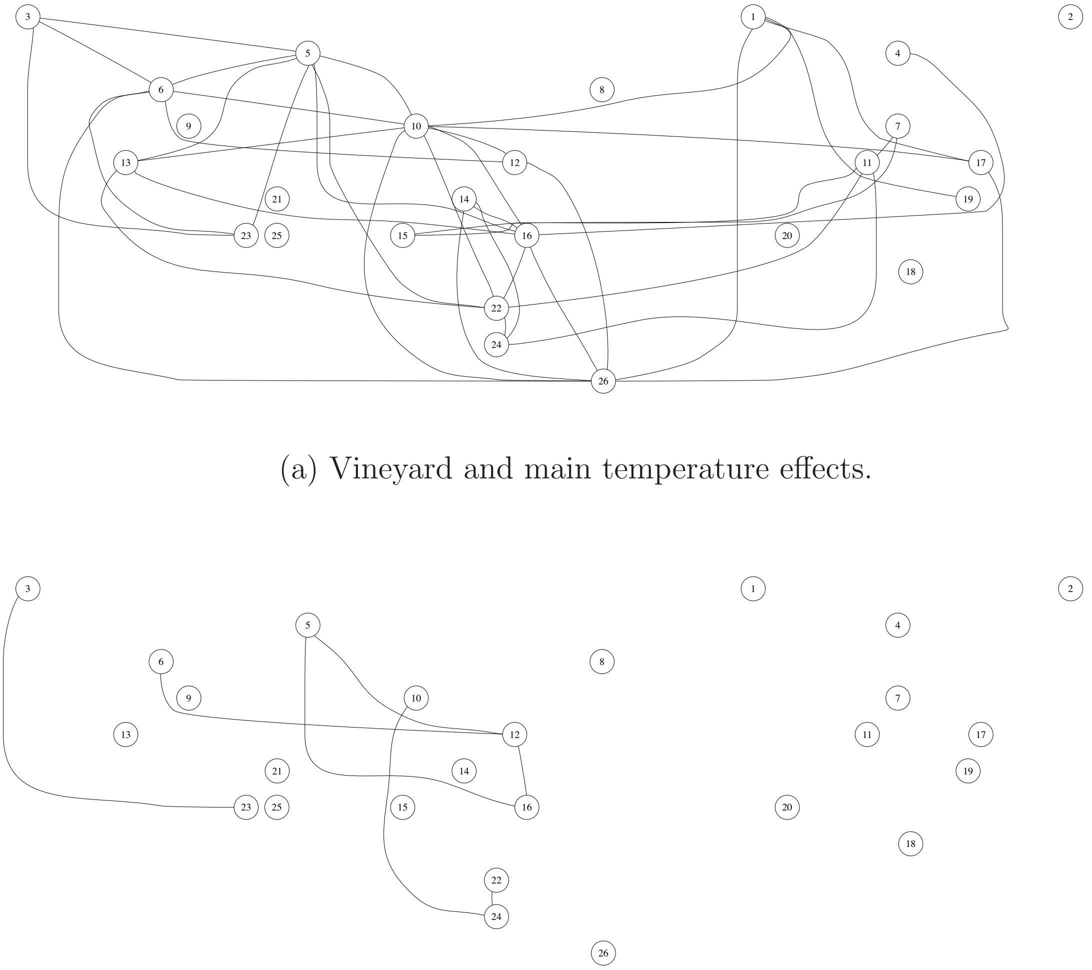
$$\boldsymbol{q}_{kl}^T = \left(q_{kl}^{30}, \ldots, q_{kl}^{330}, q_{kl}^{30} \times q_{kl}^{90}, q_{kl}^{30} \times q_{kl}^{150}, \ldots, q_{kl}^{210} \times q_{kl}^{330} \right).$$



(a) Vineyard only.

(b) Temperature only.

Figure 8.7: The moral graphs of the highest-scoring Bayesian networks found for the grape genes, when the residual approach is taken.



(a) Vineyard and main temperature effects.

(b) Vineyard and temperature (including temperature interactions).

Figure 8.8: The moral graphs of the highest-scoring Bayesian networks found for the grape genes, when the residual approach is taken.

For example, for the first sample taken from Clare,

$$\boldsymbol{q}_{11}^T = (23.70, 22.70, 22.10, 23.10, 23.10, 23.70, 537.99, 523.77, 547.47, 547.47, 561.69,$$
$$501.67, 524.37, 524.37, 537.99, 510.51, 510.51, 523.77, 533.61, 547.47).$$

Hence, b_{i1}^{Temp} is the effect of the temperature 30 minutes before the grape was picked on the expression level of gene i, and $b_{i,7}^{\text{Temp}}$ is the effect of the interaction between the temperatures 30 minutes and 90 minutes before the grape was picked on the expression level of gene i.

The highest-scoring Bayesian network obtained via this approach has 18 edges. The corresponding moral graph has 36 edges, and is displayed in Figure 8.7(b).

Vineyard and Main Temperature Effects

When vineyard and main temperature effects are included in the model for the expression levels of the genes, the following model is assumed:

$$\begin{aligned}
\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i^{\text{VT}} &\sim N_{50}\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q_{\text{VT}}\boldsymbol{b}_i^{\text{VT}}, \psi_i I_{50}\right), \\
\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\boldsymbol{0}, \tau^{-1}\psi_i I_{a_i}), \\
\psi_i^{-1} &\sim Ga\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right).
\end{aligned}$$

where the first three elements of $\boldsymbol{b}_i^{\text{VT}}$ are the same as $\boldsymbol{b}_i^V$, and the last six elements are $b_{i1}^{\text{Temp}}, b_{i2}^{\text{Temp}}, \ldots, b_{i6}^{\text{Temp}}$. Similarly, Q_{VT} consists of the matrix Q_V and the first six columns of Q_{Temp}.

The highest scoring Bayesian network obtained via this approach has 23 edges, and the corresponding moral graph, displayed in Figure 8.8(a), has 41 edges.

Vineyard and Temperature Effects

When both vineyard and temperature effects are included, the following model for the expression levels of each gene is assumed:

$$\begin{aligned}
\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i &\sim N_{50}\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q\boldsymbol{b}_i, \psi_i I_{50}\right), \\
\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I_{a_i}), \\
\psi_i^{-1} &\sim Ga\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right).
\end{aligned} \tag{8.7}$$

where

$$Q = \left(Q_V, Q_{\text{Temp}}\right) \text{ and } \boldsymbol{b}_i = \begin{pmatrix} \boldsymbol{b}_i^V \\ \boldsymbol{b}_i^{\text{Temp}} \end{pmatrix}. \tag{8.8}$$

The highest-scoring Bayesian network obtained when both vineyard and temperature effects are accounted for has 6 edges. The corresponding moral graph has 7 edges, and is displayed in Figure 8.8(b).

8.3.3 Using the S_2 score metric to Estimate a Bayesian Network for the Genes

When vineyard and temperature effects are included as random effects in the estimation of Bayesian networks for the grape genes, the S_2 score metric is used in conjunction with the High-dimensional Bayesian Covariance Selection algorithm to estimate a Bayesian network for the genes. In this section, only the model containing both vineyard and temperature effects will be considered.

The model in Equation (8.7) becomes

$$\begin{aligned}
\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i, \phi_i &\sim N_{50}\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q\boldsymbol{b}_i, \psi_i I_{50}\right), \\
\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\mathbf{0}, \tau^{-1}\psi_i I_{a_i}), \\
\psi_i^{-1} &\sim Ga\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right), \\
\boldsymbol{b}_i|\psi_i &\sim N_{23}\left(\mathbf{0}, \upsilon^{-1}\psi_i I_{23}\right).
\end{aligned} \tag{8.9}$$

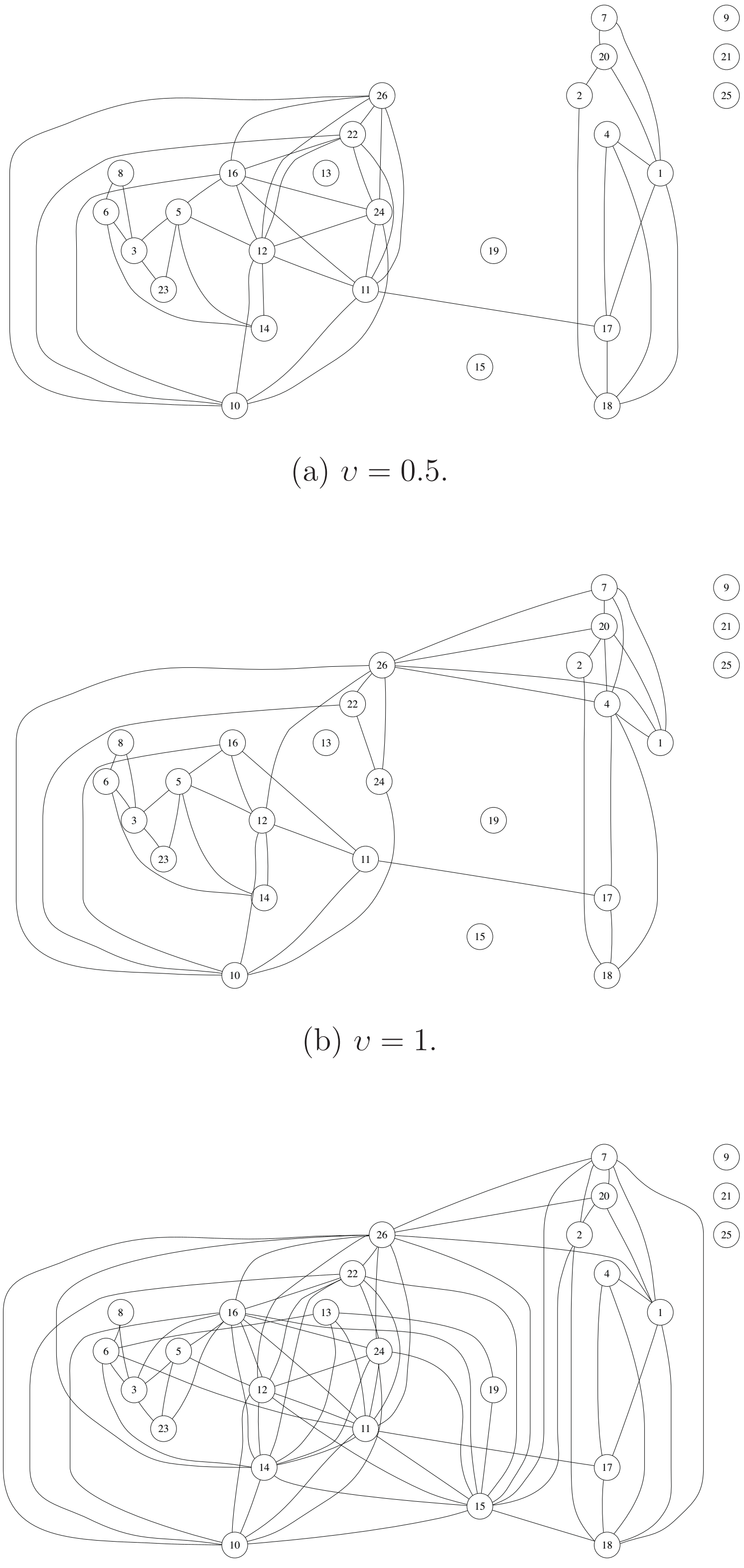
Note that it is for convenience that the prior distribution on the random effects $\boldsymbol{b}_i$ is selected. As discussed in Section 8.3.1, this prior distribution is a poor approximation to the true distribution of the vineyard and temperature effects. However, it is this prior distribution that allows the use of the S_2 score metric.

Bayesian networks are estimated for $\upsilon = 0.5, 1, 10$. The moral graphs of the highest-scoring Bayesian networks found for each of these values of υ are shown in Figure 8.9. The number of edges in the highest-scoring Bayesian networks and the corresponding moral graphs for each of these values of υ are summarised in the section of Table 8.2 headed by S_2.

Table 8.2 shows that as υ increases, so too does the number of edges in the highest-scoring Bayesian network, in accordance with the results obtained in Section 7.2.2. Figure 8.9 shows that the moral graphs have a similar structure for each value of υ assumed. These graphs are compared to those obtained through other methods in Section 8.4.

8.3.4 Using a Combination of S_2 and the Residual Approach in the Estimation of a Bayesian Network for the Grape Genes.

As discussed in Section 8.3.1, the assumption of independent and identically distributed temperature effects is probably not valid. Even if these temperature effects did satisfy such an assumption, it is almost certain that these effects are not distributed identically to the vineyard effects. In an attempt to improve the inclusion of vineyard and temperature effects in the model for the expression levels of the genes, a combination of the residual approach and the S_2 score metric are applied, in conjunction with High-dimensional Bayesian Covariance Selection algorithm, to the grape gene data. This approach corrects for the effect of temperature on the expression levels of the genes, and allows for the posterior estimation of the effect of vineyard on gene expression.



(a) $\upsilon = 0.5$.

(b) $\upsilon = 1$.

(c) $\upsilon = 10$.

Figure 8.9: The moral graphs of the highest-scoring Bayesian networks found for the grape genes when S_2 is used, for different values of υ.

The following model for the expression level of gene i is assumed:

$$\begin{aligned}
\boldsymbol{x}_i|\boldsymbol{x}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_i^V, \boldsymbol{b}_i^{\text{Temp}}, \phi_i &\sim N_{50}\left(\boldsymbol{x}_{P_i}\boldsymbol{\gamma}_i + Q_V\boldsymbol{b}_i^V + Q_{\text{Temp}}\boldsymbol{b}_i^{\text{Temp}}, \psi_i I_{50}\right), \\
\boldsymbol{\gamma}_i|\psi_i &\sim N_{a_i}(\boldsymbol{0}, \tau^{-1}\psi_i I_{a_i}), \\
\psi_i^{-1} &\sim Ga\left(\frac{\delta + a_i}{2}, \frac{\tau}{2}\right), \\
\boldsymbol{b}_i^V|\psi_i &\sim N_3\left(\boldsymbol{0}, \upsilon^{-1}\psi_i I_3\right), \qquad (8.10)
\end{aligned}$$

where, as previously, $\boldsymbol{b}_i^V$ is the vector of vineyard effects and $\boldsymbol{b}_i^{\text{Temp}}$ is the vector of temperature effects.

In this section, the temperature effects are considered to be nuisances, included in the model to improve the estimation of Bayesian networks for the data. The residual approach is used to remove the effects of temperature on gene expression levels. A $50 \times (50 - 20) = 50 \times 30$ matrix P is found such that

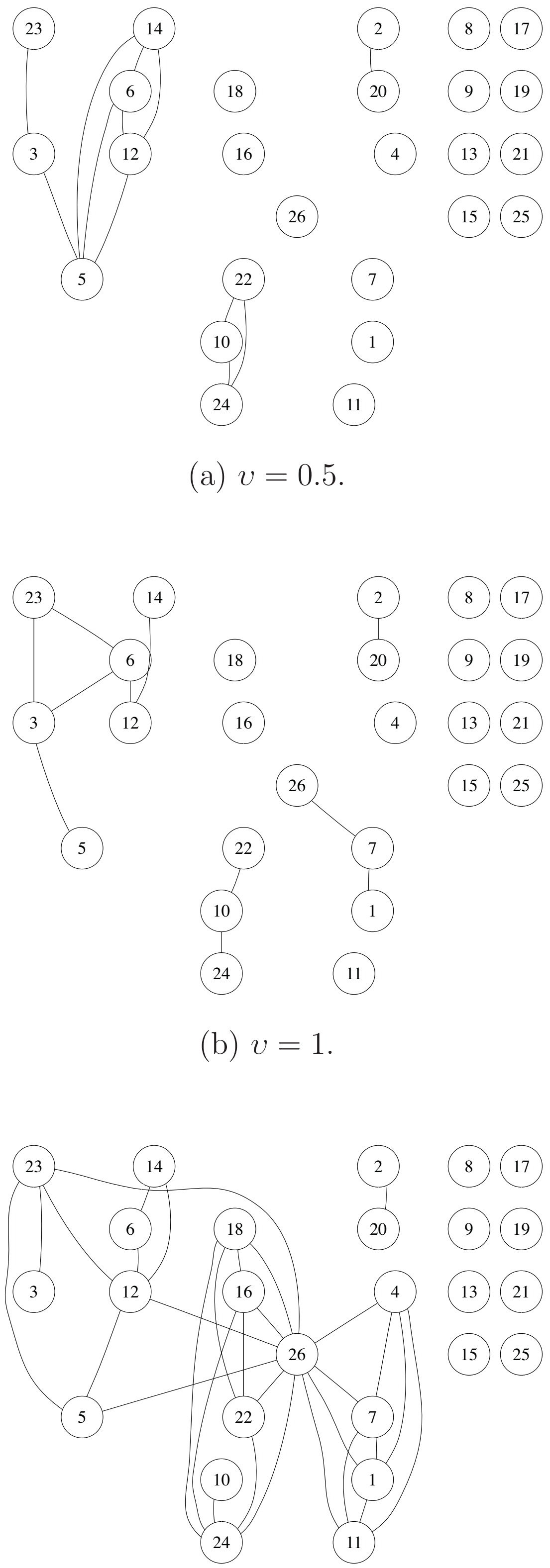
$$\begin{aligned}
P^T Q_{\text{Temp}} &= 0 \\
P^T P &= I_{30} \\
PP^T &= I_{50} - Q_{\text{Temp}}\left(Q_{\text{Temp}}^T Q_{\text{Temp}}\right)^{-1} Q_{\text{Temp}}^T.
\end{aligned}$$

Taking $\boldsymbol{y}_i = P^T\boldsymbol{x}_i$ implies that

$$\boldsymbol{y}_i|\boldsymbol{y}_{P_i}, \boldsymbol{\gamma}_i, \psi_i, \boldsymbol{b}_{V,i}, \phi_i \sim N_{30}\left(\boldsymbol{y}_{P_i}\boldsymbol{\gamma}_i + P^T Q_V \boldsymbol{b}_i^V, \psi_i I_{30}\right).$$

The S_2 score metric is then used in conjunction with the High-dimensional Bayesian Covariance Selection algorithm to estimate Bayesian networks for the grape genes, for $\upsilon = 0.5, 1, 10$. The moral graphs corresponding to the highest-scoring Bayesian networks found are displayed in Figure 8.10. The number of edges in the highest-scoring Bayesian networks and the corresponding moral graphs are summarised in the section of Table 8.2 headed by "S_2 and Residual".

As was observed when S_2 was used in the analysis of the grape gene data, Table 8.2 shows that as υ increases, so too does the number of edges in the highest-scoring Bayesian network. While there are similarities between the moral graph obtained for each value of υ, as displayed in Figure 8.10, as υ increases, so too does the degree of most genes. These graphs are compared to those obtained via the other methods in the next section.



(a) $\upsilon = 0.5$.

(b) $\upsilon = 1$.

(c) $\upsilon = 10$.

Figure 8.10: The moral graphs of the highest-scoring Bayesian networks found for the grape genes, when a combination of the residual approach and S_2 is used, for different values of υ.

8.4 The Highest-Scoring Graphs Obtained

Table 8.2: Number of edges in the highest-scoring Bayesian network and the corresponding moral graph for each of the four approaches taken.

	Ignoring Effects	Residual Approach				S_2			S_2 and Residual		
		Covariates Included				υ			υ		
		V'yard	Temp	V'yard & Main Temp	V'yard & Temp	0.5	1	10	0.5	1	10
B. N.	43	37	18	23	6	24	25	33	8	10	16
Moral	84	68	36	41	7	44	40	73	12	11	32

The moral graphs of the highest-scoring Bayesian networks for the grape genes found in Sections 8.2 and 8.3.2–8.3.4 are here considered. These graphs are compared, and the biological plausibility of highest-scoring graph obtained when the residual approach is used to remove the effects of vineyard and temperature is discussed.

Table 8.2 summarises the number of edges in the Bayesian networks and moral graphs obtained when the effects of vineyard and temperature are ignored and when the effects of vineyard or the effects of temperature, or both, are included in the estimation of the graph, using the residual approach, S_2 or a combination of S_2 and the residual approach. Figure 8.2 displays the moral graph obtained initially, when the effects of vineyard and temperature were ignored. Figure 8.7 displays the moral graphs obtained when the effects of temperature and vineyard are included individually in the model for gene expression, and Figure 8.8 displays the moral graphs obtained when vineyard and temperature are both included in the model. Figures 8.9 and 8.10 display the moral graphs obtained when S_2 and a combination of S_2 and the residual approach, respectively, are used in the analysis of the grape gene data, with $\upsilon = 0.5, 1, 10$.

As was expected, when the variation in the gene expression values due to changes in temperature and vineyard is accounted for, the Bayesian networks obtained are much sparser than that obtained when these sources of variation were ignored. The difference is most dramatic when the residual approach or a combination of the residual approach and S_2 is used to remove the effects of vineyard and temperature.

The table and the figures show that no matter which approach is taken, as more of the variation in the gene expression values is accounted for, the fewer edges there are in the highest-scoring Bayesian networks and corresponding moral graphs. Also apparent is that as fewer assumptions about the covariance structure of the effects of vineyard and temperature are made, the fewer edges there are in the highest-scoring Bayesian networks.

The results of the use of the residual approach indicate that not only do the main temperature effects need to be accounted for in the estimation of a Bayesian network for the genes, but so do the two-way temperature interactions.

Table 8.2 also shows that there are quite large differences between the highest-scoring graphs found when S_2 is used, and when a combination of S_2 and the residual approach is used. For all values of υ, far fewer edges are in the highest-scoring graphs obtained when a combination of S_2 and the residual approach is used, as opposed to when just S_2 is used. This is probably due to the fact that when S_2 is used, the covariance structure of the temperature effects is, without doubt, incorrectly specified, while no assumptions about this structure are made when the residual approach is used to remove temperature effects.

Note the similarity of the moral graphs obtained when the residual approach is used to remove the effects of vineyard and temperature and when a combination of the residual approach and S_2 is used, with $\upsilon = 0.5$ or $\upsilon = 1$. All three graphs have a connected component consisting only of genes with reference numbers 10, 22 and 24, and all three have an edge between gene number 3 and gene number 23. There are also edges involving genes 5, 6 and 12 in all three graphs.

Hence, the inclusion of temperature and vineyard effects in the estimation of Bayesian networks for the grape gene expression data results in much simpler networks than when such effects are ignored. However, it seems that if the covariance structure of such effects is vastly oversimplified, as it was when it was assumed the effects of temperature were independent and identically distributed, the graphs obtained are not substantially more simple than those obtained when such effects are ignored.

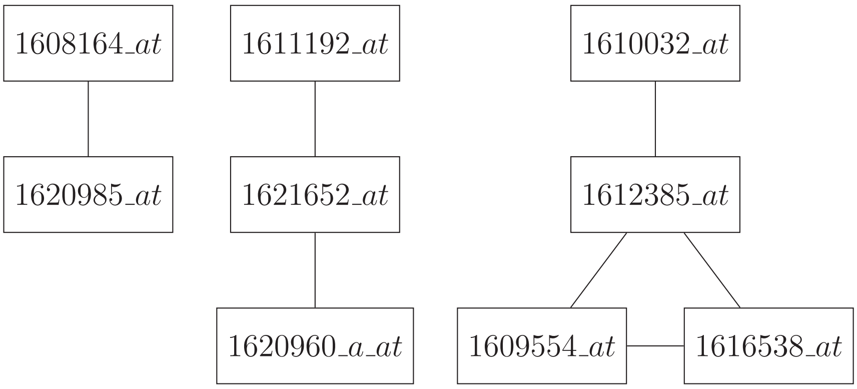


Figure 8.11: Connected components of Figure 8.8(b), with gene names included.

8.4.1 Biological Plausibility of the Graphs

The biological plausibility of these graphs is now discussed. These graphs represent conditional independence relationships between the expression levels of different genes, and any edges present may give information about biological association.

Given the relationship of the expression levels of the grape genes to vineyard and temperature, it is thought that most of the edges in the graphs obtained when these effects are not accounted for are spurious; not due to intrinsic biological association between genes, but rather included because a complex mean structure was not incorporated in the model for the expression levels of the genes.

Similarly, when the covariance structure of the effects of vineyard and temperature is oversimplified, as it probably is when it is assumed that these effects are independent and identically distributed, most of the edges in the highest-scoring moral graphs are thought to be spurious. This is also the case when the variance of the effects of vineyard and/or temperature is thought to be larger than the marginal variance of the data.

It is difficult to determine whether or not the edges in the moral graphs obtained when temperature and vineyard effects are included in the model for gene expression have any biological significance. Figure 8.11 displays the connected components of the moral graph obtained when the residual approach is used, with the Affymetrix gene names included. This moral graph has the smallest number of edges out of all the moral graphs obtained.

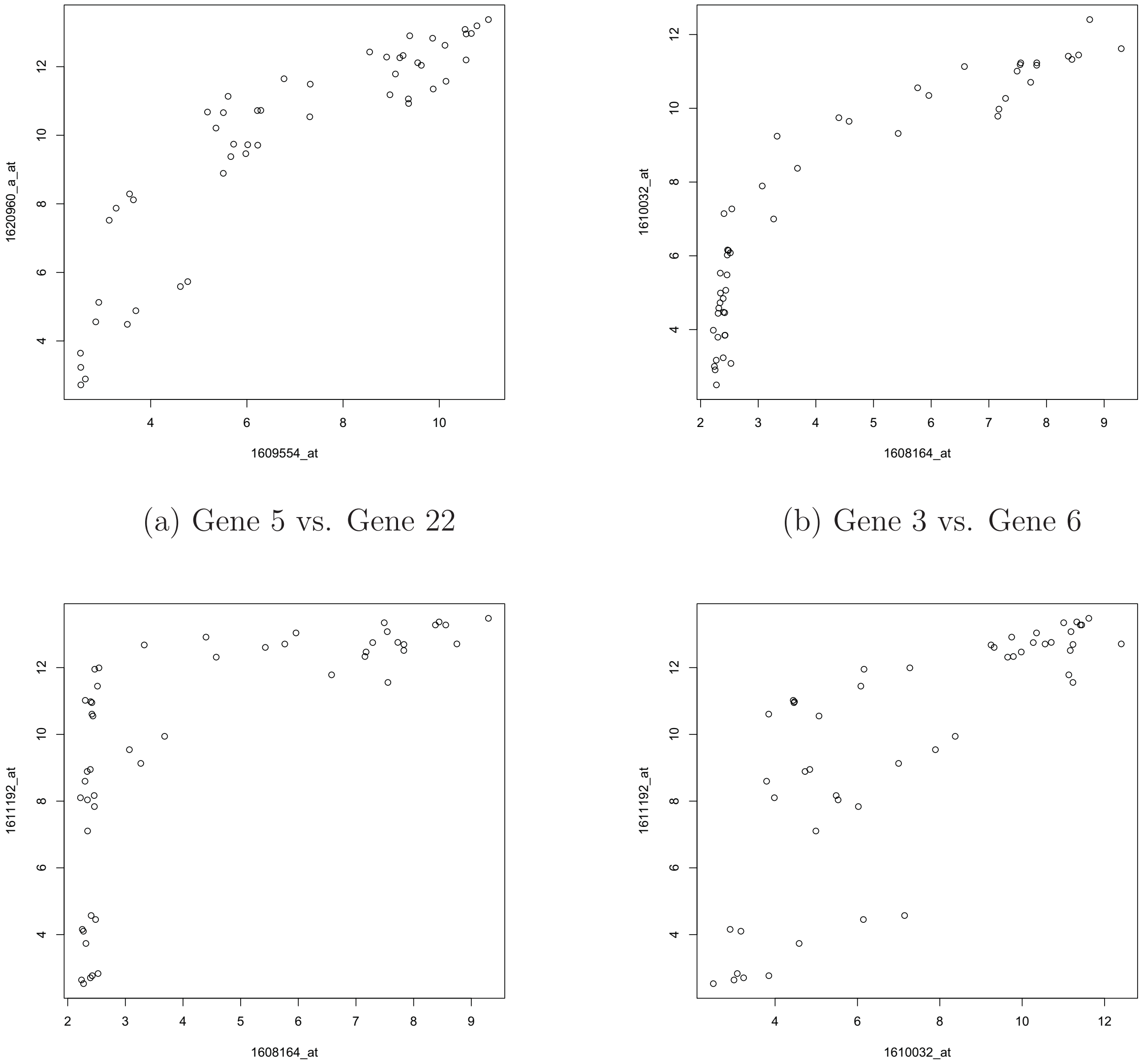
There are two things worth noting about this graph. The first is that all of the genes in the connected components code for proteins of the small Hsp group (see Table 8.1). Small Hsps appear to be responsible for binding to proteins which have been denatured by heat, preventing them from unfolding further, and clumping within the cell, [86, 87]. This may indicate that these edges represent true biological association between pairs of genes.

The second point to be made about this graph is that there are two examples where probe sets that are actually coding for the same gene are in different connected components. For instance, probe sets with Affy numbers 1608164*_at*, 1610032*_at*, and 1611192*_at* are actually three sections of the gene with NCBI reference Vvi.6787. These genes appear in different connected components. The same is true for the probe sets with Affy numbers 1609554*_at* and 1620960*_a_at*, which represent different sections of the gene with NCBI reference Vvi.7044. It would be hoped that different probe sets that code for the same gene would have an association. This result leads to the hypothesis that these edges are simply due to noise in the data.

This hypothesis is supported by the amount of variation in the gene expression levels that is explained by vineyard and temperature, and the plots presented in Figure 8.12. For example, in Figure 8.12(a), the expression levels of the probe sets with Affy numbers 1609554*_at*, known here as gene 5, and 1620960*_a_at*, known as gene 22, are plotted against one another. While these probes are ostensibly probing for the same gene, the expression levels obtained are not exactly the same, as evidenced by this scatterplot. The other scatterplots in the figure indicate the same thing for the other pairs of genes. Note that linear relationships between the expression levels of gene pairs are required for edges between these genes to be found. The fact that the relationships between the expression levels for these gene pairs are not linear could be an explanation as to why edges between these genes were not found.

8.5 Posterior Estimation of Vineyard Effects

In this section, posterior samples of parameters are simulated for the parameters associated with the Bayesian networks obtained in Section 8.3.4. Recall that three Bayesian networks were estimated in that section; one each assuming $\upsilon = 0.5, 1, 10$. Table 8.3 gives the parents



(a) Gene 5 vs. Gene 22

(b) Gene 3 vs. Gene 6

(c) Gene 3 vs. Gene 10

(d) Gene 6 vs. Gene 10

Figure 8.12: Scatterplots of the expression levels of some of the probes coding for the same genes.

of each gene in each of these three Bayesian networks, and the corresponding moral graphs are displayed in Figure 8.10. Note that posterior inference for the graphs obtained in Section 8.3.3 is not considered here, due to the (most likely) invalid assumption that the effects of temperature are independent and identically distributed.

Given that the residual approach was used to remove the variation in the expression levels of the genes associated with temperature, posterior estimates of the effect of temperature on these expression levels is unavailable. However, posterior estimates of the effect of each of the vineyards on the gene expression levels are available, as are posterior estimates of the ψ_is and γ_{ij}s.

Table 8.3: Parents of each gene in the highest-scoring Bayesian network found using a combination of S_2 and the residual approach, for $\upsilon = 0.5, 1, 10$.

υ	Gene Number												
	1	2	3	4	5	6	7	8	9	10	11	12	13
0.5			23		3					22,24		5,6,14	
1	7		6,23		3	12	26			24		14	
10		20			12,23,26							6,14	

υ	Gene Number												
	14	15	16	17	18	19	20	21	22	23	24	25	26
0.5							2						
1							2		10				
10									16,18,24,26	3	10		1,4,7,11

Gibbs sampling, as described in Section 4.4.5, is conducted for each of the three Bayesian networks considered. For each of these three Bayesian networks, samples of size 50000 are simulated from the joint posterior distributions of $f(\boldsymbol{\gamma}_i, \boldsymbol{b}_i, \psi_i | \boldsymbol{x}_i, \boldsymbol{x}_{P_i})$ for $i = 1, 2, \ldots, 26$, with the first 10000 samples for each i discarded as burn-in.

The generation of posterior samples for $\upsilon = 0.5, 1, 10$ allows an investigation into the effect of υ on posterior inference. Recall that υ relates the variance of the vineyard effects of gene i to the variance of the expression levels of gene i. When it is assumed that $\upsilon = 1$, $\phi_i = \psi_i$

for each gene. That is, the variance of the effects of vineyard on each gene is the same as the variance of the expression levels of that gene given these vineyard effects. When it is assumed that $\upsilon = 0.5$, $\phi_i = 2\psi_i$, and when it is assumed that $\upsilon = 10$, $\phi_i = 0.1\psi_i$.

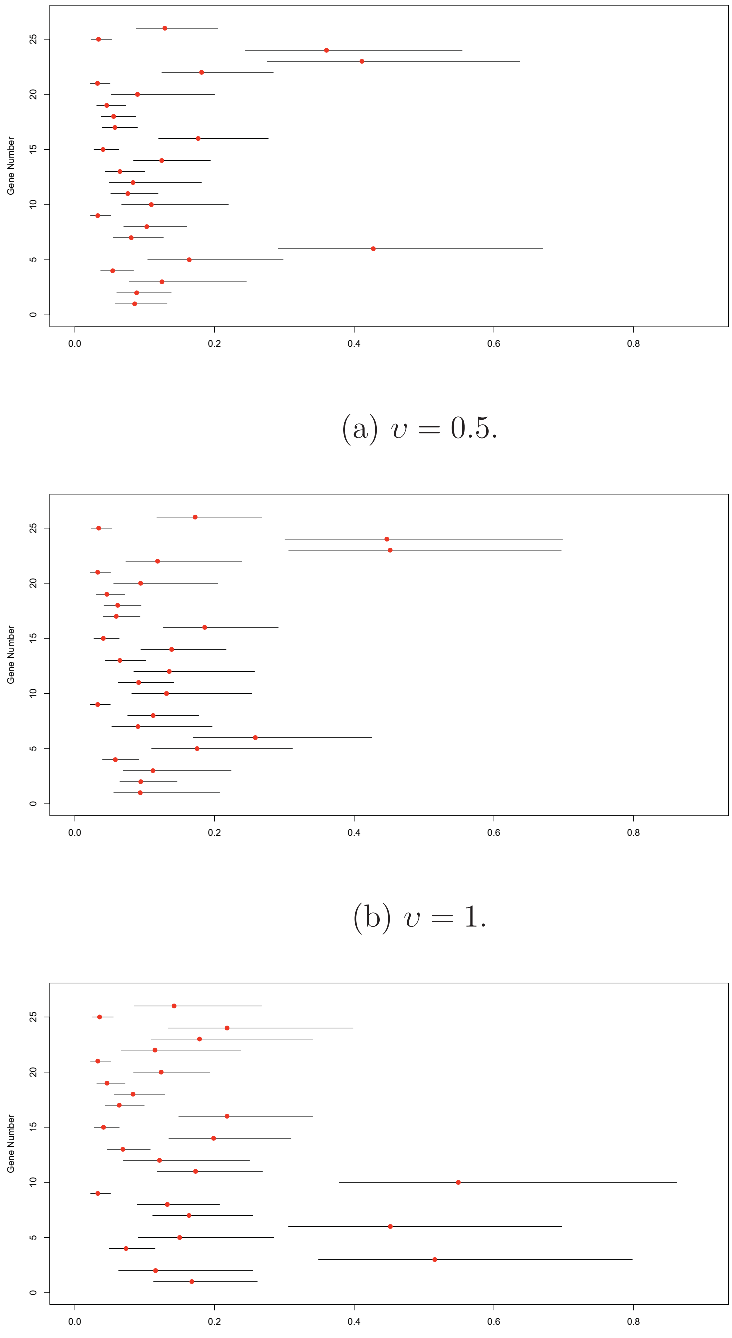
Figure 8.13 displays the 90% posterior intervals and medians for ψ_i, $i = 1, 2, \ldots, 26$, obtained given each of $\upsilon = 0.5, 1, 10$. For the genes that have the same parent set in the Bayesian networks obtained for each value of υ, the posterior intervals are quite similar for each value of υ. For those genes with different parent sets in the different Bayesian networks, these posterior intervals can vary markedly for the different values of υ. For example, gene 3 has gene 23 as a parent in the Bayesian network obtained when $\upsilon = 0.5$, has genes 6 and 23 as parents when $\upsilon = 1$, and has no parents when $\upsilon = 10$. The 90% posterior intervals for ψ_3 are quite similar for $\upsilon = 0.5$ or 1, but the interval obtained when $\upsilon = 10$ is quite different from both of these.

The posterior samples of the vineyard effects are now discussed. Figure 8.14 displays the 90% posterior intervals and posterior medians for b^V_{i1}, the effect of the Clare vineyard on the expression level of gene i, $i = 1, 2, \ldots, 26$. Figure 8.15 displays 90% posterior intervals and medians for b^V_{i2}, the effect of the Wingara vineyard on the gene expression levels. Figure 8.16 displays the 90% posterior intervals and medians for b^V_{i3}, the effect of the Willunga vineyard.

Most notable is that when $\upsilon = 10$, the posterior intervals for b^V_{i1}, b^V_{i2} and b^V_{i3} are generally much narrower than those obtained when $\upsilon = 0.5$ or 1, and the medians tend to be closer to zero. These differences are to be expected, given that assuming $\upsilon = 10$ implies much less variable vineyard effects than assuming $\upsilon = 0.5$ or 1.

For many genes, the posterior intervals for b^V_{i1}, b^V_{i2} and b^V_{i3} are quite similar for $\upsilon = 0.5$ or 1. Differences arise when the parent set of a gene in the Bayesian network estimated for $\upsilon = 0.5$ is different to that in the Bayesian network estimated for $\upsilon = 1$.

For each of b^V_{i1}, b^V_{i2} and b^V_{i3}, $i = 1, 2, \ldots, 26$, most of the posterior intervals contain zero. Despite this, vineyard has been shown to be an important factor influencing the expression levels of genes, both in terms of the regressions discussed in Section 8.3, and in terms of the graphs obtained. This disparity is due to the collinearity of the temperature terms in the

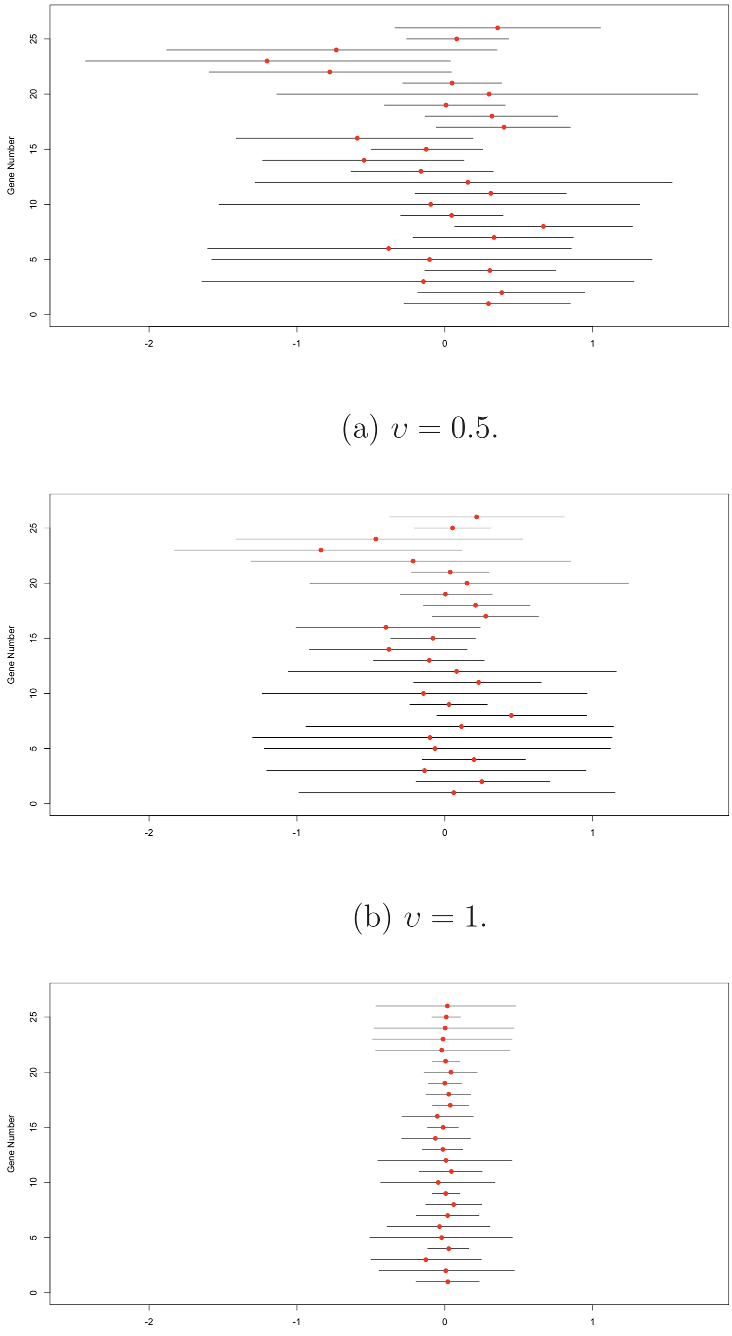


(a) $\upsilon = 0.5$.

(b) $\upsilon = 1$.

(c) $\upsilon = 10$.

Figure 8.13: 90% posterior intervals for ψ_i, $i = 1, 2, \ldots, 26$, generated given the Bayesian networks found assuming $\upsilon = 0.5, 1, 10$. The posterior medians are displayed in red.

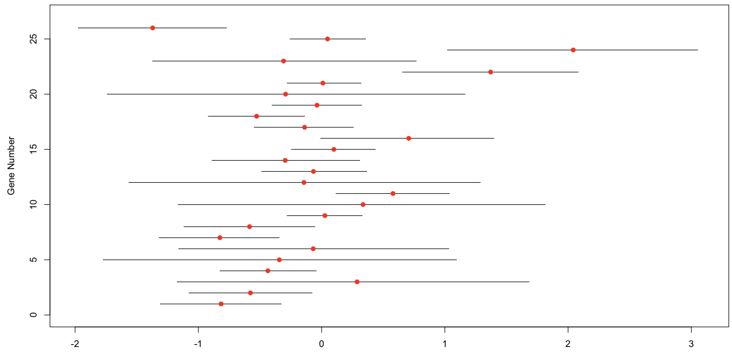


(a) $\upsilon = 0.5$.

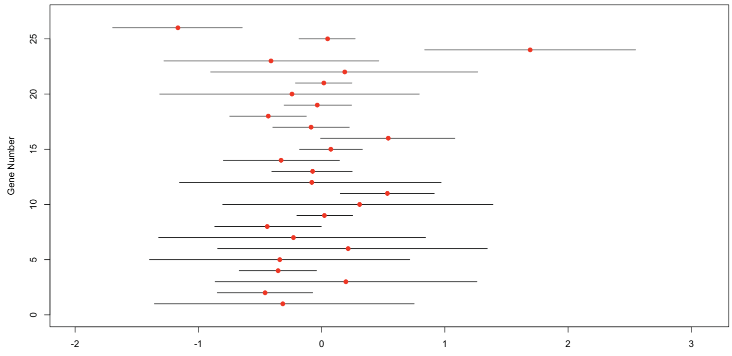
(b) $\upsilon = 1$.

(c) $\upsilon = 10$.

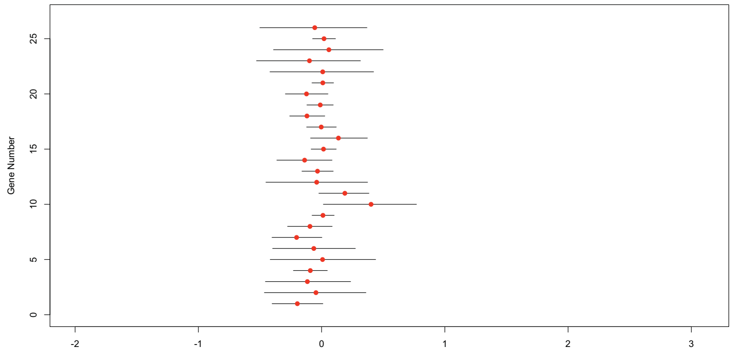
Figure 8.14: 90% posterior intervals and medians (in red) for b^V_{i1}, $i = 1, 2, \ldots, 26$, the effect of the Clare vineyard on the expression level of gene i.



(a) $\upsilon = 0.5$.



(b) $\upsilon = 1$.



(c) $\upsilon = 10$.

Figure 8.15: 90% posterior intervals and medians (in red) for b_{i2}^{V}, $i = 1, 2, \ldots, 26$, the effect of the Wingara vineyard on the expression level of gene i.

regressions.

8.6 Conclusions

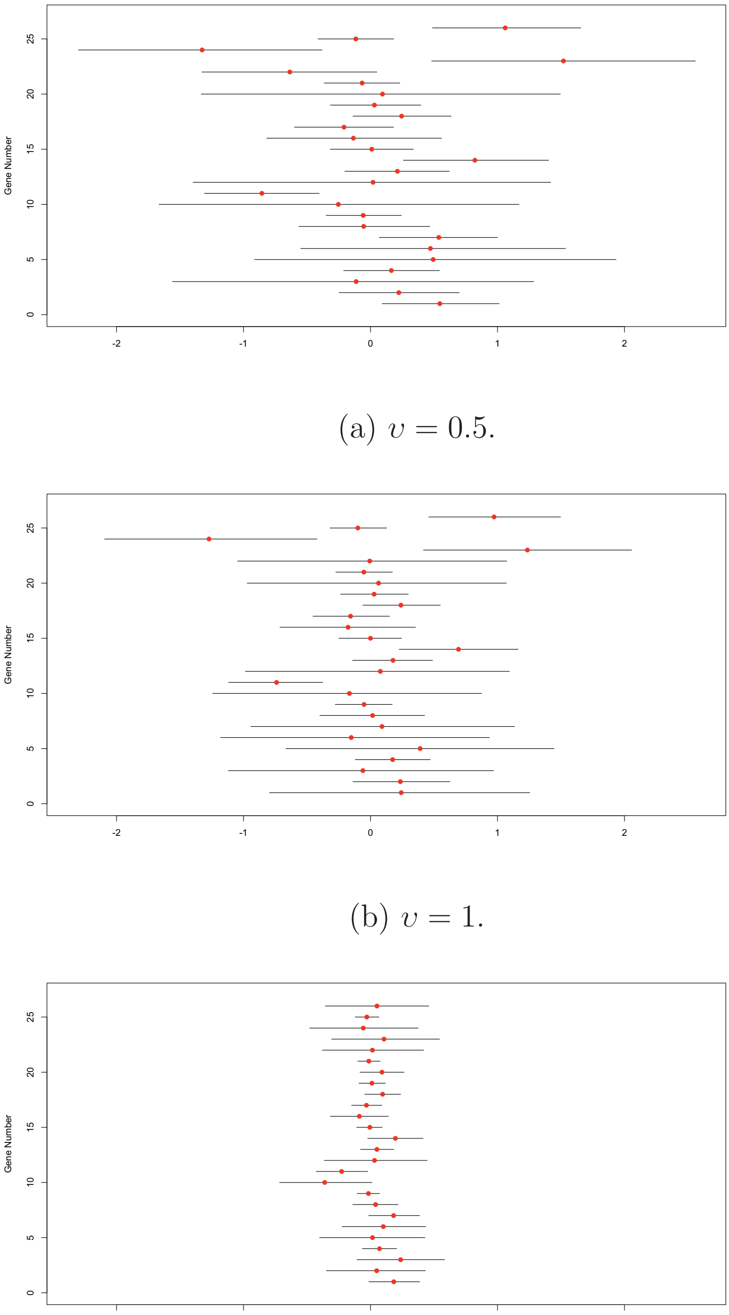
In this chapter, the utility of the score metrics developed in Chapter 4, and the residual approach developed in Chapter 6, in the estimation of Bayesian networks for a real data set was demonstrated. The data set comprised expression levels for 26 different grape heat-shock genes, where grape berry samples were drawn from three different vineyards. Temperatures leading up to the time of the picking of the grape berries were recorded.

Initially, the possible effects of vineyard and temperature on the expression levels of the genes were ignored, and the expression levels of each of the genes were treated as independent and identically distributed samples. Under this assumption, the High-dimensional Bayesian Covariance Selection algorithm was used to estimate a Bayesian network for the grape genes. This network contained 43 edges, and the corresponding moral graph contained 84 edges.

The effects of vineyard and temperature on the expression levels of each of the genes were then considered. Given that the genes considered here are known to code for heat-shock proteins, it is well known that the expression levels of these genes are strongly influenced by temperature. It was also shown that the mean expression levels of many of the genes were significantly different when grape berry samples were taken from different vineyards. As such, it seems that the assumption of independent and identically distributed samples of gene expression levels is not valid.

In Sections 8.3.2–8.3.4, the residual approach and the S_2 score metric were variously used to account for the variation in the expression levels due to temperature and vineyard in the estimation of Bayesian networks for the genes. When the residual approach was used to remove the effects of temperature and vineyard on the gene expression levels, the highest-scoring Bayesian network found had 6 edges, and the corresponding moral graph had 7 edges.

Hence, as was expected, when the variation in the gene expression levels that was due to the effects vineyard and temperature was accounted for in the estimation of a Bayesian network



(a) $\upsilon = 0.5$.

(b) $\upsilon = 1$.

(c) $\upsilon = 10$.

Figure 8.16: 90% posterior intervals and medians (in red) for b_{i3}^{V}, $i = 1, 2, \ldots, 26$, the effect of the Willunga vineyard on the expression level of gene i.

for the genes, the highest-scoring Bayesian network was much sparser than that obtained when such effects were not accounted for.

The graphs obtained when a combination of the residual approach and S_2 was used to control for the effects of vineyard and temperature were then used in the simulation of posterior samples from $f\left(\boldsymbol{\gamma}_i, \boldsymbol{b}_i^V, \psi_i\right)$, $i = 1, 2, \ldots, 26$. These posterior samples did not display any obvious patterns, and most of the 90% posterior intervals for the vineyard effects contained zero.

This analysis has shown that the covariances observed between the expression levels of different genes can be explained by temperature and vineyard. When the effects of temperature and vineyard are accounted for, the remaining variation in the expression levels of the genes are due to noise.

Chapter 9

Conclusions and Future Work

In this thesis, novel score metrics for the estimation of Bayesian networks for data sets that do not consist of independent and identically distributed samples were developed. Through the application of these score metrics to data sets with known and unknown network structure, the score metrics were found to be very useful in the estimation of Bayesian networks for data sets with complex mean structures.

Existing approaches to the estimation of Bayesian networks were discussed, with emphasis on score-based methods. It was noted that all of these approaches assume the data consists of independent and identically distributed samples. As such, these approaches were found to be inadequate in the estimation of a Bayesian network given a data set with a complex mean structure, or additional components of variance.

New score metrics that take account of the complex mean structure of a data set through a suitable Bayesian model were then developed. These metrics were labelled S_1, S_2, S_3 and S_4, and can be used in conjunction with any score-based method for the estimation of Bayesian networks. Here, these score metrics were implemented in conjunction with the High-dimensional Bayesian Covariance Selection algorithm.

These new score metrics S_1–S_4 assume that the random effects that make up the additional complexities in the mean structure of the data set being analysed are independent and identically distributed. In the analysis of real data sets, this assumption may not be particularly satisfactory, so the consequences of assuming dependence and unequal variances of the random effects on the forms of the score metrics was explored.

A residual approach was then developed, allowing the estimation of Bayesian networks for data sets with a complex mean structure without the need to specify the distribution of the random effects that constitute the mean structure. It was noted that this approach is most useful when the effect of variables that make up the mean structure of a data set are not of primary interest, but are secondary to the estimation of a Bayesian network given the data set.

The residual approach and the score metrics S_1, S_2 and S_3 were applied to simulated data sets with a complex mean structure. The results of these simulation studies showed the need to account for the mean structure in the estimation of a Bayesian network for a data set, and demonstrated the utility of the approaches developed herein. Further, it was found that when the mean structure of a data set was accounted for, the highest-scoring networks found were quite robust to changes in the score metric used.

The analysis of a gene expression data set with a complex mean structure illustrated the effectiveness of the approaches developed herein in the analysis of real data sets. It was shown that when the complex mean structure of this data set was accounted for, the resulting Bayesian network was more biologically plausible than the network obtained when such effects were ignored.

The approaches developed in this thesis increase the applicability of score-based methods for the estimation of Bayesian networks, allowing the estimation of networks for data sets that have complex mean structure. These approaches allow the variation in gene expression values that is due to external factors, such as temperature, to be accounted for in the estimation of Bayesian networks. Hence, the high-scoring Bayesian networks found will represent the conditional independence relationships inherent in any underlying genetic regulatory network more accurately.

While these approaches have been shown to be effective in the analysis of real gene expression data sets, further work remains to be done. A deeper understanding of the implications of the misspecification of the prior distribution of the random effects on high-scoring Bayesian networks is required, as is an understanding of the implications of the misspecification of the complex mean structure of a data set. Methods for the inclusion of prior information about the structure of any underlying genetic regulatory network also require investigation.

One shortcoming of the use of Bayesian networks in the analysis of gene expression data is that they fail to capture the dynamic nature of genetic regulatory networks. Genetic regulatory networks do not remain constant over time, but instead change depending upon many different conditions, both extra- and intra-cellular. The approaches developed above assume that, once such complexities have been accounted for, the relationships between genes are constant for all samples of expression levels. Dynamic Bayesian networks provide a method for the modelling of such dynamic relationships, and it seems logical that the work presented here be extended to the estimation of dynamic Bayesian networks. However, it must be noted that the associations dynamic Bayesian networks are able to detect depend upon the time-scale on which the reactions that make up a genetic regulatory network occur, a time-scale which is typically incompatible with experimentation.

More generally, it is important to make the techniques developed above accessible to biologists. It is also important to further the understanding of how these techniques may be used to investigate genetic regulatory networks. How Bayesian networks estimated on the basis of gene expression data can be used to shed light on genetic regulatory networks is a difficult problem, and one that must be approached in collaboration with biologists.

Appendix A

Gaussian Quadrature

This appendix provides a brief review of Gaussian quadrature; one class of methods for the numerical integration of functions. For a comprehensive account of numerical integration theory, the interested reader is directed to [15], and for a statistical perspective on the approximation of integrals, the reader is directed to [30]. A brief account of numerical integration, again from a statistical perspective, is given in Chapter 5 of [79], and Chapter 4 of [65] provides C++ code for many numerical integration routines.

Numerical integration involves the approximation of integrals such as

$$\int_a^b f(x)dx, \tag{A.1}$$

where $f : [a, b] \to \mathbb{R}$. Often, the integral will be approximated by some linear combination of values of the integrand:

$$\int_a^b f(x)dx \approx w_1 f(x_1) + w_2 f(x_2) + \ldots + w_q f(x_q),$$

$x_i \in [a, b]$, $w_i \in \mathbb{R}$, $\forall i$, where the w_i are known as the weights, and the x_i known as the abscissae of the numerical integration. The above equation uses q weights and abscissae, and is hence said to be a q-point numerical integration. There are many different methods for the choice of the weights and abscissae, and Gaussian quadrature is one such method, with some particularly nice properties, of selecting such quantities.

For many numerical integration methods, the abscissae consist of equally-spaced points on the region of integration. That is, the abscissae are specified in advance. Given such abscissae

$\{x_1, x_2, \ldots, x_q\}$, weights $\{w_1, w_2, \ldots, w_q\}$ may be found such that for any polynomial $f(x)$ of degree less than q,

$$\int_a^b f(x)dx = w_1 f(x_1) + w_2 f(x_2) + \ldots + w_q f(x_q),$$

a proof of which is given in [37].

Often, instead of considering an integral of the form in Equation (A.1), it is convenient to consider an integral such as

$$\int_a^b W(x)f(x)dx,$$

where $W(x)$ is a weight function, distinct from the weights w_i, and is called admissible if all of the following conditions hold over the (possibly infinite) region $[a, b]$:

- $W(x) \geq 0,\ x \in [a, b]$;
- $\int_a^b W(x)dx > 0$;
- $\int_a^b W(x)x^k dx < \infty,\ k = 0, 1, \ldots$

Before Gaussian quadrature can be discussed, some properties of orthogonal polynomials must be reviewed. First, define the inner product of functions $f(x)$ and $g(x)$ over A with respect to $W(x)$ to be

$$(f, g) = \int_a^b W(x)f(x)g(x)dx.$$

$f(x)$ and $g(x)$ are said to be orthogonal over $[a, b]$ with respect to $W(x)$ if $(f, g) = 0$. This inner product induces a norm:

$$\|f\| = \sqrt{(f, f)}. \tag{A.2}$$

Using this inner product and norm, the Gram-Schmidt process can be applied to the sequence of polynomials $1, x, x^2, \ldots$ to give a sequence of orthonormal polynomials $p_i(x)$:

$$p_i(x) = \frac{x^i - \sum_{j=0}^{i-1}(x^i, p_j)p_j(x)}{\left\| x^i - \sum_{j=0}^{i-1}(x^i, p_j)p_j(x) \right\|}$$

Often, the unnormalised orthogonal polynomials

$$P_i(x) = x^i - \sum_{j=0}^{i-1}(x^i, p_j)p_j(x) = p_i(x)\left\|x^i - \sum_{j=0}^{i-1}(x^i, p_j)p_j(x)\right\| \tag{A.3}$$

will be considered.

The following theorem allows for an easier computation of these orthogonal polynomials:

Theorem A.1. *The orthogonal polynomials $P_i(x)$ satisfy the following three-term recurrence relationship:*

$$P_{i+1}(x) = (x - a_{i+1})P_i(x) - b_{i+1}P_{i-1}(x),$$

where

$$a_{i+1} = \frac{(xP_i(x), P_i(x))}{\|P_i(x)\|^2}, b_{i+1} = \frac{(P_i(x), P_i(x))}{\|P_{i-1}(x)\|^2}.$$

Proof. See [37] for a proof. □

This theorem is important for Gaussian quadrature as the abscissae $x_1, x_2, \ldots x_q$ for q-point Gaussian quadrature associated with $W(x)$ are the roots of the polynomial $P_q(x)$. Hence, the following theorem about these roots is required:

Theorem A.2. *For $q \geq 1$, the polynomial $P_q(x)$ has q distinct real roots, $\{x_1, x_2, \ldots, x_q\}$, located in the interior of $[a, b]$.*

Proof. See [30]. □

Given this information about orthogonal polynomials, Gaussian quadrature can be defined.

Definition A.1. *The q-point Gaussian quadrature with respect to the weight function $W(x)$ has as its abscissae the roots $\{x_1, x_2, \ldots, x_q\}$ of $P_q(x)$, and has as its weights $\{w_1, w_2, \ldots, w_q\}$, where*

$$w_i = \int_a^b \frac{P_q(x)}{(x - x_i)P_q'(x_i)}W(x)dx. \tag{A.4}$$

That is, $\int_a^b W(x)f(x)dx$ can be approximated by

$$\int_a^b W(x)f(x)dx \approx \sum_{i=1}^q w_i f(x_i).$$

We then have the following theorem:

Theorem A.3. *Let $W(x)$ be an admissible weight function on $[a,b]$, and let $\{x_1, x_2, \ldots, x_q\}$ be the zeros of $P_q(x)$ given this weight function. Then the weights $\{w_1, w_2, \ldots, w_q\}$ can be calculated as in (A.4) such that for $f(x)$ a polynomial of degree at most $2q-1$,*

$$\int_a^b W(x)f(x)dx = \sum_{i=1}^q w_i f(x_i).$$

Proof. See [30]. □

The following theorem provides a justification for the use of Gaussian quadrature.

Theorem A.4. *For the q-point Gaussian quadrature with respect to the weight function $W(x)$, the following holds for continuous functions $f(x)$:*

$$\lim_{q\to\infty} \sum_{i=1}^q w_i f(x_i) = \int_a^b W(x)f(x)dx.$$

Proof. This proof is taken from [30].

By the Weierstrass approximation theorem, for all $\epsilon > 0$, there exists a polynomial $g(x)$ such that $|f(x) - g(x)| < \epsilon/2$ $\forall x \in [a,b]$.

Since $g(x)$ is a polynomial, $\exists q_0$ such that $\forall q \geq q_0$

$$\sum_{i=1}^q w_i g(x_i) = \int_a^b W(x)g(x)dx.$$

Then

$$\begin{aligned}\left|\int_a^b W(x)f(x)dx - \sum_{i=1}^q w_i f(x_i)\right| &\leq \left|\int_a^b W(x)f(x)dx - \int_a^b W(x)g(x)dx\right| \\ &\quad + \left|\int_a^b W(x)g(x)dx - \sum_{i=1}^q w_i g(x_i)\right| + \left|\sum_{i=1}^q w_i g(x_i) - \sum_{i=1}^q w_i f(x_i)\right| \\ &\leq \frac{\epsilon}{2} + 0 + \frac{\epsilon}{2} = \epsilon.\end{aligned}$$

Hence, the result is proved. □

In the present work, numerical integration over two dimensions is often required. That is, we require an estimate of the integral

$$\int_{a_1}^{b_1}\int_{a_2}^{b_2} W_1(x)W_2(y)f(x,y)dx\,dy. \tag{A.5}$$

We will use repeated one-dimensional integration in such situations. That is, the quantity in Equation (A.5) will be estimated as follows:

$$\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} w_{1i}w_{2j}f(x_i,y_j), \tag{A.6}$$

where w_{1i} are the weights and x_i the abscissae for q_1-point Gaussian quadrature with respect to $W_1(x)$, and w_{2j} are the weights and y_j the abscissae for q_2-point Gaussian quadrature with respect to $W_2(y)$.

When $W(x) = 1$, $-1 < x < 1$, the Gaussian quadrature with respect to $W(x)$ is known as Gauss-Legendre quadrature, and the orthogonal polynomials of Equation (A.3) have the following three-term recurrence relationship:

$$(i+1)P_{i+1}(x) = (2i+1)xP_i(x) - iP_{i-1}(x).$$

The Gaussian quadrature with respect to $W(x) = e^{-x}$, $0 < x < \infty$, is known as Gauss-Laguerre quadrature, and the orthogonal polynomials satisfy

$$(i+1)P_{i+1}(x) = (-x+2i+1)P_i(x) - iP_{i-1}(x).$$

Finding the weights and abscissae for such quadratures is not a particularly easy task. However, the following theorem, from [37], provides a relatively simple method for finding such quantities for q-point Gaussian quadrature with respect to weight function $W(x)$:

Theorem A.5. *The weights $\{w_1, w_2, \ldots, w_q\}$ and the abscissae $\{x_1, x_2, \ldots, x_q\}$ can be obtained from the eigenvalue decomposition of the symmetric, tridiagonal Jacobi matrix:*

$$J_q(x) = \begin{pmatrix} a_0 & \sqrt{b_1} & & & \\ \sqrt{b_1} & a_1 & \sqrt{b_2} & & \\ & \sqrt{b_2} & \ddots & \ddots & \\ & & \ddots & a_{q-2} & \sqrt{b_{q-1}} \\ & & & \sqrt{b_{q-1}} & a_{q-1} \end{pmatrix},$$

where a_i and b_i are as given by Theorem A.1. Let V be such that $V^TV = I_q$. Then $V^TJ_qV = \Lambda = diag(\lambda_1, \ldots, \lambda_q)$. It can be shown that $x_i = \lambda_i$, and $w_i = v_{i,1}^2 \int_a^b W(x)dx$, where $v_{i,j}$ is the $(i,j)^{th}$ component of V.

Proof. See [37] for a proof. □

This theorem results in a method for finding the weights and abscissae for q-point Gaussian quadrature with respect to $W(x)$ that is simple to code. In fact, the code for finding the abscissae and weights for Gauss-Legendre and Gauss-Laguerre quadrature is given in Section A.1.

A.1 R code for Gaussian Quadrature

The following code is adapted for R [76] from the Matlab® code given in [37].

The R code for q-point Gauss-Legendre quadrature requires as input q, and outputs the weights and abscissae for such a quadrature:

```
legendrequad <- function(q)
{
L<-matrix(data=0, nrow=q, ncol=q)
u<-c(1:(q-1))
for(i in 1:(q-1)){
u[i]<- sqrt(1/(4 - i^(-2)))
}

#Setting up L
for(i in 1:q){
if(i+1<q+1){
L[i+1,i]<-u[i]
L[i,i+1]<-u[i]
}}
#xk are the abscissae, wk the weights
xk <- sort(eigen(L)$values)
wk <- rev(eigen(L)$vectors[1,]^2)
t<-matrix(data=NA, nrow=2, ncol=q)
t[1,]<-xk
t[2,]<-wk
return(t)
}
```

The R code for q-point Gauss-Laguerre quadrature behaves just as the R code for Gauss-Legendre quadrature:

```
laguerrequad <- function(q)
{
L<-matrix(data=0, nrow=q, ncol=q)
a<-2*c(0:(q-1))+1
u<-c(1:(q-1))

#Setting up L
for(i in 1:q){
L[i,i]<-a[i]
if(i+1<q+1){
L[i+1,i]<-u[i]
L[i,i+1]<-u[i]
}}
#xk are the values, wk the weights
xk <- sort(eigen(L)$values)
wk <- rev(eigen(L)$vectors[1,]^2)
t<-matrix(data=NA, nrow=2, ncol=q)
t[1,]<-xk
t[2,]<-wk
return(t)
}
```

Appendix B

Random Effects Code

B.1 Code for S_1

```
double CGibbsSampler::KnownPhi(int npred, CData& Data)
{
double laguerrex [25] = {0.05670478,0.2990109,0.7359096,1.369183,2.201326,3.235676,4.
double laguerrew [25] = {0.1375260,0.2516453,0.256176,0.1862155,0.1031998,0.04471416,

int L = Data.NumberofCovs;
int N = Data.SampleSize;
double phi = Data.phi;
int i, j, k;

//Reading in the Q matrix
double w;
double Q [N][L];
FILE* in = fopen(Data.mstrQDataFile.c_str(),"r");
if(NULL==in) {
printf("Cannot open data file %s.\n",Data.mstrQDataFile.c_str());
exit(1);
}
for(i=0;i<N;i++) {
for(j=0;j<L;j++) {
fscanf(in,"%lf",&w);
Q[i][j] = w;
}}
fclose(in);

double* QTQ = new double[L*L];
memset(QTQ,0,L*L*sizeof(double));
```

```
for(i=0;i<L; i++){
for(j=0;j<L; j++){
for(k=0;k<N; k++){
QTQ[i*L +j] += Q[k][i]*Q[k][j];
}}}

//Calculating the constants:
double logConstant = 0.0;
logConstant -=  L*log(phi)/2.0+N*log(myPI)/2.0 +lgamma((deltaPrior + npred)/2.0);
logConstant += (npred-N)*log(tauPrior)/2.0;


//Begin the quadrature:
int b=0;
double NumInt = 0.0;
double temp = 0.0;

double x;

int quadsize = 25;

for(b=0; b<quadsize; b++){

x = laguerrex[b];
double logPost = 0.0;

//Need to calculate solve(R) = solve(I/phi + 2*x*QTQ/tauPrior)

double* R = new double[L*L];
memset(R,0,L*L*sizeof(double));

for(i=0; i<L; i++){
for(j=0; j<L; j++){
if(i==j){ R[i*L+i] = 1.0/phi + 2*x*QTQ[i*L+i]/tauPrior; }
else {R[i*L+j] =  2*x*QTQ[i*L+j]/tauPrior; }
}}

//inverting R, stored in R:
if(!Solve(L,R)){ return(DBL_MAX); }

//calculating log det of R
double logDetR = getlogdet(L, R);

//Now need J = I - 2*x*QRQT/tauPrior
```

```
double* QR = new double[N*L];
memset(QR,0,N*L*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<L ; j++){
for(k=0; k<L ; k++){
QR[j*N+i] += Q[i][k]*R[k*L +j];
}}}


double* QRQT = new double[N*N];
memset(QRQT,0,N*N*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<N; j++){
for(k=0; k<L; k++){
QRQT[j*N +i] += QR[k*N +i]*Q[j][k];
}}}

double* J = new double[N*N];
memset(J,0,N*N*sizeof(double));

for(i=0; i<N; i++){
for(j=0; j<N; j++){
if(i==j){ J[i*N+i] = 1 - 2*x*QRQT[i*N+i]/tauPrior; }
else { J[i*N+j] =  - 2*x*QRQT[i*N+j]/tauPrior; }
}}

//Now need yTJy, solve(tauI + ZTJZ), yTJZ, yTJZsolve(tauI + ZTJZ)ZTJy

double yTJy = 0;

//Calculating yTJy
for(i=0;i<N;i++) {
for(j=0;j<N;j++) {
yTJy += Data.Y[i]*J[i * N + j]*Data.Y[j];
}}

// Need  ZTJZ, ZTJy, G = solve(tauI + ZTJZ),
// Start with JZ

double* JZ = new double[N*npred];
memset(JZ,0,N*npred*sizeof(double));
```

```
for(i=0; i<N ; i++){
for(j=0; j<npred ; j++){
for(k=0; k<N ; k++){
JZ[j*N+i] += J[i*N + k]* Data.X[k][j];
}}}

double* yTJZ = new double[npred];
memset(yTJZ,0,npred*sizeof(double));

for(i=0; i<npred ; i++){
for(j=0; j<N ; j++){
yTJZ[i] += Data.Y[j]*JZ[i*N + j];
}}

//calculating G = tauI + ZTJZ

double* G = new double[npred*npred];
memset(G,0,npred*npred*sizeof(double));

for(i=0;i<npred;i++) {
for(j=0;j<npred;j++) {
for(k=0;k<N;k++) {
G[i*npred+j] += Data.X[k][i]*JZ[j*N + k];
}

if(i==j) {
G[i*npred + i] = G[i*npred+i]+tauPrior;
}
}}

//calculate the inverse of G, stored in G
if(!Solve(npred,G)){ return(DBL_MAX); }

//calculating the log determinant of G
double logDetG = getlogdet(npred, G);

//Now require  yTJZGZTJy

double yTJZGZTJy = 0;
for(i=0;i<npred;i++) {
for(j=0;j<npred;j++) {
yTJZGZTJy += yTJZ[i]*G[i * npred + j]*yTJZ[j];
}}

//These are all of the required matrix operations.
```

```
//Now calculate logPost
logPost += logConstant;
logPost += (N + npred + deltaPrior -2.0)*log(x)/2.0;
logPost += logDetR/2.0;
logPost += logDetG/2.0;
logPost -= x*(yTJy - yTJZGZTJy)/tauPrior;
NumInt += laguerrew[b]*exp(logPost);

delete[] R; R = NULL;
delete[] QR; QR = NULL;
delete[] QRQT; QRQT = NULL;
delete[] J; J = NULL;
delete[] JZ; JZ = NULL;
delete[] yTJZ; yTJZ = NULL;
delete[] G; G = NULL;

}

delete[] QTQ; QTQ = NULL;

return(log(NumInt));
}


double CGibbsSampler::KnownPhiZeroPredictors(int TargetGene, CData& Data)
{
double laguerrex [25] = {0.05670478,0.2990109,0.7359096,1.369183,2.201326,3.235676,4.
double laguerrew [25] = {0.1375260,0.2516453,0.256176,0.1862155,0.1031998,0.04471416,

int L = Data.NumberofCovs;
int N = Data.SampleSize;
double phi = Data.phi;
int i, j, k;

//Reading in the Q matrix
double w;
double Q [N][L];
FILE* in = fopen(Data.mstrQDataFile.c_str(),"r");
if(NULL==in) {
printf("Cannot open data file %s.\n",Data.mstrQDataFile.c_str());
exit(1);
}
for(i=0;i<N;i++) {
for(j=0;j<L;j++) {
```

```
fscanf(in,"%lf",&w);
Q[i][j] = w;
}}
fclose(in);

double* QTQ = new double[L*L];
memset(QTQ,0,L*L*sizeof(double));

for(i=0;i<L; i++){
for(j=0;j<L; j++){
for(k=0;k<N; k++){
QTQ[i*L +j] += Q[k][i]*Q[k][j];
}}}

//Calculating the constants:
double logConstant = 0.0;
logConstant -= L*log(phi)+N*log(myPI)/2.0 +lgamma((deltaPrior)/2.0);
logConstant += (-N)*log(tauPrior)/2.0;

//Begin the quadrature:
int b=0;
double NumInt = 0.0;
double temp = 0.0;

double x;

int quadsize = 25;

for(b=0; b<quadsize; b++){

x = laguerrex[b];
double logPost = 0.0;

//Need to calculate solve(R) = solve(I/phi + 2*x*QTQ/tauPrior)

double* R = new double[L*L];
memset(R,0,L*L*sizeof(double));

for(i=0; i<L; i++){
for(j=0; j<L; j++){
if(i==j){ R[i*L+i] = 1.0/phi + 2*x*QTQ[i*L+i]/tauPrior; }
else {R[i*L+j] =  2*x*QTQ[i*L+j]/tauPrior; }
}}

//inverting R, stored in R:
```

```
if(!Solve(L,R)){ return(DBL_MAX); }

//calculating log det of R
double logDetR = getlogdet(L, R);

double* QR = new double[N*L];
memset(QR,0,N*L*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<L ; j++){
for(k=0; k<L ; k++){
QR[j*N+i] += Q[i][k]*R[k*L +j];
}}}

double* QRQT = new double[N*N];
memset(QRQT,0,N*N*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<N; j++){
for(k=0; k<L; k++){
QRQT[j*N +i] += QR[k*N +i]*Q[j][k];
}}}

double* J = new double[N*N];
memset(J,0,N*N*sizeof(double));

for(i=0; i<N; i++){
for(j=0; j<N; j++){
if(i==j){ J[i*N+i] = 1 - 2*x*QRQT[i*N+i]/tauPrior; }
else { J[i*N+j] =  - 2*x*QRQT[i*N+j]/tauPrior; }
}}

//Now need yTJy

double yTJy = 0;

//Calculating yTJy
for(i=0;i<N;i++) {
for(j=0;j<N;j++) {
yTJy += Data.Y[i]*J[i * N + j]*Data.Y[j];
}}

//These are all of the required matrix operations.

//Now calculate logPost
```

```
logPost += logConstant;
logPost += (N + deltaPrior -2)*log(x)/2.0;
logPost += logDetR/2.0;
logPost -= 2*x*(yTJy)/tauPrior;

NumInt += laguerrew[b]*exp(logPost);

delete[] R; R = NULL;
delete[] QR; QR = NULL;
delete[] QRQT; QRQT = NULL;
delete[] J; J = NULL;

}

delete[] QTQ; QTQ = NULL;

return(log(NumInt));
}
```

B.2 Code for S_4

```
double CGibbsSampler::SiteEffectsWishartPosterior(int npred, CData& Data)
{
double laguerrex [25] = {0.05670478,0.2990109,0.7359096,1.369183,2.201326,3.235676,4.
double laguerrew [25] = {0.1375260,0.2516453,0.256176,0.1862155,0.1031998,0.04471416,
double legendrex [25] = {-0.995557,-0.976664,-0.9429746,-0.894992,-0.8334426,-0.75925
double legendrew [25] = {0.005696899,0.01317749,0.02046958,0.02745235,0.03401917,0.04

int L = Data.NumberofCovs;
int N = Data.SampleSize;
int i, j, k;

//Reading in the Q matrix
double w;
double Q [N][L];
FILE* in = fopen(Data.mstrQDataFile.c_str(),"r");
if(NULL==in) {
printf("Cannot open data file %s.\n",Data.mstrQDataFile.c_str());
exit(1);
}
for(i=0;i<N;i++) {
for(j=0;j<L;j++) {
fscanf(in,"%lf",&w);
```

```
Q[i][j] = w;
}
}
fclose(in);


double* QTQ = new double[L*L];
memset(QTQ,0,L*L*sizeof(double));

for(i=0;i<L; i++){
for(j=0;j<L; j++){
for(k=0;k<N; k++){
QTQ[i*L +j] += Q[k][i]*Q[k][j];
}
}
}

//Calculating the constants:

double logConstant = 0.0;
logConstant -= log(2.0) + L*log(kappa)+N*log(myPI)/2.0
+lgamma((deltaPrior + npred)/2.0);
logConstant += (npred-N)*log(tauPrior)/2.0;

//Begin the quadrature:
int c=0;
int b=0;
double NumInt = 0.0;
double temp = 0.0;

double u;
double x;

int quadsize = 25;

for(c=0; c<quadsize; c++){
for(b=0; b<quadsize; b++){

u = legendrex[c];
x = laguerrex[b];
double logPost = 0.0;

//Need to calculate solve(R) = solve((2/(kappa^2(1+u)))I + QTQ/x)

double* R = new double[L*L];
```

```
memset(R,0,L*L*sizeof(double));

for(i=0; i<L; i++){
for(j=0; j<L; j++){
//if(i==j){ R[i*L+i] = 2.0/(kappa*kappa*(1+u)) + QTQ[i*L+i]/x; }
//else {R[i*L+j] =  QTQ[i*L+j]/x; }
//Trying my new derivation:
if(i==j){ R[i*L+i] = 1.0/(kappa*kappa*(1+u)) + x*QTQ[i*L+i]/tauPrior; }
else {R[i*L+j] =  QTQ[i*L+j]/x; }
}
}

//inverting R, stored in R:
if(!Solve(L,R)){ return(DBL_MAX); }

//calculating log det of R
double logDetR = getlogdet(L, R);

//Now need J = I - QRQT/x

double* QR = new double[N*L];
memset(QR,0,N*L*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<L ; j++){
for(k=0; k<L ; k++){
QR[j*N+i] += Q[i][k]*R[k*L +j];
}
}
}

double* QRQT = new double[N*N];
memset(QRQT,0,N*N*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<N; j++){
for(k=0; k<L; k++){
QRQT[j*N +i] += QR[k*N +i]*Q[j][k];
}
}
}

double* J = new double[N*N];
memset(J,0,N*N*sizeof(double));
```

```
for(i=0; i<N; i++){
for(j=0; j<N; j++){
//if(i==j){ J[i*N+i] = 1 - QRQT[i*N+i]/x; }
//else { J[i*N+j] =  - QRQT[i*N+j]/x; }
//Trying my new derivation:
if(i==j){ J[i*N+i] = 1 - x*QRQT[i*N+i]/tauPrior; }
else { J[i*N+j] =  - x*QRQT[i*N+j]/tauPrior; }
}
}

//Now need yTJy, solve(tauI + ZTJZ), yTJZ, yTJZsolve(tauI + ZTJZ)ZTJy

double yTJy = 0;

//Calculating yTJy
for(i=0;i<N;i++) {
for(j=0;j<N;j++) {
yTJy += Data.Y[i]*J[i * N + j]*Data.Y[j];
}
}

// Need  ZTJZ, ZTJy, G = solve(tauI + ZTJZ),
// Start with JZ

double* JZ = new double[N*npred];
memset(JZ,0,N*npred*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<npred ; j++){
for(k=0; k<N ; k++){
JZ[j*N+i] += J[i*N + k]* Data.X[k][j];
}
}
}

double* yTJZ = new double[npred];
memset(yTJZ,0,npred*sizeof(double));

for(i=0; i<npred ; i++){
for(j=0; j<N ; j++){
yTJZ[i] += Data.Y[j]*JZ[i*N + j];
}
}

//calculating G = tauI + ZTJZ
```

```
double* G = new double[npred*npred];
memset(G,0,npred*npred*sizeof(double));

for(i=0;i<npred;i++) {
for(j=0;j<npred;j++) {
for(k=0;k<N;k++) {
G[i*npred+j] += Data.X[k][i]*JZ[j*N + k];
}

if(i==j) {
G[i*npred + i] = G[i*npred+i]+tauPrior;
}
}
}

//calculate the inverse of G, stored in G
if(!Solve(npred,G)){ return(DBL_MAX); }

//calculating the log determinant of G
double logDetG = getlogdet(npred, G);

//Now require  yTJZGZTJy

double yTJZGZTJy = 0;
for(i=0;i<npred;i++) {
for(j=0;j<npred;j++) {
yTJZGZTJy += yTJZ[i]*G[i * npred + j]*yTJZ[j];
}
}

//These are all of the required matrix operations.

//Now calculate logPost
logPost += logConstant;
logPost -= (L+1)*log(1+u)/2.0;
logPost += (N + npred + deltaPrior -2.0)*log(x)/2.0;
logPost += logDetR/2.0;
logPost += logDetG/2.0;
logPost -= x*(yTJy - yTJZGZTJy)/tauPrior;
NumInt += laguerrew[b]*legendrew[c]*exp(logPost);

delete[] R; R = NULL;
delete[] QR; QR = NULL;
delete[] QRQT; QRQT = NULL;
```

```
delete[] J; J = NULL;
delete[] JZ; JZ = NULL;
delete[] yTJZ; yTJZ = NULL;
delete[] G; G = NULL;

}}

delete[] QTQ; QTQ = NULL;

return(log(NumInt));
}


double CGibbsSampler::SiteEffectsZeroPredictors(int TargetGene, CData& Data)
{
double laguerrex [25] = {0.05670478,0.2990109,0.7359096,1.369183,2.201326,3.235676,4.
double laguerrew [25] = {0.1375260,0.2516453,0.256176,0.1862155,0.1031998,0.04471416,
double legendrex [25] = {-0.995557,-0.976664,-0.9429746,-0.894992,-0.8334426,-0.75925
double legendrew [25] = {0.005696899,0.01317749,0.02046958,0.02745235,0.03401917,0.04

int L = Data.NumberofCovs;
int N = Data.SampleSize;
int i, j, k;

//Reading in the Q matrix
double w;
double Q [N][L];
FILE* in = fopen(Data.mstrQDataFile.c_str(),"r");
if(NULL==in) {
printf("Cannot open data file %s.\n",Data.mstrQDataFile.c_str());
exit(1);
}
for(i=0;i<N;i++) {
for(j=0;j<L;j++) {
fscanf(in,"%lf",&w);
Q[i][j] = w;
}
}
fclose(in);

double* QTQ = new double[L*L];
memset(QTQ,0,L*L*sizeof(double));

for(i=0;i<L; i++){
for(j=0;j<L; j++){
```

```
for(k=0;k<N; k++){
QTQ[i*L +j] += Q[k][i]*Q[k][j];
}
}
}

//Calculating the constants:
double logConstant = 0.0;
logConstant -= log(2.0) + L*log(kappa)+N*log(myPI)/2.0 +lgamma((deltaPrior)/2.0);
logConstant += (-N)*log(tauPrior)/2.0;

//Begin the quadrature:
int c=0;
int b=0;
double NumInt = 0.0;
double temp = 0.0;

double u;
double x;

int quadsize = 25;

for(c=0; c<quadsize; c++){
for(b=0; b<quadsize; b++){

u = legendrex[c];
x = laguerrex[b];
double logPost = 0.0;

//Need to calculate solve(R) = solve((2/(kappa^2(1+u)))I + QTQ/x)

double* R = new double[L*L];
memset(R,0,L*L*sizeof(double));

for(i=0; i<L; i++){
for(j=0; j<L; j++){
//Trying new calcs
if(i==j){ R[i*L+i] = 1.0/(kappa*kappa*(1+u)) + x*QTQ[i*L+i]/tauPrior; }
else {R[i*L+j] =  x*QTQ[i*L+j]/tauPrior; }
//if(i==j){ R[i*L+i] = 2.0/(kappa*kappa*(1+u)) + QTQ[i*L+i]/x; }
//else {R[i*L+j] =  QTQ[i*L+j]/x; }
}
}

//inverting R, stored in R:
```

```
if(!Solve(L,R)){ return(DBL_MAX); }

//calculating log det of R
double logDetR = getlogdet(L, R);

double* QR = new double[N*L];
memset(QR,0,N*L*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<L ; j++){
for(k=0; k<L ; k++){
QR[j*N+i] += Q[i][k]*R[k*L +j];
}
}
}

double* QRQT = new double[N*N];
memset(QRQT,0,N*N*sizeof(double));

for(i=0; i<N ; i++){
for(j=0; j<N; j++){
for(k=0; k<L; k++){
QRQT[j*N +i] += QR[k*N +i]*Q[j][k];
}
}
}

double* J = new double[N*N];
memset(J,0,N*N*sizeof(double));

for(i=0; i<N; i++){
for(j=0; j<N; j++){
//Trying new calcs
if(i==j){ J[i*N+i] = 1 - x*QRQT[i*N+i]/tauPrior; }
else { J[i*N+j] =  - x*QRQT[i*N+j]/tauPrior; }
//if(i==j){ J[i*N+i] = 1 - QRQT[i*N+i]/x; }
//else { J[i*N+j] =  - QRQT[i*N+j]/x; }
}
}

//Now need yTJy

double yTJy = 0;

//Calculating yTJy
```

```
for(i=0;i<N;i++) {
for(j=0;j<N;j++) {
yTJy += Data.Y[i]*J[i * N + j]*Data.Y[j];
}
}

//These are all of the required matrix operations.

//Now calculate logPost
logPost += logConstant;
logPost -= (L+1)*log(1+u)/2.0;
logPost += (N + deltaPrior -2)*log(x)/2.0;
logPost += logDetR/2.0;
logPost -= x*(yTJy)/tauPrior;

NumInt += laguerrew[b]*legendrew[c]*exp(logPost);

delete[] R; R = NULL;
delete[] QR; QR = NULL;
delete[] QRQT; QRQT = NULL;
delete[] J; J = NULL;

}}

delete[] QTQ; QTQ = NULL;

return(log(NumInt));
}
```

B.3 Posterior Sampling Code

B.3.1 Posterior sampling when ϕ fixed

```
 posteriorSamplesS1<-function(samplesize, phi, data.matrix, Q){

# data.matrix is the matrix containing the data
# Q is the Q matrix

p <- ncol(data.matrix)
n <- nrow(data.matrix)
m<-ncol(Q)

tau<-1
delta<-2

b.samples<-matrix(data=NA, nrow=samplesize, ncol=m*p)
psi.samples<-matrix(data=NA, nrow=samplesize+1, ncol=p)

#initial guesses to start the Gibbs Sampling:
psi.samples[1,]<-rinvgamma(p, delta/2 ,tau/2)

#start sampling from joint distribution:
for(j in 1:p){
for(i in 2:(samplesize+1)){

mean.b<-solve(diag(psi.samples[i-1,j]/phi,m)+t(Q)%*%Q)%*%t(Q)%*%data.matrix[,j]
var.b<-solve(diag(1/phi,m)+ t(Q)%*%Q/psi.samples[i-1,j])

b.samples[i-1,((j-1)*m+1):(j*m)]<-rmvnorm(1,mean=mean.b, sigma=var.b)

beta.psi<-tau/2+ t(data.matrix[,j]-Q%*%b.samples[i-1,((j-1)*m+1):(j*m)])
%*%(data.matrix[,j]-Q%*%b.samples[i-1,((j-1)*m+1):(j*m)])/2

psi.samples[i,j]<-rinvgamma(1,shape=(n+delta)/2, scale=beta.psi)
}
}

results<-NULL
results$b<-b.samples
results$psi<-psi.samples

return(results)
}
```

B.3.2 Posterior sampling when $\phi_i = \upsilon^{-1}\psi_i$

```
 posteriorSamplesS2<-function(samplesize, upsilon, data.matrix, Q){

# data.matrix is the matrix containing the data
# Q is the Q matrix

p <- ncol(data.matrix)
n <- nrow(data.matrix)
m<-ncol(Q)

tau<-1
delta<-2
b.samples<-matrix(data=NA, nrow=samplesize, ncol=m*p)
psi.samples<-matrix(data=NA, nrow=samplesize+1, ncol=p)

# initial guesses to initialise the Gibbs sampling:
psi.samples[1,]<-rinvgamma(p, delta/2 ,tau/2)

# start sampling from joint distribution:
for(j in 1:p){
for(i in 2:(samplesize+1)){

mean.b<-solve(diag(upsilon,m)+t(Q)%*%Q)%*%t(Q)%*%data.matrix[,j]
var.b<-solve(diag(upsilon,m)+ t(Q)%*%Q)*psi.samples[i-1,j]

b.samples[i-1,((j-1)*m+1):(j*m)]<-rmvnorm(1,mean=mean.b, sigma=var.b)

beta.psi<-tau/2
+upsilon*t(b.samples[i-1,((j-1)*m+1):(j*m)])%*%b.samples[i-1,((j-1)*m+1):(j*m)]/2
+t(data.matrix[,j]-Q%*%b.samples[i-1,((j-1)*m+1):(j*m)])
%*%(data.matrix[,j]-Q%*%b.samples[i-1,((j-1)*m+1):(j*m)])/2

psi.samples[i,j]<-rinvgamma(1,shape=(n+m+delta)/2, scale=beta.psi)
}
}

results<-NULL
results$b<-b.samples
results$psi<-psi.samples

return(results)
}
```

B.3.3 Posterior sampling when $\phi_i^{\frac{1}{2}} \sim$ Uniform$\,(0, \kappa)$

```
posteriorSamplesS4<-function(samplesize, data.matrix, Q){

# data.matrix is the matrix containing the data
# Q is the Q matrix

p <- ncol(data.matrix)
n <- nrow(data.matrix)
m<-ncol(Q)

tau<-1
delta<-2

b.samples<-matrix(data=NA, nrow=samplesize, ncol=m*p)
psi.samples<-matrix(data=NA, nrow=samplesize+1, ncol=p)
phi.samples<-matrix(data=NA, nrow=samplesize+1, ncol=p)

#initial guesses to start the Gibbs Sampling:
psi.samples[1,]<-rinvgamma(p, delta/2,tau/2)
phi.samples[1,]<-(runif(p, min=0, max=2))^2

#start sampling from joint distribution:
for(j in 1:p){

for(i in 2:(samplesize+1)){
#samples of b
mean.b<-solve(diag(psi.samples[i-1,j]/phi.samples[i-1,j],m)
+t(Q)%*%Q)%*%t(Q)%*%data.matrix[,j]
var.b<-solve(diag(1/phi.samples[i-1,j],m)+ t(Q)%*%Q/psi.samples[i-1,j])
b.samples[i-1,((j-1)*m+1):(j*m)]<-rmvnorm(1,mean=mean.b, sigma=var.b)

#samples of psi
beta.psi<-tau/2+ t(data.matrix[,j]-Q%*%b.samples[i-1,((j-1)*m+1):(j*m)])
%*%(data.matrix[,j]-Q%*%b.samples[i-1,((j-1)*m+1):(j*m)])/2

psi.samples[i,j]<-rinvgamma(1,shape=(n+delta)/2, scale=beta.psi)

#samples of phi
beta.phi<-t(b.samples[i-1,((j-1)*m+1):(j*m)])%*%b.samples[i-1,((j-1)*m+1):(j*m)]
phi.samples[i,j]<-rinvgamma(1,shape=(m-1)/2, scale=beta.phi)
}
}

results<-NULL
results$b<-b.samples
results$psi<-psi.samples
results$phi<-phi.samples

return(results)
}
```

Appendix C

Grape Gene Data

C.1 Boxplots of Gene Expression Levels

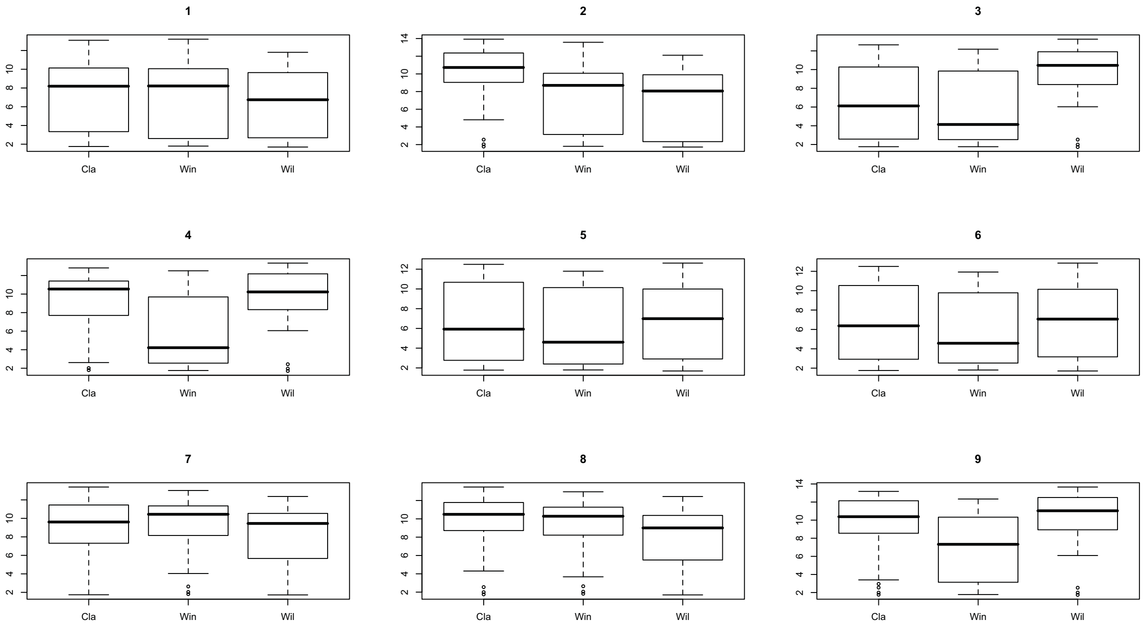


Figure C.1.1: Boxplots of the expression levels of genes 1 to 9 for grapes sampled at each of the vineyards.

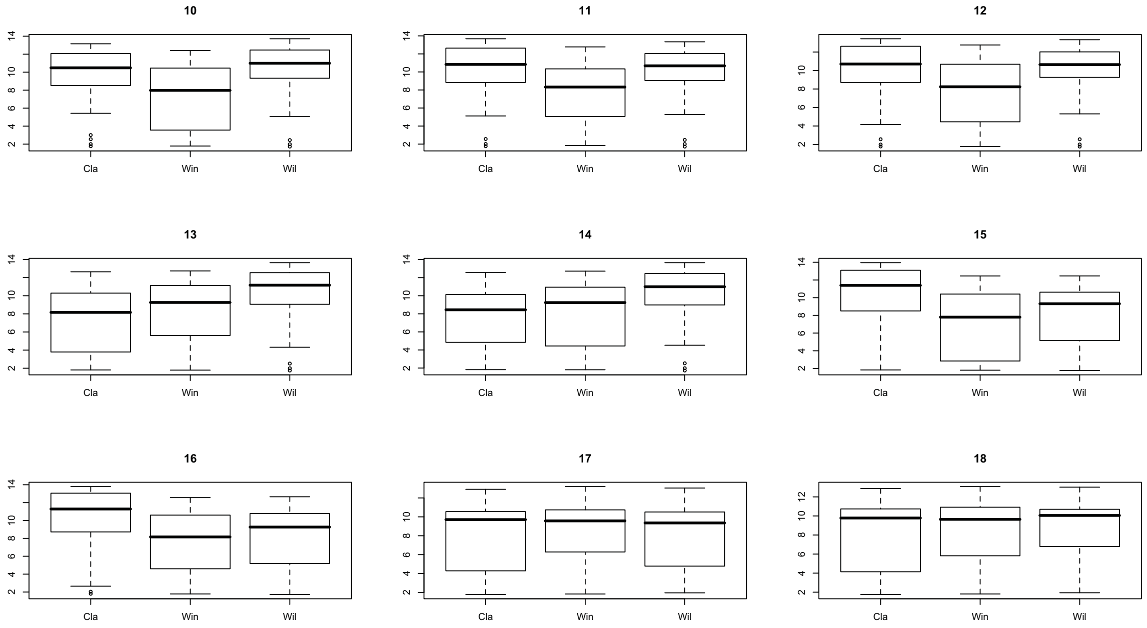


Figure C.1.2: Boxplots of the expression levels of genes 10 to 18 for grapes sampled at each of the vineyards.

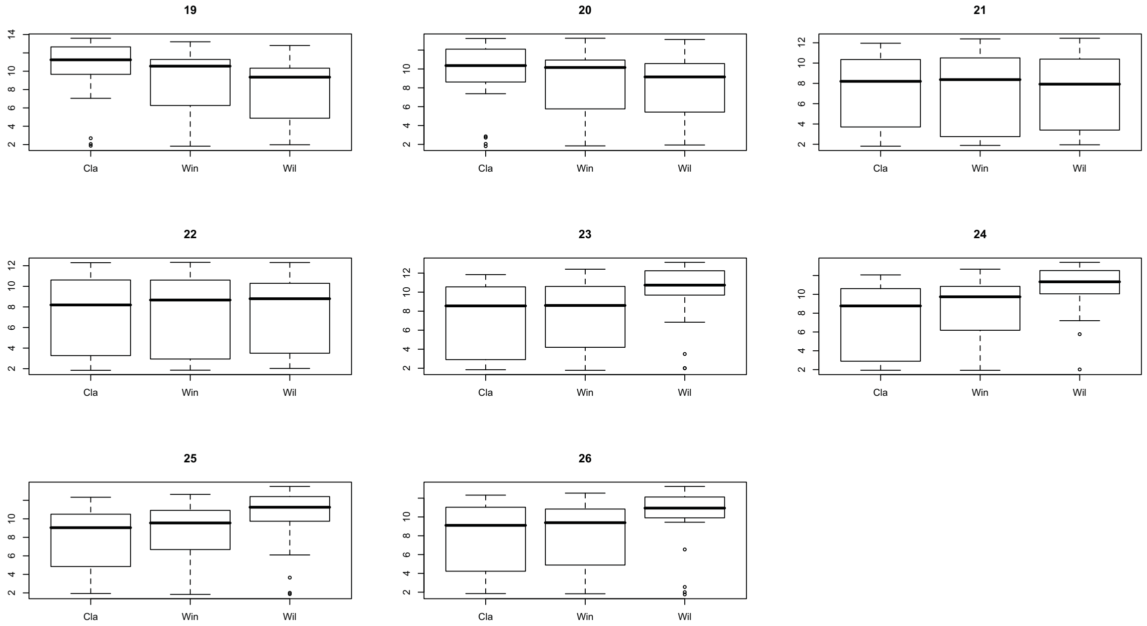


Figure C.1.3: Boxplots of the expression levels of genes 19 to 26 for grapes sampled at each of the vineyards.

C.2 Differences Between Vineyards

Anova for each of the genes: testing to see if there are differences in mean gene expression levels at each vineyard.

```
> for(i in 1:26){
+  write(i,"", sep="\n")
+  print(anova(lm(Go_heat_ALL_wks3_7[,i]˜0+Q.wks3_7[,1:3])))
+  }
1
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 4659.4  1553.1    3546 < 2.2e-16 ***
Residuals       47   20.6     0.4
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
2
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 7265.2  2421.7   24455 < 2.2e-16 ***
Residuals       47    4.7     0.1
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
3
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df  Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 1009.45  336.48  63.054 < 2.2e-16 ***
Residuals       47  250.81    5.34
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
4
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 5667.3  1889.1   22753 < 2.2e-16 ***
```

```
Residuals       47    3.9     0.1
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
5
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df  Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 2404.54  801.51  115.05 < 2.2e-16 ***
Residuals       47  327.43    6.97
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
6
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df  Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 2805.11  935.04  109.78 < 2.2e-16 ***
Residuals       47  400.30    8.52
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
7
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 6019.0  2006.3   13153 < 2.2e-16 ***
Residuals       47    7.2     0.2
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
8
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 655.10  218.37  86.782 < 2.2e-16 ***
Residuals       47 118.26    2.52
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
9
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df  Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 200.193  66.731   79039 < 2.2e-16 ***
```

```
Residuals       47   0.040   0.001
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
10
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 4738.7  1579.6  119.52 < 2.2e-16 ***
Residuals       47  621.2    13.2
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
11
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 5588.1  1862.7   12505 < 2.2e-16 ***
Residuals       47    7.0     0.1
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
12
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 3518.2  1172.7  101.39 < 2.2e-16 ***
Residuals       47  543.6    11.6
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
13
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 3379.2  1126.4  3979.6 < 2.2e-16 ***
Residuals       47   13.3     0.3
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
14
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 5580.1  1860.0  272.53 < 2.2e-16 ***
```

```
Residuals       47  320.8     6.8
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
15
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 7970.3  2656.8   17805 < 2.2e-16 ***
Residuals       47    7.0     0.1
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
16
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df  Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 2947.68  982.56  83.675 < 2.2e-16 ***
Residuals       47  551.90   11.74
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
17
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 4476.1  1492.0   15076 < 2.2e-16 ***
Residuals       47    4.7     0.1
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
18
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 6508.0  2169.3   31209 < 2.2e-16 ***
Residuals       47    3.3     0.1
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
19
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 4354.2  1451.4  9183.3 < 2.2e-16 ***
```

```
Residuals       47    7.4     0.2
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
20
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 7720.4  2573.5   11723 < 2.2e-16 ***
Residuals       47   10.3     0.2
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
21
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df  Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 156.731  52.244   97728 < 2.2e-16 ***
Residuals       47   0.025   0.001
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
22
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 4768.7  1589.6  165.83 < 2.2e-16 ***
Residuals       47  450.5     9.6
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
23
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 867.33  289.11  150.39 < 2.2e-16 ***
Residuals       47  90.36    1.92
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
24
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 4220.0  1406.7  111.63 < 2.2e-16 ***
```

```
Residuals       47  592.3    12.6
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
25
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq  Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 329.54   109.85   50217 < 2.2e-16 ***
Residuals       47   0.10 0.002187
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
26
Analysis of Variance Table

Response: Go_heat_ALL_wks3_7[, i]
                Df Sum Sq Mean Sq F value    Pr(>F)
Q.wks3_7[, 1:3]  3 4609.6  1536.5  378.09 < 2.2e-16 ***
Residuals       47  191.0     4.1
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

C.3 Regressing the Gene Expressions on Temperature

```
> summary(lm(Go_heat_ALL_wks3_7[,] ~ 0 + Q.wks3_7[,1:23]))
Response Y1 :

Call:
lm(formula = Y1 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-2.986e-01 -7.290e-02 -1.830e-14  7.225e-02  2.986e-01

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  23.926454  10.396930   2.301   0.0293 *
Q.wks3_7[, 1:23]2  23.847902  11.119684   2.145   0.0411 *
Q.wks3_7[, 1:23]3  25.246854  11.027896   2.289   0.0301 *
Q.wks3_7[, 1:23]4  -0.564532   0.636569  -0.887   0.3830
```

```
Q.wks3_7[, 1:23]5   1.673556   1.242142   1.347   0.1891
Q.wks3_7[, 1:23]6  -5.840206   4.754277  -1.228   0.2299
Q.wks3_7[, 1:23]7   3.259146   3.874966   0.841   0.4077
Q.wks3_7[, 1:23]8  -0.074284   1.542317  -0.048   0.9619
Q.wks3_7[, 1:23]9   0.324949   1.041299   0.312   0.7574
Q.wks3_7[, 1:23]10  0.029958   0.064698   0.463   0.6470
Q.wks3_7[, 1:23]11 -0.056699   0.128115  -0.443   0.6616
Q.wks3_7[, 1:23]12  0.200990   0.361875   0.555   0.5832
Q.wks3_7[, 1:23]13 -0.092370   0.151641  -0.609   0.5475
Q.wks3_7[, 1:23]14 -0.072968   0.090809  -0.804   0.4287
Q.wks3_7[, 1:23]15  0.206047   0.335091   0.615   0.5438
Q.wks3_7[, 1:23]16 -0.471765   0.383496  -1.230   0.2292
Q.wks3_7[, 1:23]17  0.132232   0.121581   1.088   0.2864
Q.wks3_7[, 1:23]18  0.035681   0.099686   0.358   0.7232
Q.wks3_7[, 1:23]19  0.125065   0.192208   0.651   0.5208
Q.wks3_7[, 1:23]20 -0.002469   0.195302  -0.013   0.9900
Q.wks3_7[, 1:23]21 -0.056914   0.133873  -0.425   0.6741
Q.wks3_7[, 1:23]22 -0.049160   0.139301  -0.353   0.7269
Q.wks3_7[, 1:23]23  0.098352   0.141443   0.695   0.4928
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.178 on 27 degrees of freedom
Multiple R-squared: 0.9998,Adjusted R-squared: 0.9997
F-statistic:  6422 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y2 :

Call:
lm(formula = Y2 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-4.963e-01 -7.192e-02 -1.487e-13  8.754e-02  3.675e-01

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  10.6425299 12.9479803   0.822   0.4183
Q.wks3_7[, 1:23]2   9.7262569 13.8480735   0.702   0.4885
Q.wks3_7[, 1:23]3  10.6074089 13.7337631   0.772   0.4466
Q.wks3_7[, 1:23]4   0.4150867  0.7927613   0.524   0.6048
Q.wks3_7[, 1:23]5  -2.8761773  1.5469214  -1.859   0.0739 .
Q.wks3_7[, 1:23]6   7.0696355  5.9208140   1.194   0.2429
Q.wks3_7[, 1:23]7  -5.3424229  4.8257498  -1.107   0.2780
```

```
Q.wks3_7[, 1:23]8   2.0093160  1.9207486   1.046   0.3048
Q.wks3_7[, 1:23]9  -1.2535260  1.2967983  -0.967   0.3423
Q.wks3_7[, 1:23]10  0.0278538  0.0805725   0.346   0.7322
Q.wks3_7[, 1:23]11  0.1589988  0.1595505   0.997   0.3278
Q.wks3_7[, 1:23]12 -0.4490539  0.4506663  -0.996   0.3279
Q.wks3_7[, 1:23]13  0.2600650  0.1888487   1.377   0.1798
Q.wks3_7[, 1:23]14 -0.0195580  0.1130905  -0.173   0.8640
Q.wks3_7[, 1:23]15 -0.3794159  0.4173106  -0.909   0.3713
Q.wks3_7[, 1:23]16  0.6855058  0.4775932   1.435   0.1627
Q.wks3_7[, 1:23]17 -0.3059023  0.1514127  -2.020   0.0534 .
Q.wks3_7[, 1:23]18  0.1130594  0.1241455   0.911   0.3705
Q.wks3_7[, 1:23]19  0.0008476  0.2393688   0.004   0.9972
Q.wks3_7[, 1:23]20  0.1535135  0.2432220   0.631   0.5332
Q.wks3_7[, 1:23]21 -0.2489243  0.1667206  -1.493   0.1470
Q.wks3_7[, 1:23]22 -0.1981405  0.1734806  -1.142   0.2634
Q.wks3_7[, 1:23]23  0.2030699  0.1761483   1.153   0.2591
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2217 on 27 degrees of freedom
Multiple R-squared: 0.9998,Adjusted R-squared: 0.9997
F-statistic:  6432 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y3 :

Call:
lm(formula = Y3 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-1.058e+00 -5.951e-02  8.633e-14  5.625e-02  1.058e+00

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   108.0462    26.1538   4.131 0.000313 ***
Q.wks3_7[, 1:23]2   118.4763    27.9720   4.236 0.000237 ***
Q.wks3_7[, 1:23]3   118.6709    27.7411   4.278 0.000211 ***
Q.wks3_7[, 1:23]4     9.4618     1.6013   5.909 2.69e-06 ***
Q.wks3_7[, 1:23]5    17.9080     3.1247   5.731 4.31e-06 ***
Q.wks3_7[, 1:23]6  -119.2291    11.9596  -9.969 1.52e-10 ***
Q.wks3_7[, 1:23]7    96.7334     9.7476   9.924 1.67e-10 ***
Q.wks3_7[, 1:23]8   -33.3215     3.8798  -8.589 3.34e-09 ***
Q.wks3_7[, 1:23]9    19.0994     2.6194   7.291 7.65e-08 ***
Q.wks3_7[, 1:23]10   -1.3058     0.1627  -8.023 1.27e-08 ***
```

```
Q.wks3_7[, 1:23]11   -2.4716     0.3223  -7.669 3.00e-08 ***
Q.wks3_7[, 1:23]12    8.1734     0.9103   8.979 1.36e-09 ***
Q.wks3_7[, 1:23]13   -3.9017     0.3815 -10.228 8.73e-11 ***
Q.wks3_7[, 1:23]14   -0.7612     0.2284  -3.332 0.002507 **
Q.wks3_7[, 1:23]15    7.3462     0.8429   8.715 2.49e-09 ***
Q.wks3_7[, 1:23]16   -8.8293     0.9647  -9.152 9.16e-10 ***
Q.wks3_7[, 1:23]17    1.4184     0.3058   4.638 8.06e-05 ***
Q.wks3_7[, 1:23]18    0.5813     0.2508   2.318 0.028266 *
Q.wks3_7[, 1:23]19   -3.6161     0.4835  -7.479 4.80e-08 ***
Q.wks3_7[, 1:23]20    1.9861     0.4913   4.043 0.000395 ***
Q.wks3_7[, 1:23]21    1.7900     0.3368   5.315 1.31e-05 ***
Q.wks3_7[, 1:23]22    2.0165     0.3504   5.755 4.05e-06 ***
Q.wks3_7[, 1:23]23   -2.2400     0.3558  -6.296 9.75e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.4477 on 27 degrees of freedom
Multiple R-squared: 0.9957,Adjusted R-squared: 0.992
F-statistic: 272.2 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y4 :

Call:
lm(formula = Y4 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-2.432e-01 -4.628e-02  1.087e-14  5.841e-02  1.739e-01

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  24.955477   6.064807   4.115 0.000326 ***
Q.wks3_7[, 1:23]2  25.400742   6.486409   3.916 0.000552 ***
Q.wks3_7[, 1:23]3  25.916738   6.432866   4.029 0.000410 ***
Q.wks3_7[, 1:23]4   0.061075   0.371328   0.164 0.870581
Q.wks3_7[, 1:23]5  -1.341997   0.724575  -1.852 0.074973 .
Q.wks3_7[, 1:23]6   1.033236   2.773297   0.373 0.712380
Q.wks3_7[, 1:23]7  -2.060551   2.260371  -0.912 0.370048
Q.wks3_7[, 1:23]8   2.103096   0.899675   2.338 0.027068 *
Q.wks3_7[, 1:23]9  -1.213053   0.607418  -1.997 0.055989 .
Q.wks3_7[, 1:23]10  0.003769   0.037740   0.100 0.921184
Q.wks3_7[, 1:23]11 -0.094216   0.074733  -1.261 0.218203
Q.wks3_7[, 1:23]12  0.204576   0.211091   0.969 0.341081
Q.wks3_7[, 1:23]13 -0.035539   0.088456  -0.402 0.691013
```

```
Q.wks3_7[, 1:23]14 -0.085914   0.052971  -1.622 0.116445
Q.wks3_7[, 1:23]15  0.094041   0.195467   0.481 0.634315
Q.wks3_7[, 1:23]16 -0.125502   0.223704  -0.561 0.579413
Q.wks3_7[, 1:23]17 -0.012323   0.070921  -0.174 0.863348
Q.wks3_7[, 1:23]18  0.109520   0.058150   1.883 0.070458 .
Q.wks3_7[, 1:23]19 -0.020771   0.112120  -0.185 0.854409
Q.wks3_7[, 1:23]20  0.135339   0.113925   1.188 0.245192
Q.wks3_7[, 1:23]21 -0.177928   0.078092  -2.278 0.030830 *
Q.wks3_7[, 1:23]22 -0.185291   0.081258  -2.280 0.030707 *
Q.wks3_7[, 1:23]23  0.224270   0.082507   2.718 0.011326 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1038 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9999
F-statistic: 2.287e+04 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y5 :

Call:
lm(formula = Y5 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-8.700e-01 -1.155e-01 -1.464e-13  1.039e-01  8.381e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   71.4174    22.1854   3.219 0.003336 **
Q.wks3_7[, 1:23]2   77.6566    23.7276   3.273 0.002914 **
Q.wks3_7[, 1:23]3   78.5476    23.5317   3.338 0.002471 **
Q.wks3_7[, 1:23]4    6.4611     1.3583   4.757 5.86e-05 ***
Q.wks3_7[, 1:23]5    0.6796     2.6505   0.256 0.799589
Q.wks3_7[, 1:23]6  -56.7707    10.1449  -5.596 6.17e-06 ***
Q.wks3_7[, 1:23]7   54.0640     8.2685   6.539 5.19e-07 ***
Q.wks3_7[, 1:23]8  -18.6473     3.2911  -5.666 5.12e-06 ***
Q.wks3_7[, 1:23]9    7.8124     2.2220   3.516 0.001568 **
Q.wks3_7[, 1:23]10  -0.8736     0.1381  -6.328 8.96e-07 ***
Q.wks3_7[, 1:23]11  -0.8356     0.2734  -3.057 0.004998 **
Q.wks3_7[, 1:23]12   3.8549     0.7722   4.992 3.11e-05 ***
Q.wks3_7[, 1:23]13  -1.7255     0.3236  -5.332 1.25e-05 ***
Q.wks3_7[, 1:23]14  -0.6042     0.1938  -3.118 0.004295 **
Q.wks3_7[, 1:23]15   3.6059     0.7150   5.043 2.71e-05 ***
Q.wks3_7[, 1:23]16  -3.3167     0.8183  -4.053 0.000385 ***
```

```
Q.wks3_7[, 1:23]17  -0.1649     0.2594  -0.636 0.530401
Q.wks3_7[, 1:23]18   0.7671     0.2127   3.606 0.001241 **
Q.wks3_7[, 1:23]19  -2.9209     0.4101  -7.122 1.17e-07 ***
Q.wks3_7[, 1:23]20   1.9774     0.4167   4.745 6.05e-05 ***
Q.wks3_7[, 1:23]21   0.5530     0.2857   1.936 0.063412 .
Q.wks3_7[, 1:23]22   0.7582     0.2972   2.551 0.016730 *
Q.wks3_7[, 1:23]23  -0.9373     0.3018  -3.105 0.004430 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.3798 on 27 degrees of freedom
Multiple R-squared: 0.9986,Adjusted R-squared: 0.9974
F-statistic: 822.3 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y6 :

Call:
lm(formula = Y6 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-1.548e+00 -1.425e-01  3.986e-13  1.425e-01  1.284e+00

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   94.5840    34.5008   2.742 0.010717 *
Q.wks3_7[, 1:23]2  102.0390    36.8991   2.765 0.010128 *
Q.wks3_7[, 1:23]3  101.7750    36.5945   2.781 0.009754 **
Q.wks3_7[, 1:23]4    3.6336     2.1124   1.720 0.096852 .
Q.wks3_7[, 1:23]5    8.9007     4.1219   2.159 0.039867 *
Q.wks3_7[, 1:23]6  -72.6286    15.7764  -4.604 8.84e-05 ***
Q.wks3_7[, 1:23]7   63.3529    12.8585   4.927 3.71e-05 ***
Q.wks3_7[, 1:23]8  -22.6167     5.1180  -4.419 0.000145 ***
Q.wks3_7[, 1:23]9   11.2088     3.4554   3.244 0.003135 **
Q.wks3_7[, 1:23]10  -0.9754     0.2147  -4.543 0.000104 ***
Q.wks3_7[, 1:23]11  -1.3151     0.4251  -3.093 0.004564 **
Q.wks3_7[, 1:23]12   5.2491     1.2008   4.371 0.000165 ***
Q.wks3_7[, 1:23]13  -2.3475     0.5032  -4.665 7.49e-05 ***
Q.wks3_7[, 1:23]14  -0.6761     0.3013  -2.244 0.033259 *
Q.wks3_7[, 1:23]15   4.6493     1.1120   4.181 0.000274 ***
Q.wks3_7[, 1:23]16  -5.1987     1.2726  -4.085 0.000353 ***
Q.wks3_7[, 1:23]17   0.5166     0.4034   1.281 0.211246
Q.wks3_7[, 1:23]18   0.6634     0.3308   2.005 0.055036 .
Q.wks3_7[, 1:23]19  -2.9110     0.6378  -4.564 9.83e-05 ***
```

```
Q.wks3_7[, 1:23]20   1.7365     0.6481   2.679 0.012405 *
Q.wks3_7[, 1:23]21   0.9126     0.4442   2.054 0.049753 *
Q.wks3_7[, 1:23]22   1.0893     0.4623   2.356 0.025959 *
Q.wks3_7[, 1:23]23  -1.2222     0.4694  -2.604 0.014796 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.5906 on 27 degrees of freedom
Multiple R-squared: 0.9971,Adjusted R-squared: 0.9946
F-statistic: 398.3 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y7 :

Call:
lm(formula = Y7 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-4.123e-01 -5.966e-02 -3.208e-15  1.044e-01  3.196e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   6.33019   10.12357   0.625    0.537
Q.wks3_7[, 1:23]2   4.92645   10.82733   0.455    0.653
Q.wks3_7[, 1:23]3   6.49027   10.73795   0.604    0.551
Q.wks3_7[, 1:23]4  -0.48003    0.61983  -0.774    0.445
Q.wks3_7[, 1:23]5   0.29334    1.20948   0.243    0.810
Q.wks3_7[, 1:23]6   3.93049    4.62928   0.849    0.403
Q.wks3_7[, 1:23]7  -4.73579    3.77309  -1.255    0.220
Q.wks3_7[, 1:23]8   1.90476    1.50177   1.268    0.216
Q.wks3_7[, 1:23]9  -0.49542    1.01392  -0.489    0.629
Q.wks3_7[, 1:23]10  0.07366    0.06300   1.169    0.252
Q.wks3_7[, 1:23]11 -0.03071    0.12475  -0.246    0.807
Q.wks3_7[, 1:23]12 -0.01960    0.35236  -0.056    0.956
Q.wks3_7[, 1:23]13 -0.11995    0.14765  -0.812    0.424
Q.wks3_7[, 1:23]14  0.11673    0.08842   1.320    0.198
Q.wks3_7[, 1:23]15 -0.22710    0.32628  -0.696    0.492
Q.wks3_7[, 1:23]16  0.20680    0.37341   0.554    0.584
Q.wks3_7[, 1:23]17 -0.03797    0.11838  -0.321    0.751
Q.wks3_7[, 1:23]18 -0.04733    0.09707  -0.488    0.630
Q.wks3_7[, 1:23]19  0.04334    0.18715   0.232    0.819
Q.wks3_7[, 1:23]20  0.25319    0.19017   1.331    0.194
Q.wks3_7[, 1:23]21 -0.17753    0.13035  -1.362    0.184
Q.wks3_7[, 1:23]22 -0.16028    0.13564  -1.182    0.248
```

```
Q.wks3_7[, 1:23]23  0.11874    0.13772   0.862    0.396

Residual standard error: 0.1733 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9998
F-statistic:  8722 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y8 :

Call:
lm(formula = Y8 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-5.043e-01 -1.698e-02  1.353e-14  2.118e-02  5.043e-01

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  -35.52468   11.77779  -3.016 0.005521 **
Q.wks3_7[, 1:23]2  -40.36369   12.59653  -3.204 0.003462 **
Q.wks3_7[, 1:23]3  -39.35459   12.49255  -3.150 0.003963 **
Q.wks3_7[, 1:23]4   -0.40828    0.72111  -0.566 0.575952
Q.wks3_7[, 1:23]5   10.83501    1.40712   7.700 2.79e-08 ***
Q.wks3_7[, 1:23]6  -22.17063    5.38571  -4.117 0.000325 ***
Q.wks3_7[, 1:23]7   18.59820    4.38962   4.237 0.000236 ***
Q.wks3_7[, 1:23]8  -10.37697    1.74716  -5.939 2.48e-06 ***
Q.wks3_7[, 1:23]9    7.90261    1.17960   6.699 3.43e-07 ***
Q.wks3_7[, 1:23]10  -0.12366    0.07329  -1.687 0.103084
Q.wks3_7[, 1:23]11  -0.17215    0.14513  -1.186 0.245885
Q.wks3_7[, 1:23]12   1.21339    0.40994   2.960 0.006337 **
Q.wks3_7[, 1:23]13  -1.64608    0.17178  -9.582 3.52e-10 ***
Q.wks3_7[, 1:23]14   0.80609    0.10287   7.836 2.00e-08 ***
Q.wks3_7[, 1:23]15   0.65083    0.37960   1.715 0.097890 .
Q.wks3_7[, 1:23]16  -0.85200    0.43443  -1.961 0.060247 .
Q.wks3_7[, 1:23]17   0.43108    0.13773   3.130 0.004169 **
Q.wks3_7[, 1:23]18  -0.71920    0.11293  -6.369 8.05e-07 ***
Q.wks3_7[, 1:23]19  -1.01871    0.21774  -4.679 7.22e-05 ***
Q.wks3_7[, 1:23]20   0.94478    0.22124   4.270 0.000216 ***
Q.wks3_7[, 1:23]21   0.77054    0.15165   5.081 2.45e-05 ***
Q.wks3_7[, 1:23]22   0.85405    0.15780   5.412 1.01e-05 ***
Q.wks3_7[, 1:23]23  -1.25199    0.16023  -7.814 2.11e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2016 on 27 degrees of freedom
```

```
Multiple R-squared: 0.9986,Adjusted R-squared: 0.9974 
F-statistic:   826 on 23 and 27 DF,  p-value: < 2.2e-16 


Response Y9 :

Call:
lm(formula = Y9 ˜ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max 
-4.342e-02 -8.557e-03 -4.083e-05  8.313e-03  4.342e-02 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)  
Q.wks3_7[, 1:23]1   2.919876   1.346744   2.168   0.0391 *
Q.wks3_7[, 1:23]2   3.044390   1.440364   2.114   0.0439 *
Q.wks3_7[, 1:23]3   2.922446   1.428475   2.046   0.0506 .
Q.wks3_7[, 1:23]4   0.061339   0.082457   0.744   0.4634  
Q.wks3_7[, 1:23]5   0.129719   0.160898   0.806   0.4272  
Q.wks3_7[, 1:23]6  -0.913068   0.615835  -1.483   0.1497  
Q.wks3_7[, 1:23]7   0.767532   0.501935   1.529   0.1379  
Q.wks3_7[, 1:23]8  -0.240670   0.199781  -1.205   0.2388  
Q.wks3_7[, 1:23]9   0.109755   0.134882   0.814   0.4229  
Q.wks3_7[, 1:23]10 -0.013932   0.008380  -1.662   0.1080  
Q.wks3_7[, 1:23]11 -0.020772   0.016595  -1.252   0.2214  
Q.wks3_7[, 1:23]12  0.071905   0.046875   1.534   0.1367  
Q.wks3_7[, 1:23]13 -0.033063   0.019643  -1.683   0.1039  
Q.wks3_7[, 1:23]14 -0.005003   0.011763  -0.425   0.6740  
Q.wks3_7[, 1:23]15  0.065339   0.043405   1.505   0.1439  
Q.wks3_7[, 1:23]16 -0.071417   0.049675  -1.438   0.1620  
Q.wks3_7[, 1:23]17  0.007644   0.015749   0.485   0.6313  
Q.wks3_7[, 1:23]18  0.006850   0.012913   0.530   0.6001  
Q.wks3_7[, 1:23]19 -0.037382   0.024897  -1.501   0.1448  
Q.wks3_7[, 1:23]20  0.020080   0.025298   0.794   0.4343  
Q.wks3_7[, 1:23]21  0.011048   0.017341   0.637   0.5294  
Q.wks3_7[, 1:23]22  0.016235   0.018044   0.900   0.3762  
Q.wks3_7[, 1:23]23 -0.015768   0.018322  -0.861   0.3970  
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1 

Residual standard error: 0.02305 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9999 
F-statistic: 1.638e+04 on 23 and 27 DF,  p-value: < 2.2e-16 
```

```
Response Y10 :

Call:
lm(formula = Y10 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max 
-6.819e-01 -1.397e-01  1.932e-13  1.692e-01  4.830e-01 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
Q.wks3_7[, 1:23]1   50.5622    18.1601   2.784 0.009682 ** 
Q.wks3_7[, 1:23]2   57.8483    19.4225   2.978 0.006058 ** 
Q.wks3_7[, 1:23]3   54.1999    19.2622   2.814 0.009022 ** 
Q.wks3_7[, 1:23]4    5.4504     1.1119   4.902 3.96e-05 ***
Q.wks3_7[, 1:23]5   -0.2265     2.1696  -0.104 0.917614    
Q.wks3_7[, 1:23]6  -60.5969     8.3042  -7.297 7.54e-08 ***
Q.wks3_7[, 1:23]7   69.7684     6.7683  10.308 7.38e-11 ***
Q.wks3_7[, 1:23]8  -32.1010     2.6939 -11.916 2.91e-12 ***
Q.wks3_7[, 1:23]9   13.9749     1.8188   7.683 2.90e-08 ***
Q.wks3_7[, 1:23]10  -1.0788     0.1130  -9.546 3.81e-10 ***
Q.wks3_7[, 1:23]11   0.2803     0.2238   1.253 0.221070    
Q.wks3_7[, 1:23]12   2.2462     0.6321   3.554 0.001422 ** 
Q.wks3_7[, 1:23]13  -0.9689     0.2649  -3.658 0.001086 ** 
Q.wks3_7[, 1:23]14  -0.6005     0.1586  -3.786 0.000778 ***
Q.wks3_7[, 1:23]15   3.1500     0.5853   5.382 1.09e-05 ***
Q.wks3_7[, 1:23]16  -2.2087     0.6698  -3.297 0.002739 ** 
Q.wks3_7[, 1:23]17  -0.3002     0.2124  -1.414 0.168872    
Q.wks3_7[, 1:23]18   0.5140     0.1741   2.952 0.006463 ** 
Q.wks3_7[, 1:23]19  -3.2196     0.3357  -9.590 3.46e-10 ***
Q.wks3_7[, 1:23]20   0.7393     0.3411   2.167 0.039205 *  
Q.wks3_7[, 1:23]21   1.5867     0.2338   6.786 2.75e-07 ***
Q.wks3_7[, 1:23]22   1.9335     0.2433   7.947 1.53e-08 ***
Q.wks3_7[, 1:23]23  -2.0088     0.2471  -8.131 9.83e-09 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1 

Residual standard error: 0.3109 on 27 degrees of freedom
Multiple R-squared: 0.9995,Adjusted R-squared: 0.9991 
F-statistic:  2410 on 23 and 27 DF,  p-value: < 2.2e-16 


Response Y11 :
```

```
Call:
lm(formula = Y11 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-1.921e-01 -6.634e-02  1.767e-15  5.513e-02  2.453e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  28.46115    7.45361   3.818 0.000714 ***
Q.wks3_7[, 1:23]2  30.39943    7.97176   3.813 0.000724 ***
Q.wks3_7[, 1:23]3  28.53249    7.90595   3.609 0.001233 **
Q.wks3_7[, 1:23]4   1.22104    0.45636   2.676 0.012519 *
Q.wks3_7[, 1:23]5  -1.21778    0.89050  -1.368 0.182738
Q.wks3_7[, 1:23]6  -9.02639    3.40837  -2.648 0.013346 *
Q.wks3_7[, 1:23]7   9.40741    2.77798   3.386 0.002185 **
Q.wks3_7[, 1:23]8  -2.93838    1.10570  -2.657 0.013062 *
Q.wks3_7[, 1:23]9   0.86939    0.74651   1.165 0.254366
Q.wks3_7[, 1:23]10 -0.17838    0.04638  -3.846 0.000664 ***
Q.wks3_7[, 1:23]11 -0.01763    0.09185  -0.192 0.849207
Q.wks3_7[, 1:23]12  0.29290    0.25943   1.129 0.268831
Q.wks3_7[, 1:23]13  0.13945    0.10871   1.283 0.210484
Q.wks3_7[, 1:23]14 -0.28061    0.06510  -4.310 0.000194 ***
Q.wks3_7[, 1:23]15  0.64026    0.24023   2.665 0.012828 *
Q.wks3_7[, 1:23]16 -0.55724    0.27493  -2.027 0.052657 .
Q.wks3_7[, 1:23]17  0.00489    0.08716   0.056 0.955675
Q.wks3_7[, 1:23]18  0.18099    0.07147   2.533 0.017446 *
Q.wks3_7[, 1:23]19 -0.23563    0.13780  -1.710 0.098734 .
Q.wks3_7[, 1:23]20 -0.35059    0.14001  -2.504 0.018623 *
Q.wks3_7[, 1:23]21  0.29079    0.09597   3.030 0.005339 **
Q.wks3_7[, 1:23]22  0.30196    0.09987   3.024 0.005422 **
Q.wks3_7[, 1:23]23 -0.19503    0.10140  -1.923 0.065038 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1276 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9999
F-statistic: 1.494e+04 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y12 :

Call:
lm(formula = Y12 ~ 0 + Q.wks3_7[, 1:23])
```

```
Residuals:
       Min         1Q     Median         3Q        Max 
-6.736e-01 -1.320e-01 -2.673e-13  1.309e-01  5.728e-01 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
Q.wks3_7[, 1:23]1   41.18000   19.96496   2.063 0.048890 *  
Q.wks3_7[, 1:23]2   44.23740   21.35284   2.072 0.047968 *  
Q.wks3_7[, 1:23]3   44.84897   21.17658   2.118 0.043539 *  
Q.wks3_7[, 1:23]4    5.41697    1.22239   4.431 0.000140 ***
Q.wks3_7[, 1:23]5   -0.33743    2.38525  -0.141 0.888552    
Q.wks3_7[, 1:23]6  -52.77004    9.12952  -5.780 3.78e-06 ***
Q.wks3_7[, 1:23]7   60.32648    7.44100   8.107 1.04e-08 ***
Q.wks3_7[, 1:23]8  -28.49325    2.96167  -9.621 3.23e-10 ***
Q.wks3_7[, 1:23]9   12.86614    1.99958   6.434 6.79e-07 ***
Q.wks3_7[, 1:23]10  -0.85130    0.12424  -6.852 2.32e-07 ***
Q.wks3_7[, 1:23]11   0.10926    0.24602   0.444 0.660498    
Q.wks3_7[, 1:23]12   2.11356    0.69490   3.042 0.005188 ** 
Q.wks3_7[, 1:23]13  -1.05495    0.29119  -3.623 0.001189 ** 
Q.wks3_7[, 1:23]14  -0.46220    0.17438  -2.651 0.013276 *  
Q.wks3_7[, 1:23]15   2.79388    0.64347   4.342 0.000178 ***
Q.wks3_7[, 1:23]16  -2.28189    0.73642  -3.099 0.004505 ** 
Q.wks3_7[, 1:23]17   0.02237    0.23347   0.096 0.924382    
Q.wks3_7[, 1:23]18   0.38420    0.19142   2.007 0.054853 .  
Q.wks3_7[, 1:23]19  -2.59236    0.36909  -7.024 1.50e-07 ***
Q.wks3_7[, 1:23]20   0.52103    0.37503   1.389 0.176100    
Q.wks3_7[, 1:23]21   1.36693    0.25707   5.317 1.30e-05 ***
Q.wks3_7[, 1:23]22   1.74641    0.26750   6.529 5.32e-07 ***
Q.wks3_7[, 1:23]23  -1.76357    0.27161  -6.493 5.83e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1 

Residual standard error: 0.3418 on 27 degrees of freedom
Multiple R-squared: 0.9992,Adjusted R-squared: 0.9986 
F-statistic:  1511 on 23 and 27 DF,  p-value: < 2.2e-16 


Response Y13 :

Call:
lm(formula = Y13 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max 
-3.842e-01 -7.952e-02  1.247e-13  9.909e-02  2.566e-01 
```

```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  12.59495   10.72366   1.175    0.250
Q.wks3_7[, 1:23]2  13.77066   11.46912   1.201    0.240
Q.wks3_7[, 1:23]3  13.95991   11.37445   1.227    0.230
Q.wks3_7[, 1:23]4   0.04151    0.65657   0.063    0.950
Q.wks3_7[, 1:23]5   1.15612    1.28118   0.902    0.375
Q.wks3_7[, 1:23]6  -6.96323    4.90368  -1.420    0.167
Q.wks3_7[, 1:23]7   6.30598    3.99674   1.578    0.126
Q.wks3_7[, 1:23]8  -1.29134    1.59078  -0.812    0.424
Q.wks3_7[, 1:23]9   0.19306    1.07402   0.180    0.859
Q.wks3_7[, 1:23]10 -0.09998    0.06673  -1.498    0.146
Q.wks3_7[, 1:23]11 -0.10577    0.13214  -0.800    0.430
Q.wks3_7[, 1:23]12  0.44544    0.37325   1.193    0.243
Q.wks3_7[, 1:23]13 -0.12117    0.15641  -0.775    0.445
Q.wks3_7[, 1:23]14 -0.10953    0.09366  -1.169    0.252
Q.wks3_7[, 1:23]15  0.43306    0.34562   1.253    0.221
Q.wks3_7[, 1:23]16 -0.44710    0.39555  -1.130    0.268
Q.wks3_7[, 1:23]17 -0.01231    0.12540  -0.098    0.922
Q.wks3_7[, 1:23]18  0.08077    0.10282   0.786    0.439
Q.wks3_7[, 1:23]19 -0.27761    0.19825  -1.400    0.173
Q.wks3_7[, 1:23]20  0.10798    0.20144   0.536    0.596
Q.wks3_7[, 1:23]21  0.14376    0.13808   1.041    0.307
Q.wks3_7[, 1:23]22  0.08720    0.14368   0.607    0.549
Q.wks3_7[, 1:23]23 -0.11098    0.14589  -0.761    0.453

Residual standard error: 0.1836 on 27 degrees of freedom
Multiple R-squared: 0.9997,Adjusted R-squared: 0.9995
F-statistic:  4376 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y14 :

Call:
lm(formula = Y14 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-5.288e-01 -6.257e-02  1.475e-15  8.927e-02  5.288e-01

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   18.14671   16.07258   1.129 0.268810
Q.wks3_7[, 1:23]2   20.35633   17.18988   1.184 0.246653
```

```
Q.wks3_7[, 1:23]3   21.43305   17.04799   1.257 0.219441    
Q.wks3_7[, 1:23]4    1.38182    0.98407   1.404 0.171660    
Q.wks3_7[, 1:23]5   -0.07492    1.92022  -0.039 0.969165    
Q.wks3_7[, 1:23]6  -20.47375    7.34962  -2.786 0.009649 ** 
Q.wks3_7[, 1:23]7   24.03799    5.99030   4.013 0.000428 ***
Q.wks3_7[, 1:23]8  -10.42288    2.38426  -4.372 0.000165 ***
Q.wks3_7[, 1:23]9    4.43870    1.60974   2.757 0.010321 *  
Q.wks3_7[, 1:23]10  -0.35664    0.10002  -3.566 0.001379 ** 
Q.wks3_7[, 1:23]11   0.21800    0.19805   1.101 0.280735    
Q.wks3_7[, 1:23]12   0.59325    0.55942   1.060 0.298325    
Q.wks3_7[, 1:23]13  -0.22401    0.23442  -0.956 0.347761    
Q.wks3_7[, 1:23]14  -0.25897    0.14038  -1.845 0.076066 .  
Q.wks3_7[, 1:23]15   0.98173    0.51802   1.895 0.068823 .  
Q.wks3_7[, 1:23]16  -0.57158    0.59285  -0.964 0.343539    
Q.wks3_7[, 1:23]17  -0.24841    0.18795  -1.322 0.197376    
Q.wks3_7[, 1:23]18   0.23215    0.15410   1.506 0.143564    
Q.wks3_7[, 1:23]19  -1.13012    0.29713  -3.803 0.000743 ***
Q.wks3_7[, 1:23]20   0.30095    0.30192   0.997 0.327706    
Q.wks3_7[, 1:23]21   0.47966    0.20695   2.318 0.028282 *  
Q.wks3_7[, 1:23]22   0.61091    0.21534   2.837 0.008536 ** 
Q.wks3_7[, 1:23]23  -0.60046    0.21866  -2.746 0.010601 *  
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1 

Residual standard error: 0.2751 on 27 degrees of freedom
Multiple R-squared: 0.9997,Adjusted R-squared: 0.9994 
F-statistic:  3388 on 23 and 27 DF,  p-value: < 2.2e-16 


Response Y15 :

Call:
lm(formula = Y15 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max 
-1.433e-01 -4.584e-02 -3.181e-14  3.674e-02  1.836e-01 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
Q.wks3_7[, 1:23]1  -5.35159    5.34565  -1.001 0.325662    
Q.wks3_7[, 1:23]2  -5.29977    5.71726  -0.927 0.362151    
Q.wks3_7[, 1:23]3  -5.36999    5.67007  -0.947 0.352000    
Q.wks3_7[, 1:23]4   0.19374    0.32730   0.592 0.558822    
Q.wks3_7[, 1:23]5  -1.13470    0.63866  -1.777 0.086887 .  
```

```
Q.wks3_7[, 1:23]6   4.43386    2.44444   1.814 0.080830 .
Q.wks3_7[, 1:23]7  -1.48442    1.99234  -0.745 0.462671
Q.wks3_7[, 1:23]8  -0.54909    0.79299  -0.692 0.494578
Q.wks3_7[, 1:23]9   0.07464    0.53539   0.139 0.890155
Q.wks3_7[, 1:23]10  0.02745    0.03326   0.825 0.416469
Q.wks3_7[, 1:23]11  0.21763    0.06587   3.304 0.002694 **
Q.wks3_7[, 1:23]12 -0.51839    0.18606  -2.786 0.009638 **
Q.wks3_7[, 1:23]13  0.22805    0.07797   2.925 0.006901 **
Q.wks3_7[, 1:23]14  0.03239    0.04669   0.694 0.493747
Q.wks3_7[, 1:23]15 -0.37842    0.17229  -2.196 0.036829 *
Q.wks3_7[, 1:23]16  0.63377    0.19718   3.214 0.003377 **
Q.wks3_7[, 1:23]17 -0.24524    0.06251  -3.923 0.000542 ***
Q.wks3_7[, 1:23]18  0.01372    0.05125   0.268 0.790936
Q.wks3_7[, 1:23]19 -0.02064    0.09882  -0.209 0.836113
Q.wks3_7[, 1:23]20  0.03212    0.10042   0.320 0.751534
Q.wks3_7[, 1:23]21 -0.03859    0.06883  -0.561 0.579661
Q.wks3_7[, 1:23]22  0.01656    0.07162   0.231 0.818840
Q.wks3_7[, 1:23]23 -0.03262    0.07272  -0.449 0.657344
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.09151 on 27 degrees of freedom
Multiple R-squared:     1,Adjusted R-squared: 0.9999
F-statistic: 4.142e+04 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y16 :

Call:
lm(formula = Y16 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-5.011e-01 -1.615e-01 -4.816e-14  1.615e-01  5.376e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   20.9981    21.5030   0.977 0.337474
Q.wks3_7[, 1:23]2   24.7925    22.9978   1.078 0.290554
Q.wks3_7[, 1:23]3   23.5659    22.8079   1.033 0.310662
Q.wks3_7[, 1:23]4    5.3008     1.3166   4.026 0.000413 ***
Q.wks3_7[, 1:23]5    2.9640     2.5690   1.154 0.258719
Q.wks3_7[, 1:23]6  -50.2544     9.8328  -5.111 2.26e-05 ***
Q.wks3_7[, 1:23]7   54.0744     8.0142   6.747 3.03e-07 ***
Q.wks3_7[, 1:23]8  -25.7251     3.1898  -8.065 1.15e-08 ***
```

```
Q.wks3_7[, 1:23]9   12.3009     2.1536   5.712 4.54e-06 ***
Q.wks3_7[, 1:23]10  -0.8019     0.1338  -5.993 2.16e-06 ***
Q.wks3_7[, 1:23]11  -0.2581     0.2650  -0.974 0.338664
Q.wks3_7[, 1:23]12   2.4850     0.7484   3.320 0.002585 **
Q.wks3_7[, 1:23]13  -1.2753     0.3136  -4.066 0.000371 ***
Q.wks3_7[, 1:23]14  -0.2694     0.1878  -1.435 0.162882
Q.wks3_7[, 1:23]15   2.7834     0.6930   4.016 0.000424 ***
Q.wks3_7[, 1:23]16  -2.5185     0.7931  -3.175 0.003722 **
Q.wks3_7[, 1:23]17   0.2580     0.2515   1.026 0.313932
Q.wks3_7[, 1:23]18   0.1719     0.2062   0.834 0.411732
Q.wks3_7[, 1:23]19  -2.3354     0.3975  -5.875 2.94e-06 ***
Q.wks3_7[, 1:23]20   0.6217     0.4039   1.539 0.135431
Q.wks3_7[, 1:23]21   1.3392     0.2769   4.837 4.72e-05 ***
Q.wks3_7[, 1:23]22   1.5256     0.2881   5.295 1.38e-05 ***
Q.wks3_7[, 1:23]23  -1.7090     0.2925  -5.842 3.21e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.3681 on 27 degrees of freedom
Multiple R-squared: 0.999,Adjusted R-squared: 0.9981
F-statistic:  1122 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y17 :

Call:
lm(formula = Y17 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-2.545e-01 -4.073e-02  4.520e-15  4.664e-02  1.715e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  28.86275    6.65419   4.338 0.000180 ***
Q.wks3_7[, 1:23]2  29.73603    7.11676   4.178 0.000276 ***
Q.wks3_7[, 1:23]3  29.45126    7.05802   4.173 0.000280 ***
Q.wks3_7[, 1:23]4   0.30263    0.40741   0.743 0.464010
Q.wks3_7[, 1:23]5   1.06632    0.79499   1.341 0.190999
Q.wks3_7[, 1:23]6  -8.74650    3.04281  -2.874 0.007798 **
Q.wks3_7[, 1:23]7   5.82827    2.48004   2.350 0.026331 *
Q.wks3_7[, 1:23]8  -1.11169    0.98711  -1.126 0.269989
Q.wks3_7[, 1:23]9   0.95967    0.66645   1.440 0.161366
Q.wks3_7[, 1:23]10 -0.08665    0.04141  -2.093 0.045905 *
Q.wks3_7[, 1:23]11 -0.21069    0.08200  -2.570 0.016023 *
```

```
Q.wks3_7[, 1:23]12  0.61150    0.23161   2.640 0.013599 *
Q.wks3_7[, 1:23]13 -0.19370    0.09705  -1.996 0.056133 .
Q.wks3_7[, 1:23]14 -0.12944    0.05812  -2.227 0.034473 *
Q.wks3_7[, 1:23]15  0.62910    0.21446   2.933 0.006761 **
Q.wks3_7[, 1:23]16 -0.83218    0.24544  -3.391 0.002162 **
Q.wks3_7[, 1:23]17  0.18814    0.07781   2.418 0.022632 *
Q.wks3_7[, 1:23]18  0.06369    0.06380   0.998 0.326982
Q.wks3_7[, 1:23]19 -0.09921    0.12302  -0.806 0.427018
Q.wks3_7[, 1:23]20 -0.13392    0.12500  -1.071 0.293468
Q.wks3_7[, 1:23]21  0.16350    0.08568   1.908 0.067039 .
Q.wks3_7[, 1:23]22  0.17526    0.08915   1.966 0.059687 .
Q.wks3_7[, 1:23]23 -0.10983    0.09053  -1.213 0.235527
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1139 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9999
F-statistic: 1.501e+04 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y18 :

Call:
lm(formula = Y18 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-1.952e-01 -4.445e-02 -5.670e-14  4.445e-02  1.952e-01

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   9.909333   7.022891   1.411  0.16966
Q.wks3_7[, 1:23]2   9.157078   7.511095   1.219  0.23334
Q.wks3_7[, 1:23]3  10.004246   7.449094   1.343  0.19045
Q.wks3_7[, 1:23]4   0.274855   0.429988   0.639  0.52807
Q.wks3_7[, 1:23]5  -2.446399   0.839039  -2.916  0.00706 **
Q.wks3_7[, 1:23]6   6.935713   3.211407   2.160  0.03984 *
Q.wks3_7[, 1:23]7  -5.505424   2.617452  -2.103  0.04489 *
Q.wks3_7[, 1:23]8   2.084277   1.041800   2.001  0.05558 .
Q.wks3_7[, 1:23]9  -1.353108   0.703374  -1.924  0.06499 .
Q.wks3_7[, 1:23]10  0.045482   0.043702   1.041  0.30723
Q.wks3_7[, 1:23]11  0.103405   0.086539   1.195  0.24252
Q.wks3_7[, 1:23]12 -0.385221   0.244438  -1.576  0.12668
Q.wks3_7[, 1:23]13  0.222389   0.102430   2.171  0.03888 *
Q.wks3_7[, 1:23]14 -0.001671   0.061339  -0.027  0.97847
```

```
Q.wks3_7[, 1:23]15 -0.345677   0.226346  -1.527  0.13834
Q.wks3_7[, 1:23]16  0.577015   0.259043   2.227  0.03444 *
Q.wks3_7[, 1:23]17 -0.243188   0.082125  -2.961  0.00632 **
Q.wks3_7[, 1:23]18  0.080989   0.067336   1.203  0.23951
Q.wks3_7[, 1:23]19  0.049490   0.129832   0.381  0.70605
Q.wks3_7[, 1:23]20  0.126722   0.131922   0.961  0.34529
Q.wks3_7[, 1:23]21 -0.235110   0.090428  -2.600  0.01493 *
Q.wks3_7[, 1:23]22 -0.198231   0.094095  -2.107  0.04457 *
Q.wks3_7[, 1:23]23  0.207254   0.095542   2.169  0.03904 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1202 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9999
F-statistic: 1.958e+04 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y19 :

Call:
lm(formula = Y19 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min          1Q     Median          3Q         Max
-2.462e-01 -2.438e-02  4.074e-14  3.390e-02  2.037e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  -0.61898    7.41318  -0.083   0.9341
Q.wks3_7[, 1:23]2  -0.73365    7.92851  -0.093   0.9270
Q.wks3_7[, 1:23]3  -0.64265    7.86307  -0.082   0.9355
Q.wks3_7[, 1:23]4   0.31316    0.45388   0.690   0.4961
Q.wks3_7[, 1:23]5   0.71785    0.88567   0.811   0.4247
Q.wks3_7[, 1:23]6  -0.46070    3.38988  -0.136   0.8929
Q.wks3_7[, 1:23]7   0.40892    2.76291   0.148   0.8834
Q.wks3_7[, 1:23]8  -0.69475    1.09970  -0.632   0.5329
Q.wks3_7[, 1:23]9   0.59000    0.74246   0.795   0.4337
Q.wks3_7[, 1:23]10 -0.01981    0.04613  -0.429   0.6710
Q.wks3_7[, 1:23]11 -0.16611    0.09135  -1.818   0.0801 .
Q.wks3_7[, 1:23]12  0.35817    0.25802   1.388   0.1764
Q.wks3_7[, 1:23]13 -0.25305    0.10812  -2.340   0.0269 *
Q.wks3_7[, 1:23]14  0.07945    0.06475   1.227   0.2304
Q.wks3_7[, 1:23]15  0.13970    0.23893   0.585   0.5636
Q.wks3_7[, 1:23]16 -0.15380    0.27344  -0.562   0.5784
Q.wks3_7[, 1:23]17  0.02131    0.08669   0.246   0.8076
```

```
Q.wks3_7[, 1:23]18 -0.03522    0.07108  -0.496   0.6242
Q.wks3_7[, 1:23]19 -0.19560    0.13705  -1.427   0.1650
Q.wks3_7[, 1:23]20  0.23325    0.13925   1.675   0.1055
Q.wks3_7[, 1:23]21  0.04431    0.09545   0.464   0.6462
Q.wks3_7[, 1:23]22  0.05709    0.09932   0.575   0.5702
Q.wks3_7[, 1:23]23 -0.12943    0.10085  -1.283   0.2103
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1269 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9998
F-statistic: 1.177e+04 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y20 :

Call:
lm(formula = Y20 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-8.385e-01 -4.491e-02 -3.331e-15  3.747e-02  4.674e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   4.15964   13.09689   0.318    0.753
Q.wks3_7[, 1:23]2   2.55406   14.00734   0.182    0.857
Q.wks3_7[, 1:23]3   3.47299   13.89171   0.250    0.804
Q.wks3_7[, 1:23]4  -0.49151    0.80188  -0.613    0.545
Q.wks3_7[, 1:23]5   0.72155    1.56471   0.461    0.648
Q.wks3_7[, 1:23]6   1.24666    5.98891   0.208    0.837
Q.wks3_7[, 1:23]7  -0.94388    4.88125  -0.193    0.848
Q.wks3_7[, 1:23]8  -0.37079    1.94284  -0.191    0.850
Q.wks3_7[, 1:23]9   0.71336    1.31171   0.544    0.591
Q.wks3_7[, 1:23]10  0.03672    0.08150   0.451    0.656
Q.wks3_7[, 1:23]11  0.14169    0.16139   0.878    0.388
Q.wks3_7[, 1:23]12 -0.24514    0.45585  -0.538    0.595
Q.wks3_7[, 1:23]13 -0.03492    0.19102  -0.183    0.856
Q.wks3_7[, 1:23]14  0.12118    0.11439   1.059    0.299
Q.wks3_7[, 1:23]15 -0.23784    0.42211  -0.563    0.578
Q.wks3_7[, 1:23]16  0.32844    0.48309   0.680    0.502
Q.wks3_7[, 1:23]17 -0.09573    0.15315  -0.625    0.537
Q.wks3_7[, 1:23]18 -0.07690    0.12557  -0.612    0.545
Q.wks3_7[, 1:23]19 -0.03307    0.24212  -0.137    0.892
Q.wks3_7[, 1:23]20  0.10887    0.24602   0.443    0.662
```

```
Q.wks3_7[, 1:23]21 -0.01161    0.16864  -0.069    0.946
Q.wks3_7[, 1:23]22  0.05594    0.17548   0.319    0.752
Q.wks3_7[, 1:23]23 -0.07851    0.17817  -0.441    0.663

Residual standard error: 0.2242 on 27 degrees of freedom
Multiple R-squared: 0.9998,Adjusted R-squared: 0.9997
F-statistic:  6685 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y21 :

Call:
lm(formula = Y21 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-3.889e-02 -1.096e-02 -9.901e-15  1.222e-02  3.873e-02

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   2.621272   1.356409   1.933   0.0639 .
Q.wks3_7[, 1:23]2   2.699846   1.450701   1.861   0.0737 .
Q.wks3_7[, 1:23]3   2.593648   1.438726   1.803   0.0826 .
Q.wks3_7[, 1:23]4   0.067007   0.083048   0.807   0.4268
Q.wks3_7[, 1:23]5  -0.011326   0.162053  -0.070   0.9448
Q.wks3_7[, 1:23]6  -0.508065   0.620255  -0.819   0.4199
Q.wks3_7[, 1:23]7   0.446710   0.505538   0.884   0.3847
Q.wks3_7[, 1:23]8  -0.136379   0.201214  -0.678   0.5037
Q.wks3_7[, 1:23]9   0.066695   0.135850   0.491   0.6274
Q.wks3_7[, 1:23]10 -0.011169   0.008441  -1.323   0.1969
Q.wks3_7[, 1:23]11 -0.011772   0.016714  -0.704   0.4873
Q.wks3_7[, 1:23]12  0.048224   0.047211   1.021   0.3161
Q.wks3_7[, 1:23]13 -0.023193   0.019783  -1.172   0.2513
Q.wks3_7[, 1:23]14 -0.003664   0.011847  -0.309   0.7595
Q.wks3_7[, 1:23]15  0.043022   0.043717   0.984   0.3338
Q.wks3_7[, 1:23]16 -0.037598   0.050032  -0.751   0.4589
Q.wks3_7[, 1:23]17 -0.002309   0.015862  -0.146   0.8853
Q.wks3_7[, 1:23]18  0.009121   0.013005   0.701   0.4891
Q.wks3_7[, 1:23]19 -0.034209   0.025076  -1.364   0.1838
Q.wks3_7[, 1:23]20  0.021662   0.025480   0.850   0.4027
Q.wks3_7[, 1:23]21  0.002002   0.017465   0.115   0.9096
Q.wks3_7[, 1:23]22  0.010513   0.018174   0.578   0.5677
Q.wks3_7[, 1:23]23 -0.009123   0.018453  -0.494   0.6250
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
Residual standard error: 0.02322 on 27 degrees of freedom
Multiple R-squared: 0.9999,Adjusted R-squared: 0.9998
F-statistic: 1.264e+04 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y22 :

Call:
lm(formula = Y22 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-5.283e-01 -1.323e-01  1.264e-13  1.344e-01  5.235e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   15.5326    17.9247   0.867  0.39383
Q.wks3_7[, 1:23]2   20.2086    19.1708   1.054  0.30116
Q.wks3_7[, 1:23]3   17.6566    19.0125   0.929  0.36128
Q.wks3_7[, 1:23]4    3.7921     1.0975   3.455  0.00183 **
Q.wks3_7[, 1:23]5   -0.5216     2.1415  -0.244  0.80940
Q.wks3_7[, 1:23]6  -41.8799     8.1966  -5.109 2.27e-05 ***
Q.wks3_7[, 1:23]7   53.1504     6.6806   7.956 1.50e-08 ***
Q.wks3_7[, 1:23]8  -26.3224     2.6590  -9.899 1.76e-10 ***
Q.wks3_7[, 1:23]9   11.3010     1.7952   6.295 9.76e-07 ***
Q.wks3_7[, 1:23]10  -0.7741     0.1115  -6.940 1.85e-07 ***
Q.wks3_7[, 1:23]11   0.5822     0.2209   2.636  0.01374 *
Q.wks3_7[, 1:23]12   0.9473     0.6239   1.518  0.14054
Q.wks3_7[, 1:23]13  -0.5179     0.2614  -1.981  0.05786 .
Q.wks3_7[, 1:23]14  -0.3132     0.1566  -2.001  0.05558 .
Q.wks3_7[, 1:23]15   1.8670     0.5777   3.232  0.00323 **
Q.wks3_7[, 1:23]16  -0.9571     0.6612  -1.448  0.15924
Q.wks3_7[, 1:23]17  -0.3228     0.2096  -1.540  0.13516
Q.wks3_7[, 1:23]18   0.2438     0.1719   1.418  0.16754
Q.wks3_7[, 1:23]19  -2.3784     0.3314  -7.177 1.02e-07 ***
Q.wks3_7[, 1:23]20   0.3786     0.3367   1.124  0.27078
Q.wks3_7[, 1:23]21   1.3051     0.2308   5.654 5.28e-06 ***
Q.wks3_7[, 1:23]22   1.6214     0.2402   6.751 3.00e-07 ***
Q.wks3_7[, 1:23]23  -1.6855     0.2439  -6.912 1.99e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.3069 on 27 degrees of freedom
Multiple R-squared: 0.9995,Adjusted R-squared: 0.9991
```

```
F-statistic:  2409 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y23 :

Call:
lm(formula = Y23 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-8.603e-01 -7.831e-02 -2.235e-13  7.682e-02  8.603e-01

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1  125.3643    22.4820   5.576 6.51e-06 ***
Q.wks3_7[, 1:23]2  136.7157    24.0449   5.686 4.86e-06 ***
Q.wks3_7[, 1:23]3  137.5655    23.8464   5.769 3.90e-06 ***
Q.wks3_7[, 1:23]4    9.2512     1.3765   6.721 3.24e-07 ***
Q.wks3_7[, 1:23]5    2.5891     2.6860   0.964  0.34363
Q.wks3_7[, 1:23]6  -76.5064    10.2805  -7.442 5.26e-08 ***
Q.wks3_7[, 1:23]7   61.6692     8.3791   7.360 6.45e-08 ***
Q.wks3_7[, 1:23]8  -15.8569     3.3351  -4.755 5.89e-05 ***
Q.wks3_7[, 1:23]9    6.9450     2.2517   3.084  0.00467 **
Q.wks3_7[, 1:23]10  -1.1024     0.1399  -7.880 1.80e-08 ***
Q.wks3_7[, 1:23]11  -2.3648     0.2770  -8.536 3.77e-09 ***
Q.wks3_7[, 1:23]12   7.1561     0.7825   9.145 9.32e-10 ***
Q.wks3_7[, 1:23]13  -3.0668     0.3279  -9.353 5.85e-10 ***
Q.wks3_7[, 1:23]14  -0.9084     0.1964  -4.626 8.31e-05 ***
Q.wks3_7[, 1:23]15   5.9409     0.7246   8.199 8.36e-09 ***
Q.wks3_7[, 1:23]16  -6.6286     0.8293  -7.993 1.37e-08 ***
Q.wks3_7[, 1:23]17   0.6612     0.2629   2.515  0.01817 *
Q.wks3_7[, 1:23]18   1.0648     0.2156   4.940 3.58e-05 ***
Q.wks3_7[, 1:23]19  -3.3559     0.4156  -8.074 1.13e-08 ***
Q.wks3_7[, 1:23]20   2.7359     0.4223   6.478 6.06e-07 ***
Q.wks3_7[, 1:23]21   0.2140     0.2895   0.739  0.46624
Q.wks3_7[, 1:23]22   0.3692     0.3012   1.226  0.23091
Q.wks3_7[, 1:23]23  -0.4492     0.3059  -1.469  0.15346
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.3849 on 27 degrees of freedom
Multiple R-squared: 0.9958,Adjusted R-squared: 0.9923
F-statistic: 279.9 on 23 and 27 DF,  p-value: < 2.2e-16
```

```
Response Y24 :

Call:
lm(formula = Y24 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
      Min        1Q    Median        3Q       Max
-0.942533 -0.037939  0.004024  0.088803  0.908841

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   10.29033   27.66191   0.372 0.712794
Q.wks3_7[, 1:23]2   15.39203   29.58486   0.520 0.607118
Q.wks3_7[, 1:23]3   11.30940   29.34064   0.385 0.702923
Q.wks3_7[, 1:23]4    3.58336    1.69365   2.116 0.043732 *
Q.wks3_7[, 1:23]5    1.14944    3.30482   0.348 0.730682
Q.wks3_7[, 1:23]6  -50.17912   12.64915  -3.967 0.000483 ***
Q.wks3_7[, 1:23]7   63.93174   10.30967   6.201 1.25e-06 ***
Q.wks3_7[, 1:23]8  -33.27613    4.10346  -8.109 1.04e-08 ***
Q.wks3_7[, 1:23]9   15.11591    2.77046   5.456 8.97e-06 ***
Q.wks3_7[, 1:23]10  -0.91796    0.17213  -5.333 1.25e-05 ***
Q.wks3_7[, 1:23]11   0.75741    0.34086   2.222 0.034850 *
Q.wks3_7[, 1:23]12   1.10791    0.96280   1.151 0.259936
Q.wks3_7[, 1:23]13  -0.75406    0.40345  -1.869 0.072505 .
Q.wks3_7[, 1:23]14  -0.24413    0.24161  -1.010 0.321242
Q.wks3_7[, 1:23]15   2.18248    0.89154   2.448 0.021149 *
Q.wks3_7[, 1:23]16  -1.23543    1.02032  -1.211 0.236455
Q.wks3_7[, 1:23]17  -0.12767    0.32348  -0.395 0.696177
Q.wks3_7[, 1:23]18   0.06699    0.26522   0.253 0.802504
Q.wks3_7[, 1:23]19  -2.77582    0.51138  -5.428 9.66e-06 ***
Q.wks3_7[, 1:23]20   0.09042    0.51962   0.174 0.863157
Q.wks3_7[, 1:23]21   1.87796    0.35618   5.273 1.47e-05 ***
Q.wks3_7[, 1:23]22   2.26524    0.37062   6.112 1.58e-06 ***
Q.wks3_7[, 1:23]23  -2.32495    0.37632  -6.178 1.33e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.4735 on 27 degrees of freedom
Multiple R-squared: 0.9987,Adjusted R-squared: 0.9977
F-statistic: 931.9 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y25 :

Call:
```

```
lm(formula = Y25 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-8.728e-02 -1.503e-02 -2.673e-15  1.889e-02  6.528e-02

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   3.396684   2.519204   1.348    0.189
Q.wks3_7[, 1:23]2   3.497972   2.694329   1.298    0.205
Q.wks3_7[, 1:23]3   3.288651   2.672088   1.231    0.229
Q.wks3_7[, 1:23]4   0.118955   0.154242   0.771    0.447
Q.wks3_7[, 1:23]5   0.097042   0.300974   0.322    0.750
Q.wks3_7[, 1:23]6  -1.132783   1.151974  -0.983    0.334
Q.wks3_7[, 1:23]7   1.082533   0.938915   1.153    0.259
Q.wks3_7[, 1:23]8  -0.463950   0.373707  -1.241    0.225
Q.wks3_7[, 1:23]9   0.237339   0.252310   0.941    0.355
Q.wks3_7[, 1:23]10 -0.021782   0.015676  -1.389    0.176
Q.wks3_7[, 1:23]11 -0.015922   0.031043  -0.513    0.612
Q.wks3_7[, 1:23]12  0.087089   0.087683   0.993    0.329
Q.wks3_7[, 1:23]13 -0.053753   0.036743  -1.463    0.155
Q.wks3_7[, 1:23]14  0.002111   0.022003   0.096    0.924
Q.wks3_7[, 1:23]15  0.076300   0.081193   0.940    0.356
Q.wks3_7[, 1:23]16 -0.063909   0.092922  -0.688    0.497
Q.wks3_7[, 1:23]17 -0.005837   0.029459  -0.198    0.844
Q.wks3_7[, 1:23]18  0.010878   0.024154   0.450    0.656
Q.wks3_7[, 1:23]19 -0.074711   0.046572  -1.604    0.120
Q.wks3_7[, 1:23]20  0.057675   0.047322   1.219    0.233
Q.wks3_7[, 1:23]21  0.004216   0.032438   0.130    0.898
Q.wks3_7[, 1:23]22  0.024543   0.033753   0.727    0.473
Q.wks3_7[, 1:23]23 -0.026036   0.034272  -0.760    0.454

Residual standard error: 0.04313 on 27 degrees of freedom
Multiple R-squared: 0.9998,Adjusted R-squared: 0.9997
F-statistic:  7705 on 23 and 27 DF,  p-value: < 2.2e-16


Response Y26 :

Call:
lm(formula = Y26 ~ 0 + Q.wks3_7[, 1:23])

Residuals:
       Min         1Q     Median         3Q        Max
-4.318e-01 -5.707e-02  6.383e-14  5.342e-02  4.318e-01
```

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
Q.wks3_7[, 1:23]1   15.63872   11.50050   1.360 0.185132
Q.wks3_7[, 1:23]2   14.12681   12.29997   1.149 0.260826
Q.wks3_7[, 1:23]3   16.84279   12.19844   1.381 0.178685
Q.wks3_7[, 1:23]4    0.33687    0.70414   0.478 0.636209
Q.wks3_7[, 1:23]5    2.23375    1.37399   1.626 0.115622
Q.wks3_7[, 1:23]6  -18.22929    5.25892  -3.466 0.001781 **
Q.wks3_7[, 1:23]7   18.61194    4.28627   4.342 0.000178 ***
Q.wks3_7[, 1:23]8   -7.75734    1.70602  -4.547 0.000103 ***
Q.wks3_7[, 1:23]9    4.40869    1.15183   3.828 0.000697 ***
Q.wks3_7[, 1:23]10  -0.17111    0.07157  -2.391 0.024037 *
Q.wks3_7[, 1:23]11   0.26717    0.14171   1.885 0.070192 .
Q.wks3_7[, 1:23]12   0.34069    0.40029   0.851 0.402192
Q.wks3_7[, 1:23]13  -0.50451    0.16774  -3.008 0.005637 **
Q.wks3_7[, 1:23]14   0.06538    0.10045   0.651 0.520620
Q.wks3_7[, 1:23]15   0.56697    0.37066   1.530 0.137743
Q.wks3_7[, 1:23]16  -0.47543    0.42420  -1.121 0.272257
Q.wks3_7[, 1:23]17   0.08524    0.13449   0.634 0.531543
Q.wks3_7[, 1:23]18  -0.11164    0.11027  -1.012 0.320325
Q.wks3_7[, 1:23]19  -0.70360    0.21261  -3.309 0.002657 **
Q.wks3_7[, 1:23]20   0.10433    0.21603   0.483 0.633036
Q.wks3_7[, 1:23]21   0.56956    0.14808   3.846 0.000664 ***
Q.wks3_7[, 1:23]22   0.66898    0.15409   4.342 0.000178 ***
Q.wks3_7[, 1:23]23  -0.69917    0.15646  -4.469 0.000127 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1969 on 27 degrees of freedom
Multiple R-squared: 0.9998,Adjusted R-squared: 0.9996
F-statistic:  5384 on 23 and 27 DF,  p-value: < 2.2e-16
```

Bibliography

[1] N. Angelopoulos and J. Cussens. Bayesian learning of Bayesian networks with informative priors. *Annals of Mathematics and Artificial Intelligence*, 54:53–98, 2008.

[2] S. Anjum, A. Doucet, and C. C. Holmes. A boosting approach to structure learning of graphs with and without prior knowledge. *Bioinformatics*, 25(22):2929–2936, 2009.

[3] L. Bottolo and S. Richardson. Evolutionary stochastic search. under revision at Journal of Computational and Graphical Statistics.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[5] B. M. Broom and D. Subramanian. Computational methods for learning Bayesian networks from high-throughput biological data. In K.-A. Do, P. Müller, and M. Vannucci, editors, *Bayesian Inference for Gene Expression and Proteomics*, pages 385–400. Cambridge University Press, 2006.

[6] R. Castelo and A. Roverato. A robust procedure for Gaussian graphical model search from microarray data with p larger than n. *Journal of Machine Learning Research*, 7:2621–2650, 2006.

[7] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137:43–90, 2002.

[8] D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 87–98, 1995.

[9] D. M. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, February 2002.

[10] D. M. Chickering, C. Meek, and D. Heckerman. Large-sample learning of Bayesian networks is NP-hard. In *Proceedings of the Ninteenth Conference on Uncertainty in Artificial Intelligence*, pages 124–133, 2003.

[11] B. G. Coombe. The regulation of set and development of the grape berry. *Acta Horticulturae*, 34:261–271, 1973.

[12] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[13] D. R. Cox and N. Wermuth. *Multivariate Dependencies.* Chapman & Hall, London, 1996.

[14] C. Davies. personal communication, January 2009.

[15] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration.* Academic Press, Inc., 2 edition, 1984.

[16] A. C. Davison. *Statistical Models.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1993.

[17] A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society, Series B*, 41(1):1–31, 1979.

[18] A. P. Dawid and S. L. Lauritzen. Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics*, 21(3):1272–1317, September 1993.

[19] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[20] A. de la Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574, 2004.

[21] A. P. Dempster. Covariance selection. *Biometrics*, 28(1):157–175, 1972.

[22] A. Dobra, L. Briollais, H. Jarjanazi, and H. Ozcelik. Applications of the mode oriented stochastic search (MOSS) algorithm for discrete multi-way data to genomewide studies. To appear in the volume *Bayesian Modeling in Bioinformatics*,Taylor & Francis (D. Dey, S. Ghosh and B. Mallick, eds.), 2008.

[23] A. Dobra, C. Hans, B. Jones, J.R. Nevins, and M. West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

[24] A. Dobra and H. Massam. The mode oriented stochastic search (MOSS) algorithm for log-linear models with conjugate priors. Submitted for Publication, 2008.

[25] A. Dobra and M. West. Bayesian covariance selection. ISDS Discussion Paper 04-23, Duke University, 2004.

[26] D. Edwards. Hierarchical interaction models. *Journal of the Royal Statistical Society, Series B*, 52(1):3–20, 1990.

[27] D. Edwards. *Introduction to Graphical Modelling*. Springer-Verlag, New York, 1995.

[28] D. Edwards, G. C. G. de Abreu, and R. Labouriau. Selecting high-dimensional mixed graphical models using minimal AIC or BIC forests. *BMC Bioinformatics*, 11(18), 2010.

[29] G. Elidan, I. Nachman, and N. Friedman. “Ideal Parent ” structure learning for continuous variable Bayesian networks. *Journal of Machine Learning Research*, 8:1799–1833, August 2007.

[30] M. Evans and T. Swartz. *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford Statistical Science Series. Oxford University Press, 2000.

[31] N. Friedman and D. Koller. Being Bayesian about network structure. A Bayesian approch to structure discovery in Bayesian networks. *Machine Learning*, 50:95–125, 2003.

[32] N. Friedman, M. Linial, I. Nachman, and D. Pe’er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

[33] N. Friedman, I. Nachman, and D. Pe’er. Learning Bayesian network structure from massive datasets: The “sparse candidate” algorithm. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI ’99)*, pages 196–205, 1999.

[34] D. Geiger and D. Heckerman. Learning Gaussian networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 1994.

[35] D. Geiger and D. Heckerman. Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. *The Annals of Statistics*, 30(5):1412–1440, October 2002.

[36] A. Gelman. Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–533, 2006.

[37] J. A. Gubner. Gaussian quadrature and the eigenvalue problem. Technical report, University of Wisconsin, 2007.

[38] C. Hans, A. Dobra, and M. West. Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, 102(478):507–516, June 2007.

[39] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[40] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–401, November 1999.

[41] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.

[42] T. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292:929–934, May 2001.

[43] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[44] B. Jones, C. Carvalho, A. Dobra, C. Hans, C. Carter, and M. West. Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science*, 20(4):388–400, 2005.

[45] M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8:613–636, March 2007.

[46] H. Kishino and P. J. Waddell. Correspondence analysis of genes and tissue types and finding genetic links from microarray data. *Genome Informatics*, 11:83–95, 2000.

[47] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[48] M. Kovisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.

[49] N. M. Laird and J. H. Ware. Random-effects models for longitudinal data. *Biometrics*, 38(4):963–974, December 1982.

[50] S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 2004.

[51] S.L. Lauritzen and N. Wermuth. Graphical models for association between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(1):31–57, 1989.

[52] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88:365–411, 2004.

[53] A. Lenkoski and A. Dobra. Bayesian structural learning and estimation in Gaussian graphical models. Technical Report 545, Department of Statistics, University of Washington, 2008.

[54] D. Madigan and A. E. Raftery. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

[55] P. M. Magwene and J. Kim. Estimating genomic coexpression networks using first-order conditional independence. *Genome Biology*, 5(12):R100, 2004.

[56] F. Markowetz and R. Spang. Inferring cellular networks - a review. *BMC Bioinformatics*, 8(Suppl 6):S5, September 2007.

[57] N. Meinshausen and P. Bühlmann. High-dimensional graphs and and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[58] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society, Series B*, 72(4):1–32, 2010.

[59] A. Moore and W. Wong. Optimal reinsertion: A new search operator for accelerated and more accurate Bayesian network structure learning. In *Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

[60] S. Mukherjee and T. P. Speed. Network inference using informative priors. *Proceedings of the National Academy of Sciences of the United States of America*, 105(38):14313–14318, 2008.

[61] D. V. Nguyen, A. B. Arpat, N. Wang, and R. J. Carroll. DNA microarray experiments: Biological and technological aspects. *Biometrics*, 58(4):701–717, 2002.

[62] H. D. Patterson and R. Thompson. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554, December 1971.

[63] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Francisco, 1988.

[64] R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

[65] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++.* Cambridge University Press, 2 edition, 2002.

[66] A. E. Raftery, D. Madigan, and J. A. Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.

[67] R. W. Robinson. Counting labeled acyclic digraphs. In F. Harary, editor, *New Directions in the Theory of Graphs*, pages 239–273. Academic Press, 1973.

[68] S. P. Robinson and C. Davies. Molecular biology of grape berry ripening. *Australian Journal of Grape and Wine Research*, 6:175–188, 2000.

[69] A. Roverato and G. Consonni. Compatible prior distributions for directed acyclic graph models. *Journal of the Royal Statistical Society, Series B*, 66(1):47–61, 2004.

[70] J. Schäfer and P. Bühlmann. Modeling inhomogeneous high-dimensional data-sets: with applications in learning large-scale gene correlations. In *S. Co. 2007*, 2007.

[71] J. Schäfer and K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

[72] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):Article 32, 2005.

[73] T. Silander and P. Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI '06)*, 2006.

[74] A. P. Singh and A. Moore. Finding optimal Bayesian networks by dynamic programming. Technical report, Carnegie Mellon University, 2005.

[75] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* Springer-Verlag, 1993.

[76] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2007.

[77] S. L. Teng, X. J. Zhou, and H. Huang. A statistical framework to infer functional gene associations from multiple biologically interrelated microarray experiments. To appear in *Journal of the American Statistical Association.*

[78] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, pages 584–590, July 2005.

[79] R. A. Thisted. *Elements of Statistical Computing.* Chapman & Hall, New York ; London, 1988.

[80] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[81] H. Toh and K. Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, 18(2):287–297, 2002.

[82] H. Toh and K. Horimoto. System for automatically inferring a genetic profile from expression profiles. *Journal of Biological Physics*, 28(3):449–464, 2002.

[83] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, March 2006.

[84] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002.

[85] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227, 1990.

[86] W. Wang, B. Vinocur, O. Shoseyov, and A. Altman. Role of plant heat-shock proteins and molecular chaperones in the abiotic stress response. *Trends in Plant Science*, 9(5):244–252, May 2004.

[87] E. R. Waters, G. J. Lee, and E. Vierling. Evolution, structure and function of the small heat shock proteins in plants. *Journal of Experimental Biology*, 47(296):325–338, March 1996.

[88] N. Wermuth. Linear recursive equations, covariance selection, and path analysis. *Journal of the American Statistical Association*, 75(372):963–972, December 1980.

[89] J. Whittaker. *Graphical models in applied mathematical multivariate statistics*. John Wiley & Sons, New York, 1990.

[90] A. Wille and P. Bühlmann. Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, 5(1):article 1, 2006.

[91] F. Wong, C. K. Carter, and R. Kohn. Efficient estimation of covariance selection models. *Biometrika*, 90(4):809–830, 2003.