Penalised graphical models for combined mRNA and miRNA data

Master Thesis
Mathematics: Statistical Science for Life and Behavioural Sciences
Leiden University
August 2013

Sjoerd Huisman
s0515299

Institute: VUmc, Epidemiology and Biostatistics department
First supervisor: Dr. W.N. van Wieringen
Second Supervisor: Prof. dr. A.W. van der Vaart
Third Supervisor: Dr. C.F.W. Peeters
Penalised graphical models for combined mRNA and miRNA data

Sjoerd Huisman

Introduction

Genetics is a rapidly changing field. The central dogma of molecular biology (Crick, 1970) says that deoxyribonucleic acid (DNA) is transcribed into ribonucleic acid (RNA), which is then translated into proteins. In this context RNA refers to messenger RNA (mRNA), also called coding RNA. The first types of non-coding RNA that were described are tRNA and rRNA, both needed for protein synthesis. Over the past decade a number of new non-coding RNAs have been discovered, which play a role in the regulation of gene expression (Farazi et al., 2008). One of these types is microRNA (miRNA), a type of short molecule that consist of around 23 base pairs of single stranded RNA. Part of its nucleotide sequence is complementary to that of specific mRNAs, to which it can therefore bind. This binding results in translational repression and often decay of the target mRNA (Djurancovic et al., 2011). In this way miRNAs regulate gene expression in eukaryotic cells.

Gene expression is now commonly studied with mRNA microarrays, chips which are used to measure mRNA concentrations in tissues. The goal is often to gain understanding in cellular processes or to find differences between types of tissues, for instance between healthy human tissue and tumour cells (Liang and Pardee, 2003). The use of an mRNA microarray allows for analysis of thousands of mRNAs simultaneously. This can now also be done for miRNAs, which have also been studied in relation to cancer (Farazi et al., 2011). Since an mRNA can regulate the expression of another mRNA through its protein products, the study of miRNAs often involves genetic network analysis (Choi et al., 2005). We propose here to combine mRNA data with miRNA data to find biologically relevant networks.

Combining these types of data can be done in a number of ways. This thesis explores some of these approaches, in which the focus lies on finding the topology of a part of the network: the dependencies among mRNAs and the effects of miRNAs on mRNAs. To find the topology of a genetic network, gene expression data can be analysed in several types of graphical models — see Scutari and Strimmer (2011) for an introduction. First, in section 1, we will give a general explanation of graphical models and an introduction to the commonly used GGM in an ordinary, low-dimensional, setting. Section 2 then deals with assumptions and restrictions of the proposed models. Having microarray data complicates matters, because in these datasets the number of measured
variables often exceeds the number of samples. Therefore we consider three regularised models in this thesis: a Gaussian Graphical Model (GGM) in section 3; a Maximum Likelihood Simultaneous Equation Model (ML-SEM) with only observed variables in section 4; and an Ordinary Least Squares (OLS) regression approximate approach in section 5. Finally the proposed analyses are compared both on simulated data (in section 6) and on data of a well-known pathway (in section 7).

1 Graphical models

A graph consists of nodes, which are connected by edges. In a graphical model the nodes represent variables and the edges a dependence structure (Scutari and Strimmer, 2011). The interpretation of the edges will be discussed in this section and it depends on the type of graphical model. There are two main types of graphical models. The first type has directed edges so it is a directed graph, or a Bayesian network (Wasserman, 2004). Note that the term Bayesian does not refer to Bayesian inference in this case. A special case of the directed graph is the directed acyclic graph (DAG). The second main type of graphical model is represented by an undirected graph and is sometimes called a Markov network (Kindermann and Snell, 1980). A special case of an undirected graphical model is the Gaussian graphical model, or concentration graph.

Both main types of models depict a conditional dependence structure. The way in which conditional independencies are represented is different in directed and undirected graphs. This is related to their Markov properties. The pairwise Markov property states that two nodes are independent given all other nodes, so $Y_1 \perp \perp Y_2 | Y\setminus \{Y_1, Y_2\}$ and the global Markov property implies that a pair of nodes is independent given a separating set of nodes $Y_S$, so $Y_1 \perp \perp Y_2 | Y_S$ (Wasserman, 2004). (Random variables are written upper-case italic in this thesis.) In an undirected model the absence of an edge between two nodes represents a conditional independence both in the pairwise and global sense (Wasserman, 2004). Nodes without an edge between them are separated: once we know the state of all other nodes (pairwise), or a separating set of nodes (global), the probability distribution associated with either of the two does not affect that of the other. The Markov properties in a directed graph are slightly more complicated. To understand this we will now have a look at a different type of separation, called d-separation, which also implies pairwise independence.

Figure 1A shows a directed acyclic graph. An acyclic graph is a graph without cycles or self-feedback and it can represent a recursive model (Spirtes et al., 2004). A collider is a node on a path where both edges in the path are incoming (Richardson, 1994). On the path between $Y_1$ and $Y_2$ the node $Y_3$ is a collider, because both edges of the path point towards $Y_3$. The path between $Y_1$ and $Y_5$ contains no colliders; there is a directed path from $Y_1$ to $Y_5$. In this case we say that $Y_1$ and $Y_5$ are marginally $d$-connected (Spirtes, 1993). However, conditioned on $Y_3$ they are $d$-separated. For $Y_1$ and $Y_2$ this is the other way round: there is no directed path from $Y_1$ to $Y_2$ or from $Y_2$ to $Y_1$ so marginally they are $d$-separated. This independence is broken when we condition on the collider $Y_3$. 

3
Intuitively this can be seen with a simple example, analogous to examples given by Wasserman (2004). Imagine we are looking at plants in a lab environment, for a very simple experiment. Let $Y_1$ denote whether a plant gets water and $Y_2$ whether it gets light. In our laboratory these events are randomised, so they are independent. Whether the plant has received water does not tell us anything about whether it has received light. However, if we know it has received water and we observe that the plant has died (given by $Y_3$), this increases our belief that it has been deprived of light.

As stated before, the absence of edges in an undirected graph translates directly to conditional independence. Knowing about d-separation allows us to construct from a directed graph an undirected graph adequately representing its Markov properties, by the process of moralisation (Scutari and Strimmer, 2011). To moralise a directed graph, one first connects each pair of nodes with a common (d-separating) collider and then one makes all edges undirected (Wermuth and Cox, 1998). As an example Figure 1B shows the Markov equivalent undirected graph of the directed graph in Figure 1A. This method of moralisation also works for graphical models in which cycles do occur (Richardson, 1994).

![Figure 1: (A.) A directed acyclic graph. (B.) Its induced moral graph.](image)

### 1.1 Gaussian graphical model

The Gaussian graphical model (GGM) is an undirected graph with attached probability statements. The GGM has the assumption that the data are multivariate normally distributed. In this case there is a direct relationship between conditional dependence and partial correlations (Cox and Wermuth, 1993). Partial correlations can be calculated from the elements of the inverse of the covariance matrix (Wermuth, 1976). Consider a multivariate normally distributed
random variable $Z \in \mathbb{R}^m$ and denote its covariance matrix $\Sigma$. Define the inverse of this covariance, also called the concentration or precision matrix, as $\Sigma^{-1} = \Omega$, with elements $\omega_{jk}$. Now we can calculate the partial correlation between variables $j$ and $k$, denoted by $\rho_{jk}$, as:

$$\rho_{jk} = -\frac{\omega_{jk}}{\sqrt{\omega_{jj}\omega_{kk}}} \quad \forall j \neq k.$$ (1)

A zero partial correlation corresponds to a zero in the precision matrix. This is a theoretical result; it holds for the true precision matrix. When we are confronted with data we can obtain an estimate of this precision matrix by maximum likelihood (ML) estimation. Appendix A shows that for the $m$-variate realizations of $Z$, the observations $Z = [z_1, \ldots, z_n]^T$, this estimate is

$$\Omega_{ML} = \left(\frac{1}{n} \sum_{i=1}^{n} z_i z_i^T\right)^{-1}.$$  

The partial correlations can therefore be estimated from the sample precision matrix with equation (1).

There is an equivalent way to find the partial correlations, which will later bridge the conceptual gap between GGM and the proposed methods of this thesis. Estimate the coefficient matrix $\Delta = [\delta_{jk}] \in \mathbb{R}^{m \times m}$ in equation $z_i = \Delta z_i + \eta_i$ (2) by ordinary least squares, with the assumptions that $\eta_i \sim \mathcal{N}(0, \Pi)$ and the diagonal of $\Delta$ is zero (no self-feedback). Partial correlations between any set of two variables can now be calculated from the regression coefficients as

$$\rho_{jk} = \text{sign}(\delta_{jk})\sqrt{\delta_{jk}\delta_{kj}},$$ (3)

where the $\text{sign}$ function is defined as

$$\text{sign}(x) = \begin{cases} 
-1 & \text{if } x < 0 \\
0 & \text{if } x = 0 \\
1 & \text{if } x > 0.
\end{cases}$$

Equations (1) and (3) are equivalent in the sense that they give the same estimate of the partial correlation (Krämer et al., 2009; Whittaker, 1990). This idea is used in several papers (Krämer et al., 2009; Friedman et al., 2008; Banerjee et al., 2008; Peng et al., 2009) to extend the GGM to a high dimensional setting by penalised regression. Finding zeroes in the precision matrix now corresponds to finding zeroes in the regression coefficient matrix.

The next section serves as an introduction to the three approaches that are central in this thesis: penalised versions of the GGM (section 3), the ML-SEM approach (section 4) and the OLS approach (section 5).
1.2 Penalised models

Regardless of the fitting method, in a (near) high-dimensional setting some form of penalisation is needed, since the covariance matrix of the expressions becomes singular and a poor approximation of the true covariance (Schäfer and Strimmer, 2005). Finding a penalised graphical model can be done in a number of ways.

One can extend the GGM, to find an undirected graph directly by calculating partial correlations from a penalised estimate of the covariance structure. This has been done with a lasso penalty in the Graphical lasso (Glasso) method by Friedman et al. (2008), with a ridge penalty by Stifanelli et al. (2012), and with alternative shrunken covariance estimators by Schäfer and Strimmer (2005). Alternatively one can calculate partial correlations from penalised regression equations, making use of the general idea of equation (3). This is done equation by equation with lasso regression by Meinshausen and Bühlmann (2006). Peng et al. (2009) have a more elaborate approach in which a lasso penalty is placed directly on the partial correlations. We choose here to fit GGMs with lasso and ridge estimates of the covariance structure.

In our biological context we do this by combining mRNA and miRNA expression data (in \( Z \)) and fitting a penalised graphical model using the partial correlations obtained with equation (1) or (3). However, this simple GGM approach cannot incorporate prior information about the presence of edges. For our type of data we can hypothesise a causal restriction: miRNA expression causes changes in mRNA expression, but not the other way round. We can incorporate this idea in the construction of a graphical model. This thesis discusses two types of methods in which this is possible: Maximum Likelihood Simultaneous Equation Modelling (ML-SEM) and multiple Ordinary Least Squares (OLS) regression.

For both types it is convenient to reformulate equation (2) and split it up in two parts. Let \( y_i \) be a vector of centred mRNA expressions and \( x_i \) be a vector of centred miRNA expressions. The vectors \( \zeta_i \) and \( \epsilon_i \) are error vectors, with mean zero and \( \text{cov}(\zeta_i) = \Psi \), \( \text{cov}(\epsilon_i) = \Phi \) and \( \text{cov}(\zeta_i, \epsilon_i) = \Xi \).

Now define a model

\[
\begin{align*}
y_i &= B y_i + \Gamma x_i + \zeta_i \tag{4a} \\
x_i &= \Delta y_i + \Lambda x_i + \epsilon_i. \tag{4b}
\end{align*}
\]

This reformulated model allows us to incorporate the prior information we are interested in. If mRNA expression is changed by, but does not change miRNA expression, this means that \( \Delta = 0 \). Now if we set out to find the edges among mRNAs and from miRNAs to mRNAs only, we can leave out equation (4b). We are then left with

\[
y_i = B y_i + \Gamma x_i + \zeta_i, \tag{5}
\]

and can consider the miRNA expressions exogenous in a simultaneous equation model, with a covariance matrix defined as \( \text{cov}(x_i) = \Theta \). This idea is used in
the ML-SEM and OLS approaches of this thesis, as will be explained in sections 4 and 5.

To summarise: The GGM extensions that are implemented in this thesis are the Glasso (Friedman et al., 2008) and a ridge version of the GGM similar to that of (Stifanelli et al., 2012). These extensions of the GGM are explained in section 3. The ML-SEM method (lasso and ridge) of this thesis (explained in section 4) is a two-stage addition to these approaches in which we can make use of prior information about directionality. The OLS regression method (lasso and ridge) is in turn a limited information approximation related to the regression approaches, as is explained in section 5.

2 Model assumptions and restrictions

The three methods discussed in this thesis (GGM, ML-SEM, OLS) have several assumptions in common. The centred mRNA and miRNA expressions are both assumed to follow a normal distribution, with covariance matrices $\Sigma_{yy}$ and $\Sigma_{xx}$, a covariance matrix between $y_i$ and $x_i$ denoted by $\Sigma_{yx}$ and its transpose by $\Sigma_{xy}$:

$$\begin{bmatrix} y_i \\ x_i \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0_p \\ 0_q \end{bmatrix}, \begin{bmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{bmatrix}\right)$$

For the GGM this is the only assumption that is needed, since under this assumption conditional dependencies are partial correlations, which can be calculated from an estimate of the precision by equation (1). For the ML-SEM and OLS approaches we estimate parameters of equation (5) under the assumptions that $\zeta_i \sim \mathcal{N}(0, \Psi)$ and $\zeta_i \perp \perp x_i$.

The ML-SEM approach makes use of the model-implied covariance (Bollen, 1989), which is explained in section 4. The regression coefficients of equation (5) are estimated by maximising the likelihood of the model-implied covariance matrix $\Sigma$ with respect to some estimate of the covariance $S$, over all free parameters in the model. To find unique estimates of the model parameters, restrictions are needed. This can be shown by first rewriting equation (5) to

$$y_i = By_i + \Gamma x_i + \zeta_i$$

$$(I - B)y_i - \Gamma x_i = \zeta_i$$

$$[(I - B) - \Gamma'] \begin{bmatrix} y_i \\ x_i \end{bmatrix} = \zeta_i,$$

and then multiplying both sides of this equation with their transposes and substituting the model-implied covariance $\Sigma$ and the error covariance matrix $\Psi$, to get:
Define: \( C = [(I - B) - \Gamma] \).

Now define a non-singular matrix \( A \in \mathbb{R}^{(p+q)(p+q)} \) and see that

\[
\Psi = C\Sigma C^T = CAA^{-1}\Sigma(AA^{-1})^TC^T = (CA)[A^{-1}\Sigma(A^{-1})^T](CA)^T = C^*\Sigma^*(C^*)^T,
\]

which means there is an infinite number of matrices \( C^* = CA \) and \( \Sigma^* = A^{-1}\Sigma(A^{-1})^T \) that give the same \( \Psi \). Therefore restrictions on at least part of the coefficients are needed.

A common way to do this is by fitting a recursive model with a diagonal \( \Psi \), which is always identified (Bollen, 1989). Under the recursive model restrictions the diagonal of \( B \) is zero, which excludes self-feedback, and the rows and columns of \( B \) can be permuted to make it a triangular matrix. In a regression context a triangular coefficient matrix is equivalent to forward selection: the second variable in \( y \) is regressed against the first, the third against the first two \( y \)s and so on. This translates to a directed acyclic graph between the variables. It is important to note that the ordering of the variables in \( y \) influences the coefficient estimates. In our case \( y_i \) contains mRNA expressions in an arbitrary order. Therefore this set of restrictions is not used here.

We propose an alternative set of restrictions, in which \( \Psi \) is still diagonal and the diagonal of \( B \) is zero, but both triangles of \( B \) are equal: \( \beta_{jk} = \beta_{kj} \). In a directed graph this can be represented by directed edges in both directions, if the coefficients are non-zero. Note that this model is non-recursive, but it is still directed. The identifiability under this set of restrictions is not proven here, but it was observed in simulations. For the ML-SEM model we further assume that the errors \( \zeta_i \) in equation (5) are independent of the exogenous miRNA variables, so \( \zeta_i \perp x_i \), as noted before.

The OLS approach needs only one restriction for identification, which is that self-feedback is not allowed: the diagonal of \( B \) is restricted to be zero. With this restriction each mRNA can be regressed against all other mRNAs and the miRNAs to obtain OLS estimates. However, for comparison to the ML-SEM model we impose the same symmetry restriction on the structure of \( B \). Because unpenalised OLS regression is a projection, we cannot impose restrictions on the error covariance matrix, but we can choose to assume independence of the errors (as implied by a diagonal \( \Psi \)).
3 GGM

Penalised estimation of Gaussian graphical models is a widely discussed topic, especially for \( L_1 \)-type penalties (Banerjee and Ghaoui, 2006; Yuan and Lin, 2007; D’Aspremont et al., 2008; Rothman et al., 2008). These methods are aimed at likelihood maximisation (see equation (14) in appendix A) with a penalty on the precision matrix:

\[
\max_{\Omega} [\log(|\Omega|) - \text{tr}(S\Omega)] + \frac{1}{2} \lambda \|\Omega\|_1.
\]

The \( L_1 \)-norm \( \|\Omega\|_1 \) is the sum of the absolute values of the precision matrix elements. To estimate the penalised precision we use the R package glasso by Friedman et al. (2008), which implements a coordinate descent procedure. With increasing values of the penalisation parameter \( \lambda \), elements of the precision matrix get set to zero. It is this built-in variable selection that makes lasso methods popular.

An alternative penalisation is the \( L_2 \)-norm, or ridge method. This has been used for GGMs by Stifanelli et al. (2012) and it involves maximisation of

\[
\max_{\Omega} [\log(|\Omega|) - \text{tr}(S\Omega)] + \frac{1}{2} \lambda \|\Omega\|_2^2.
\]

The \( L_2 \)-norm (or squared Frobenius norm) \( \|\Omega\|_2^2 \) is the sum of all squared elements of the precision. This equation has an analytical solution, here explained as given by Van Wieringen and Peeters (submitted). We can rewrite the previous equation as

\[
\max_{\Omega} [\log(|\Omega|) - \text{tr}(S\Omega)] + \frac{1}{2} \lambda \text{tr}(\Omega\Omega^T).
\]

Take its derivative and set that to zero to obtain

\[
\Omega^{-1} - S - \lambda \Omega = 0_{m \times m}.
\]

If we add \( \lambda \Omega \) to both sides and post-multiply both sides by \( \Omega^{-1} \) we get

\[
\Omega^{-2} - S\Omega^{-1} = \lambda I_{m \times m},
\]

but if we pre-multiply both sides by \( \Omega^{-1} \) we get

\[
\Omega^{-2} - \Omega^{-1}S = \lambda I_{m \times m}.
\]

These statements are equivalent and we can add these two alternatives up and divide by two to obtain:

\[
\Omega^{-2} - \frac{1}{2} \Omega^{-1}S - \frac{1}{2} S\Omega^{-1} = \lambda I_{m \times m}.
\]
This is equivalent to

\[
(\Omega^{-1} - \frac{1}{2} S)^2 - \frac{1}{4} S^2 = \lambda I_{m \times m}.
\]

Now solve this for \( \Omega \) by first rewriting to

\[
(\Omega^{-1} - \frac{1}{2} S)^2 = \frac{1}{4} S^2 + \lambda I_{m \times m},
\]

and then taking the matrix square root (Harville, 1997) to get

\[
\Omega^{-1} - \frac{1}{2} S = \left( \frac{1}{4} S^2 + \lambda I_{m \times m} \right)^{1/2}.
\]

Add \( \frac{1}{2} S \) to both sides and finally take the inverse to end up with a ridge estimate for the precision:

\[
\hat{\Omega}_{ridge} = \left[ \left( \frac{1}{4} S^2 + \lambda I_{m \times m} \right)^{1/2} + \frac{1}{2} S \right]^{-1}.
\]

Both the lasso and the ridge estimate of the precision can be plugged into equation (1) to find estimates of the partial correlations. In the lasso approach more and more estimates go to zero when the penalty parameter increases (Friedman et al., 2008). We decide an edge is found to be present when its estimate is non-zero. In the ridge approach the estimates shrink towards zero, but they do not reach zero for finite values of the penalty parameter.

### 3.1 Cut-off for ridge

The lasso approach has edge selection built-in, but ridge estimators do not set coefficients exactly to zero and therefore a cut-off has to be specified to determine whether an edge is present. One approach to do this is by significance testing of the partial correlations. For unpenalised estimates of partial correlations in low-dimensional datasets significance testing is well-developed. Let \( \rho_{jk} \) be a correlation or partial correlation estimate of two independent normally distributed variables, so the true (partial) correlation is zero. Then after Fisher transformation

\[
\hat{z}_{jk} = \frac{1}{2} \log \frac{1 + \rho_{jk}}{1 - \rho_{jk}},
\]

we have a close to normally distributed variable \( \hat{z}_{jk} \), the standard deviation of which can be estimated as

\[
\hat{sd}(\hat{z}_{jk}) = \frac{1}{\sqrt{df - 3/2}}.
\]
in which the number of degrees of freedom $df$ is the sample size minus the number of considered variables (Fisher, 1921). For partial correlations $df = n - (p + q - 2)$, since we condition on $p + q - 2$ variables for each partial correlation (Fisher, 1924). This shows that in a high dimensional setting the number of degrees of freedom becomes negative and Fisher’s approximation can not be used. Moreover, the degrees of freedom for a penalised estimate of a partial correlation is not defined.

To avoid having to estimate the distribution of partial correlation coefficients, we take the naive approach of setting a cut-off value for the partial correlations at 0.1. This is arbitrary and far from ideal, but solving this problem is outside the scope of this thesis. Testing procedures for partial correlation coefficients have been undertaken by Schäfer and Strimmer (2005), but implementation of their multiple testing corrected method (Strimmer, 2008) combined with our penalisation schemes did not yield satisfactory results in the simulations of section 6.

4 ML-SEM

In the ML-SEM approach we extend the penalised GGM model to include our prior knowledge about the relationship between miRNAs and mRNAs. The algorithm starts with fitting a penalised GGM. With lasso penalisation this gives a graph topology in which only part of the variables are connected. With ridge penalisation none of the edges are set to zero, so a cut-off is taken for the partial correlations to trim away edges (see section 3.1).

After trimming away edges the precision is re-estimated by maximum likelihood estimation, with equal-to-zero restrictions on the elements that represent absent edges in the undirected graph. This is also done with the glasso package (Friedman et al., 2008). This new estimate of the precision matrix also implies a new estimate of the covariance matrix (its inverse). This penalised covariance matrix estimate is then used to fit a structural equation model. According to Pearl (2000) only the partial correlations between variables that are d-separated are zero, independent of the edge parameters and error variances. Thus we go from a penalised estimated of an undirected graph, as in the extended GGM, to a directed graph that is perfectly represented by this undirected graph.

Under the restrictions of section 2, the ML-SEM model is saturated: the maximum likelihood solution is unique and the model-implied covariance is equal to the covariance matrix on which the model is maximised. Therefore we can "plug in" an estimate of a precision matrix corresponding to an undirected graph to end up with estimates for a directed graph that is Markov equivalent to this undirected graph.

So once we have obtained the estimate of the covariance corresponding to the undirected graph, we use this estimate to find the maximum likelihood values for $B$ and $\Gamma$ in equation (5). The non-zero elements of $B$ and $\Gamma$ represent the final directed graph. Figure 2 illustrates this method. For estimation of the parameters in the SEM we need the model-implied covariance, which is derived in the next section.
4.1 Implied covariance and precision

Simultaneous equation models can be fitted by maximum likelihood estimation. This is done by maximizing the model-implied covariance with respect to a covariance matrix estimate, over the free parameter values (Bollen, 1989). To find the model-implied covariance matrix

\[ \Sigma = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{bmatrix}, \]

we look at each block of this matrix in turn, using covariance algebra as in Bollen (1989). For \( \Sigma_{yy} \) rewrite equation (5) to

\[
\begin{align*}
y_i &= \text{By}_i + \Gamma x_i + \zeta_i \\
(I - B)y_i &= \Gamma x_i + \zeta_i \\
y_i &= (I - B)^{-1}(\Gamma x_i + \zeta_i).
\end{align*}
\]

Now post-multiply both sides by their transposes and substitute the covariances parameters (keeping in mind that \( \zeta_i \perp \perp x_i \)):

\[
\begin{align*}
y_i y_i^T &= (I - B)^{-1}(\Gamma x_i + \zeta_i)(\Gamma x_i + \zeta_i)^T(I - B)^{-1} \Gamma x_i \zeta_i^T \\
\Sigma_{yy} &= (I - B)^{-1}(\Gamma x_i x_i^T \Gamma^T + \zeta_i \zeta_i^T)(I - B)^{-1} \\
\Sigma_{yy} &= (I - B)^{-1}(\Gamma \Theta \Gamma^T + \Psi)(I - B)^{-1}.
\end{align*}
\]

To find the implied \( \Sigma_{yx} \) again rewrite equation (5), but now post-multiply both sides with \( x_i^T \) and substitute to obtain

\[
\begin{align*}
y_i x_i^T &= (I - B)^{-1}(\Gamma x_i + \zeta_i)x_i^T \\
y_i x_i^T &= (I - B)^{-1}(\Gamma x_i x_i^T) \\
\Sigma_{yx} &= (I - B)^{-1}(\Gamma \Theta).
\end{align*}
\]

The miRNAs are exogenous, so their covariance is just given by \( \Theta \). If we further realize that \( \Sigma_{xy} = \Sigma_{yx}^T \), we have completed the implied covariance:

\[ \Sigma = \begin{bmatrix} (I - B)^{-1}(\Gamma \Theta \Gamma^T + \Psi)(I - B)^{-1} & (I - B)^{-1}(\Gamma \Theta) \\ (I - B)^{-1}(\Gamma \Theta \Gamma^T + \Psi)(I - B)^{-1} & \Theta \end{bmatrix}. \quad (7) \]

Using the Schur complement (Harville, 1997), we can find the inverse of this block matrix, which is the model-implied precision \( \Omega \):

\[ \Sigma^{-1} = \Omega = \begin{bmatrix} \Omega_{yy} & \Omega_{yx} \\ \Omega_{xy} & \Omega_{xx} \end{bmatrix}. \]
The Schur complement $Q$ of $\Sigma_{yy}$ is:

$$Q = \Omega_{yy} = (\Sigma_{yy} - \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy}) = (\Sigma_{yy} - (I - B)^{-1} \Theta \Theta^{-1} \Theta \Gamma^T (I - B)^{-1} (I - B)^{-1} \Gamma^{-1}) = \{(I - B)^{-1} \Psi [(I - B)^{-1} (I - B)^{-1}]^{-1} = (I - B)^T \Psi^{-1} (I - B).$$

And we can find the other elements of the implied precision as:

$$\Omega_{yx} = -Q \Sigma_{yx} \Sigma_{xx}^{-1} = -(I - B)^T \Psi^{-1} (I - B) (I - B)^{-1} \Gamma \Theta^{-1} = -(I - B)^T \Psi^{-1} \Gamma,$$

$$\Omega_{xy} = \Omega_{yx}^T = -\Gamma^T \Psi^{-1} (I - B)$$

and

$$\Omega_{xx} = \Sigma_{xx}^{-1} + \Sigma_{xx}^{-1} \Sigma_{xy} Q \Sigma_{yx} \Sigma_{xx}^{-1} = \Theta^{-1} + \Theta^{-1} \Theta \Gamma^T [(I - B)^{-1}]^T (I - B)^T \Psi^{-1} (I - B) (I - B)^{-1} \Gamma \Theta^{-1} = \Theta^{-1} + \Gamma^T \Psi^{-1} \Gamma,$$

to complete the expression for the implied precision:

$$\Omega = \begin{bmatrix} (I - B)^T \Psi^{-1} (I - B) & -(I - B)^T \Psi^{-1} \Gamma \\ -\Gamma^T \Psi^{-1} (I - B) & \Theta^{-1} + \Gamma^T \Psi^{-1} \Gamma \end{bmatrix}.$$ (8)

### 4.2 Estimation

Now we have an expression for the model-implied covariance and precision as a function of the model parameters. In the low-dimensional case we would find the ML estimates of these parameters by searching for

$$\max_{\Sigma} \left[ -\log(|\Sigma|) - \text{tr}(S \Sigma^{-1}) \right],$$ (9)

under the restrictions discussed in section 2. The derivation of this log-likelihood function for the low-dimensional case can be found in appendix A.

As explained at the start of this section, we plug in our penalised estimate of $S$ (which corresponds to an undirected graph) in equation (9) to find estimates under the restrictions of section 2. Fitting this SEM is implemented with the R package sem (Fox, 2006). The estimates of $B$ and $\Gamma$ describe the final estimated
directed graph. Elements of $B$ and $\Gamma$ will be zero when edges are absent in the undirected model of the first step, but the remaining edge estimates do not go to zero numerically. Without any cut-off we would just recover all the edges of the undirected network. Therefore we again take an arbitrary cut-off of 0.1. Edges with an estimated magnitude greater than this value are found to be present.

![Figure 2: A small example of a graphical model for mRNAs $Y_1$, $Y_2$, and $Y_3$, and miRNAs $X_1$ and $X_2$. Part (A.) shows an undirected graph and part (B.) a directed graph with the same Markov properties. There are multiple (B.)-type graphs with the same (A.)-type graph, but there is only one way to construct an (A.)-type graph from a (B.)-type graph. In the ML-SEM approach we estimate the (A.)-type first and then find the optimal (B.)-type; in the OLS approach we compute an approximation of the parameters of the (B.)-type and then find its implied (A.)-type. The two edges between each pair of nodes, which are indicated by parallel arrows in (B.), are restricted to be equal.](image)

5 OLS

In the OLS regression approach we estimate coefficients among the mRNAs ($B$) and between the miRNAs and the mRNAs ($\Gamma$) and then approximate partial correlations from these coefficients. This is analogous to the Meinshausen and Bühlmann (2006) method, but with covariates (miRNAs) added to the regressions. Furthermore we will impose restrictions on the parameters between the mRNAs (in $B$), as discussed in section 2. Note that this is a limited information
approach, in which the covariances of the miRNAs are not modelled and the parameters we estimate are in that sense not directed. Still we obtain estimates for $B$ and $\Gamma$ which we can insert in the implied precision equation (8) of section 4 as if we are moralising a directed graph. In that case we also need estimates for the error covariance $\Psi$ and the covariance matrix of the miRNAs $\Theta$. In this way we approximate a precision matrix from which we can calculate estimated partial correlations. This an approximate method which has the advantages that it is relatively fast for medium sized datasets and penalisations can be built in easily.

We start by finding coefficient estimates for equation (5) by ordinary least squares regression. We try to impose the same symmetry restriction on the coefficient matrix $B$ as in the ML-SEM approach and achieve this with some data manipulation.

We first go to a slightly different notation: we have used column vectors for samples $y_i \in \mathbb{R}^p$ and $x_i \in \mathbb{R}^q$, but now we use column vectors for RNAs $y^*_j \in \mathbb{R}^n$ and $x^*_l \in \mathbb{R}^n$. Let $Y = (y^*_1, y^*_2, ..., y^*_p)$ with $y^*_j = (y_{1j}, y_{2j}, ..., y_{nj})^T$ denote the matrix of observed and centred mRNA expressions and $X = (x^*_1, x^*_2, ..., x^*_q)$ with $x^*_l = (x_{1l}, x_{2l}, ..., x_{nl})^T$ the matrix of observed and centred miRNA expressions.

Vectorise $Y$ to make $v = (y^*_1, y^*_2, y^*_3, y^*_4)^T$, a vector that contains all mRNA expressions. Construct a matrix of explanatory variables $R$ that contains all combinations of two $y$s in its columns, together with the block diagonal matrix $I_{p \times p} \otimes X$. Now find a penalised ordinary least squares fit of equation

$$v = R\alpha + \xi,$$

(10)

to obtain a regression coefficient vector $\hat{\alpha}$ and an error vector $\xi$. An example is shown below for $p = 4$ and $q = 1$:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} y_{2} & y_{3} & y_{4} & 0 & 0 & 0 & x_{1} & 0 & 0 \\ y_{1} & 0 & 0 & y_{3} & y_{4} & 0 & 0 & x_{1} & 0 \\ 0 & y_{1} & 0 & y_{2} & 0 & y_{4} & 0 & 0 & x_{1} \\ 0 & 0 & y_{1} & 0 & y_{2} & y_{3} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_{12} \\ \beta_{13} \\ \beta_{14} \\ \beta_{23} \\ \beta_{24} \\ \beta_{34} \\ \gamma_{11} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{14} \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} \otimes \mathbf{X}.$$ 

The unpenalised OLS fit

$$\hat{\alpha} = (R^T R)^{-1} R^T v$$

(11)

minimises the sum of the squared errors in predicting the $y$ variables with the restriction that the coefficients between pairs of $y$s are equal ($\beta_{jk} = \beta_{kj}$). We propose to find penalised versions of this fit.
For a lasso penalty add an $L_1$-norm of the coefficients to the squared error loss function for equation (10) to obtain

$$\hat{\alpha} = \arg \min_{\alpha} \|v - R\alpha\|_2^2 + \lambda \|\alpha\|_1.$$ 

The solution of this equation can be found with a combination of gradient ascent and the Newton-Rhapson algorithm in a method by Goeman (2010). This method is implemented in the R package **penalized**.

For ridge regression similarly add an $L_2$-norm of the coefficients to the squared error loss to obtain

$$\hat{\alpha} = \arg \min_{\alpha} \|v - R\alpha\|_2^2 + \lambda \|\alpha\|_2^2,$$

which can be maximised by

$$\hat{\alpha} = (R^T R + \lambda I)^{-1} R^T v,$$

as shown by Hoerl and Kennard (1970).

The vector $\hat{\alpha}$ contains the $p(p - 1)/2$ unique coefficients of $\hat{B}$ and the $pq$ coefficients of $\hat{\Gamma}$. Once we have found the restricted and penalised estimates $\hat{B}$ and $\hat{\Gamma}$, we can make use of equation (8) to estimate precision elements and corresponding partial correlations.

In the OLS lasso approach an estimate of the magnitudes of the partial correlations is not needed, because some regression coefficients will be set to zero and this yields an estimate of a directed graph topology. When this topology is moralised we end up with our final OLS lasso estimate of the undirected graph.

In the OLS ridge approach we set a cut-off for partial correlations as in section 3.1, so estimates of the precision and partial correlations are needed. In the limited information OLS approach $\Theta$ is not modelled and no restrictions are imposed on the error covariance $\Psi$. Also note that equation (8) relies on the assumption $\zeta_i \perp \perp x_i$. Since OLS regression is an orthogonal projection, in the unpenalised case $\zeta_i \perp \perp x_i$, by construction, but this orthogonality is distinct from the theoretical assumption $\zeta_i \perp \perp x_i$, and it does not hold for the penalised estimates.

Despite these theoretical problems, we take the diagonal matrix of error variances for $\Psi$. The sample covariance of $X$ is singular in a high dimensional setting, so it can not be inverted. In its place we insert a penalised estimate of the covariance of $X$. The expression $(R^T R + \lambda I)$ in the ridge normal equation (12) contains $(X^T X + \lambda I)$ in blocks, because $R$ contains $I_{p \times p} \otimes X$. This leads to a ridge estimate

$$\hat{\Theta} = \frac{1}{n}(X^T X + \lambda I),$$
which is more in line with the OLS ridge estimation than the ridge estimate of section 3, and uses the same penalty parameter. Now we have OLS ridge estimates for $B$, $\Gamma$, $\Psi$ and $\Theta$, we can insert these in equation (8) to obtain a precision matrix and calculate partial correlations. With a cut-off of 0.1 these partial correlations are translated into the final OLS ridge undirected graph.

The process of penalised OLS estimation of an undirected graph is also illustrated in Figure 2. In a way it is opposite to the ML-SEM approach: we first approximate a directed graph and then moralise it to an undirected graph.

6 Simulation

To test these six methods, GGM, OLS and ML-SEM with both lasso and ridge, we perform a simulation study. For time considerations the comparison will be on a medium-dimensional dataset of 50 RNA expressions with sample size 100. To draw these samples we specify coefficient matrices which are transformed to an implied covariance matrix by equation (7). This implied covariance was derived from equation (5) under the assumption $\zeta_i \perp \perp x_i$, so this holds for the sampling as well. We also specify $\zeta_i \sim \mathcal{N}(0, \Psi_{SIM})$ and construct $B_{SIM}$ matrices with zeroes on the diagonal (no self-feedback). The parameter matrices $\Psi_{SIM}$ and $\Theta_{SIM}$ are the same in each scheme and the others vary, to give the following sampling procedure:

1. Choose the number of mRNAs $p$ and miRNAs $q$:
   (a) $p = 15$ and $q = 35$.
   (b) $p = 25$ and $q = 25$.
   (c) $p = 35$ and $q = 15$.

2. Specify coefficient matrices for a directed generative model:
   - $B_{SIM}$ for edges between mRNAs (Figure 3):
     (a) $B_{band}$: banded/toeplitz structure.
     (b) $B_{block}$: block diagonal structure.
     (c) $B_{path}$: structure based on part of the Kyoto Encyclopedia of Genes and Genomes (KEGG) version map04210 of the apoptosis pathway (Kanehisa and Goto, 2000; Kanehisa et al., 2012).
   - $\Gamma_{SIM}$ for edges from miRNAs to mRNAs (Figure 3):
     (a) $Geach2$: 2 miRNAs for each mRNA.
     (b) $Grange5$: one mRNA with 5 miRNAs, one mRNA with 4 miRNAs, etc. The other mRNAs have no incoming miRNA edges.
   - $\Psi_{SIM}$ for the error covariance matrix:
     - Identity matrix $I_{p \times p}$.
   - $\Theta_{SIM}$ for the covariance matrix of miRNAs:
– Matrix with 1 on diagonal and 0.3 for all off-diagonal elements.

3. Find the implied covariance matrix with equation (7) and the implied precision with equation (8).

4. Convert the implied covariance to a correlation matrix, for equal scaling.

5. Sample multivariate normal data sets of size 100 from this correlation matrix with \texttt{rmvnorm} in \texttt{R} and centre these.

With three combinations for \( p \) and \( q \), three for \( \mathbf{B}_{SIM} \) and two for \( \mathbf{\Gamma}_{SIM} \), this gives 18 sampling schemes in total. See Figure 3 for examples of the \( \mathbf{B}_{SIM} \) and \( \mathbf{\Gamma}_{SIM} \) structures. In this type of sampling the equation (5) represents a generating directed graph, the model that induces the conditional dependencies (Wermuth and Cox, 1998).

The performance of the six models is judged with receiver operating characteristic (ROC) curves. Each model is fitted on ten samples and the found presence of edges is compared to their presence in the “simulation truth”. This is done for a range of penalty parameter values, to obtain false positive rates (FPR) and true positive rates (TPR) that vary with the penalty parameter value and are averaged over the samples to obtain ROC curves. Figures 4, 5 and 6 show the ROC curves for the 18 sampling schemes.

The GGM and OLS approaches give estimates of undirected graphs. Therefore their outcomes are compared to the edges in the moralised version of the generating graph, which is given by the non-zero entries of the simulation implied precision. The ML-SEM approach gives an estimate of a directed graph. The outcome of the ML-SEM models is therefore compared to the generating directed graph given by \( \mathbf{B}_{SIM} \) and \( \mathbf{\Gamma}_{SIM} \). The \texttt{Bband} and \texttt{Bblock} schemes are symmetric in the edges among mRNAs, just like the ML-SEM fitted graphs. For the \texttt{Bpath} schemes a pair of edges between two mRNAs in the ML-SEM graph is declared correct if a directed edge (of either direction) is present between those mRNAs according to the generating directed graph.

The curves for the lasso OLS method stand out since in most sampling schemes the TPR levels-off. Selecting more edges in this region does not increase the number of correct edges found. To look deeper into this, Figures 7 and 8 show heatmaps of the edge presence. The plots at the top show the true presence of edges according to the sampling schemes, and the other plots indicate the percentage of times each edge is detected among 20 samples with the GGM lasso method and the OLS lasso method. The fits were chosen to have an average FPR of 0.05. In Figures 7 GGM has a higher TPR and in Figure 8 OLS has a higher TPR.

The high performance of OLS lasso in some of the sampling schemes is a result of the fact that it selects very few edges between miRNAs and mRNAs. The GGM lasso method does not have a preference for selecting edges among mRNAs, but it does a poor job at finding edges between miRNAs and mRNAs. Because in our sampling schemes detecting edges between miRNAs and mRNAs seems very difficult, the fact that the OLS lasso method has a preference for detecting among-mRNA edges, lowers its FPR for the sparser fits. Especially in sampling schemes with few true miRNA-mRNA edges (\texttt{Grange5}) this benefits the OLS lasso performance.

18
In Figure 8 we can see that the last two mRNAs are both connected to clusters of other mRNAs, but not to each other in the sampling scheme. The GGM and OLS methods connect these mRNAs to each other as well. Edges in a generating directed graph induce covariances between variables that are pairwise Markov independent, as long as there is a directed (collisionless) path between them (Wermuth and Cox, 1998). Presumably the noise induced by these covariances makes detection of the absence of these edges more difficult. This would also explain the relatively high performance of all models in the block diagonal BSIM sampling schemes, because these blocks are not connected by edges in the generating directed model.

The ROC curves for the ML-SEM models are the best in most sampling schemes. These models are of a different type (directed rather than undirected), which we have to bear in mind. It does show that finding a directed graph is feasible, if the truth is in line with the model assumptions. In the Bpath scheme the edges among mRNAs are not symmetric and this results in a worse performance of the ML-SEM models. The symmetry restrictions in the ML-SEM approach were chosen to ensure identification, but the truth is most likely not symmetric in its edges among mRNAs.

In Figures 4 to 6 the ridge methods show comparable performance to the lasso methods for the banded (Bband) and block diagonal (Bblock) schemes and lower performance for the pathway (Bpath) scheme. Given that we have used an arbitrary cut-off for determining the presence of edges, these results cannot be given a fully fair comparison here. The noticeable decrease of the TPR for increasing FPR in the ML-SEM ridge in the Bblock schemes can be attributed to convergence issues of the sem algorithm.
Figure 3: Examples of the generating $B_{SIM}$ and $\Gamma_{SIM}$ matrices. These plots show the edges that were used to find the implied covariance from which was sampled in the simulation. The three options for $B_{SIM}$ and the two options for $\Gamma_{SIM}$, together with the three ratios of $p$ and $q$ give 18 sampling schemes in total.
Figure 4: ROC curves for the sampling schemes with $p = 15$ and $q = 35$, each based on 10 samples of size 100.
Figure 5: ROC curves for the sampling schemes with $p = 25$ and $q = 25$, each based on 10 samples of size 100.
Figure 6: ROC curves for the sampling schemes with $p = 35$ and $q = 15$, each based on 10 samples of size 100.
Figure 7: Example of performance. The true edges between mRNAs and miRNAs (above) and the percentage of times each edge is found among 20 samples of size 100 with the GGM lasso and OLS lasso methods, penalty parameter chosen for which $FDR = .05$. 
Figure 8: Example of performance. The true edges between mRNAs and miRNAs (above) and the percentage of times each edge is found among 20 samples of size 100 with the GGM lasso and OLS lasso methods, penalty parameter chosen for which $FDR = .05$. 
7 Case study

For the case study we downloaded mRNA and miRNA expression data for glioblastoma multiforme cancer samples, from the TCGA website (The Cancer Genome Atlas Research Network, 2008). Samples were taken from tumour expressions compared to healthy control tissue, and expressions were measured by the University of North Carolina with the Affymetrix HT Human Genome U133 Array Plate Set for mRNAs and by the Broad Institute with the Agilent 8 x 15K Human miRNA-specific microarray for miRNAs. From the mRNA expression profiles all genes were selected that are part of the apoptosis pathway map04210 according to KEGG (Kanehisa and Goto, 2000; Kanehisa et al., 2012), to yield a set of 80 mRNAs. The genes in the apoptosis pathway were checked for being validated miRNA targets, as registered by miRecords (Xiao et al., 2009), to reduce the list of 534 measured miRNAs to 30 relevant miRNAs. The conversion of the various mRNA and miRNA identifiers used in these datasets was performed with the database-to-database-conversions tool of bioDBnet (Mudunuri et al., 2009). The overlapping 419 samples between the mRNA and the miRNA datasets were used to fit penalised graphical models.

Glioblastoma multiforme is the most common type of brain cancer and the median survival time for patients is less than one year (Johnson and O’Neill, 2012). It was the first type of cancer to be studied by TCGA (The Cancer Genome Atlas Research Network, 2008). An important pathway under study for treatment targets in glioblastoma multiforme is the apoptosis pathway (Eisele and Weller, 2013). The apoptosis pathway is responsible for a type of programmed cell death. This is often disturbed in cancer tissues, in which cells fail to die because the apoptosis pathway does not function properly (Ouyang et al., 2012).

Figure 9 shows the apoptosis pathway, with directed edges as given by KEGG and with miRNAs connected to their validated targets according to miRecords. This graph can be moralised, see section 1, to the undirected graph in Figure 10. If we now fit graphical models to the glioblastoma data we have several expectations. Those gene products (mRNAs and miRNAs) that directly cause changes in the expressions of other genes could be connected to the genes they affect. The same is true for gene products that have common targets (colliders). In this respect we expect to see the structure of Figure 10 reflected in the fitted undirected graphical models. On the other hand, genes that are commonly differentially expressed together in glioblastoma independent of the other genes are pairwise Markov dependent, and should therefore also be connected in the fitted undirected graphical models. To find only the pathway topology in the fitted models we should have used expression levels in normal tissues, which has been done with graphical models Ma et al. (2007); Krämer et al. (2009).

The GGM lasso and ML-SEM lasso models were fitted to the expression data. The OLS model could not be fitted, because the R matrix of equation (10) grows strongly with the number of samples and variables. For this pathway it has \(pn = 80 \times 419 = 33,520\) rows and \(p(p−1)/2+pq = 80 \times 79/2 + 80 \times 30 = 5,560\) columns.

Figure 11 shows the GGM lasso fit with the same number of edges (549) as the undirected (moralised) version of the pathway in Figure 10. This is again an
arbitrary choice, since a testing procedure for edge presence is not developed in this thesis. This fit is an undirected graph, for which we can find the most likely directed graph with the ML-SEM procedure. This graph is shown in Figure 12, for an edge-size cut-off of 0.1. In a similar, but larger, case study Peng et al. (2009) fit a penalised graphical model to breast cancer samples. They select 1,217 genes that are associated with tumour progression, with measured mRNA expressions only. They point out that the variables with the largest numbers of edges are likely to be important in breast cancer regulation.

Along the same lines we can look at Figure 12 to find interesting mRNAs and miRNAs. There are relatively few edges from miRNAs to mRNAs, but three miRNAs stand out. The miRNA with the largest number of edges (6) is \textit{miR-21}. This miRNA has been shown to be up-regulated in glioblastoma (Chan et al., 2005), and its knock-down in mice strongly reduces glioblastoma growth (Corsten et al., 2007). The second mRNA, with 5 edges, is \textit{miR-34a}. This miRNA is important for inducing apoptosis in cancer cells (Chen et al., 2012), also in glioblastoma (Yin et al., 2013). The third mRNA, with 4 edges is \textit{miR-9}, which has been detected as a highly connected gene in glioblastoma before (Sun et al., 2012) and has been implicated in glioblastoma progression (Ben-Hamo and Efroni, 2011).

The strongly connected mRNAs might also point to potential targets for glioblastoma treatment. A better understanding of the genes that are differentially expressed in glioblastoma is important to treat this highly resistant type of cancer (Eisele and Weller, 2013).
Figure 9: The apoptosis pathway, with edges given as in map04210 according to KEGG.
Figure 10: The apoptosis pathway, the moralised version of the edges given as in map04210 according to KEGG.
Figure 11: GGM lasso fit (undirected graph) of the apoptosis data.
Figure 12: ML-SEM lasso fit (directed graph) of the apoptosis data, with the same Markov properties as the undirected GGM lasso fit.
8 Discussion

This thesis explores graphical models and discusses three approaches of fitting them in a penalised way. The first approach, Gaussian graphical modelling (GGM), has been used before in its lasso form (Friedman et al., 2008) and in its ridge form (Stifanelli et al., 2012). The second approach, ML-SEM, starts with these GGM models and in a second stage incorporates prior knowledge to obtain a directed graph. The third approach, OLS, starts with an approximate estimate of a directed graph and is then moralised to an undirected graph.

The lasso versions have the advantage that edges are set to zero, so edge selection is built in. The lasso has the feature that for sets of variables with a high correlation often only one variable is selected in a model (Zou and Hastie, 2005). If the truth is not so sparse, ridge methods might work better. However, in ridge models one has to perform variable selection, which was done in a rather naive way in this thesis. The value of fitting the ridge versions in the three approaches mainly lies in showing that it can be used if one has a suitable method for determining a cut-off. Moreover, their performance in the estimation of edge magnitudes is not considered here. The performance in edge selection was close to that of the lasso versions in most sampling schemes. An important possible addition to the penalisations is allowing for different penalty parameters \( \lambda \) for different (types of) edges. In the OLS lasso approach we have seen that a single penalty parameter for all edges favours selection of edges of one type, which could be adjusted by allowing for multiple penalty parameters.

The ML-SEM and OLS approaches incorporate an assumption about the causal relationships in the data: miRNAs affect mRNA expression and not the other way round. However, the expression of miRNA genes is also regulated by mRNA genes (Pasquinelli, 2012). By focussing on a relatively small proportion of all mRNAs, as in the apoptosis pathway, and including only miRNAs that are known to target mRNAs in this pathway, we might hope that the miRNA effects on mRNAs dominate. In the apoptosis pathway of the case study, the gene \( P53 \) is involved in miRNA regulation, also of the miRNA \( \text{miR-34a} \) (Hermeking, 2012). This indicates that the assumption that miRNAs cause changes in mRNA expressions and not the other way round is too strong.

The ML-SEM approach yields an estimate of a directed network. A directed network gives more information than an undirected network about causality (Spirtes et al., 2004), so it has added value if the estimate is reliable. In our simulations the performance of ML-SEM is good as long as its assumptions are not violated. There is a one-stage alternative for the two-stage ML-SEM approach: rather than fitting a directed model that is Markov equivalent to a penalised estimate of an undirected model, one could fit a penalised directed model directly. To do this the likelihood function of equation (9) can be extended with a lasso or ridge penalty over the \( \mathbf{B} \) and \( \mathbf{\Gamma} \) parameters. With the software packages used in this thesis it is however very challenging to find converging algorithms for these penalised SEMs. This is the reason the two-stage approach was chosen.

The OLS approach is a limited information approach in which not all parameters are modelled. It nevertheless performs better than the GGM approach in
some simulation schemes. For datasets of the size used in the simulation OLS is considerably faster than the ML-SEM approach. However, for higher dimensional datasets the method is less suited. This can be seen for the ridge version, since it involves inversion of the \([p(p-1)/2 + pq] \times [p(p-1)/2 + pq]\) matrix (\(\mathbf{R}^T \mathbf{R} + \lambda \mathbf{I}\)), compared to the \([p + q] \times [p + q]\) matrix in the GGM approach.

This thesis has explored just a few types of models to find a graphical model topology for combined mRNA and miRNA data. It shows that incorporating prior information about causal relationships can pay off. For the proposed approaches to be useful in daily practice more research is needed in finding optimal penalty parameters and cut-offs.

**Bibliography**


A Derivation of the maximum likelihood estimate for $\Sigma$

Define the observed and centred realisations of $Y$ as $Y \in \mathbb{R}^{n \times p} = [y_1, ..., y_n]^T$. Take the vector of observed centred variable values $y_i \in \mathbb{R}^p = [y_{i1}, ..., y_{ip}]$ for $i = (1, ..., n)$ and assume it follows a normal distribution. Denote this distribution function by $f(Y)$, so

$$f(Y) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^\frac{p}{2}}\exp\left(-\frac{1}{2}Y^T\Sigma^{-1}Y\right)$$

Now, we can write the likelihood of $Y = [y_1, ..., y_n]^T$ as a function of $\Sigma$:

$$L(\Sigma|y_1, ..., y_n) = f(y_1; \Sigma) \cdot f(y_2; \Sigma) \cdots f(y_n; \Sigma) = (2\pi)^{-\frac{np}{2}}|\Sigma|^{-\frac{np}{2}}\exp\left(-\frac{1}{2}\sum_{i=1}^{n}y_i^T\Sigma^{-1}y_i\right).$$

For maximisation of the likelihood we take the logarithm:

$$\log[L(\Sigma|y_1, ..., y_n)] = \ell(\Sigma|Y) = -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{i=1}^{n}y_i^T\Sigma^{-1}y_i.$$
For a given dataset $\frac{\eta}{2} \log(2\pi)$ and $\frac{n}{2}$ are constants, so the maximum likelihood estimate is

$$\max_{\Sigma} [-\log(|\Sigma|) - \text{tr}(\Sigma^{-1}S)].$$

(13)

Alternatively we can find

$$\max_{\Omega} [\log(|\Omega|) - \text{tr}(\Omega S)],$$

(14)

for the precision matrix $\Omega$. These two equations are maximised by the empirical biased covariance matrix $\frac{1}{n} \sum_{i=1}^{n} \frac{y_i y_i^T}{n}$ and its inverse respectively. To see this, take the derivative of equation (14) using Harville (1997):

$$\frac{\partial}{\partial \Omega} [\log(|\Omega|) - \text{tr}(\Omega S)] = (\Omega^{-1})^T - S^T,$$

and set to zero to obtain:

$$\Omega^{-1} = S.$$

This shows that the maximum likelihood estimates of the covariance and the precision are the empirical covariance and its inverse.

### B R code for simulation

```r
library(mvtnorm)
pqchoice <- c("p15q35", "p25q25", "p35q15")
Bchoice <- c("Bband", "Bblock", "Bpath")
Gchoice <- c("Geach2", "Grange5")
choices <- expand.grid(pqchoice, Bchoice, Gchoice)
for (i in 1:18) choices$name[i] <- paste(as.matrix(choices[i, 1:3]), sep = "", collapse = ".")
# RUN ONCE - pathway for Bpath ####
load("KEGGadjMat.Rdata")
Path15 <- adjMat4210[1:15 + 10, 1:15 + 10]
Path25 <- adjMat4210[1:25 + 10, 1:25 + 10]
Path35 <- adjMat4210[1:35 + 10, 1:35 + 10]
# RUN ONCE - make files with covariance matrices
# ####
set.seed(4)
for (choice.ind in 1:18) {
  name <- choices$name[choice.ind]
  print(name)
  # p, q ###
  if (choices[choice.ind, 1] == "p15q35") {
    p <- 15
    q <- 35
  } else if (choices[choice.ind, 1] == "p25q25") {
```
p <- 25
q <- 25
}
else {
p <- 35
q <- 15
}
OmegaYX.SIM <- t(diag(p) - B.SIM) %*% solve(Psi.SIM) %*% G.SIM
Omega.SIM <- rbind(cbind(OmegaYY.SIM, OmegaYX.SIM),
                   cbind(t(OmegaYX.SIM), matrix(, q, q)))
diag(Omega.SIM) <- NA
parcor.non0 <- Omega.SIM != 0
# Save sampling covariances ###
rownames(Sigma.SIM) <- colnames(Sigma.SIM) <- c(paste("y",
                   1:p, sep = ","), paste("x", 1:q, sep = ","))
attr(Sigma.SIM, "parcor.non0") <- parcor.non0
attr(Sigma.SIM, "name") <- name
attr(Sigma.SIM, "pars") <- list(p = p, q = q, B.SIM = B.SIM,
                                  G.SIM = G.SIM, Psi.SIM = Psi.SIM)
save(Sigma.SIM, file = paste("final", name, ".RData",
                              sep = ""))
}
# RUN ONCE - make files with samples #####
n <- 100
for (sigma.ind in 1:18) {
  name <- choices$name[sigma.ind]
  load(paste("final", name, ".RData", sep = ""))
samples <- vector("list")
  for (sample.ind in 1:20) {
    W <- scale(rmvnorm(n, rep(0, p + q), Sigma.SIM),
               scale = F)
samples[[sample.ind]] <- W
  }
  attr(samples, "pars") <- list(p = attr(Sigma.SIM,
                                   "pars")$p, q = attr(Sigma.SIM, "pars")$q)
save(samples, file = paste("final", name, ".samples",
                           ".RData", sep = ""))
}

C R code for model fitting

library(Matrix)
library(glasso) # Friedman et al. (2008)
library(sem) # Fox (2006)
library(penalized) # Goeman (2010)
### GGM
coef.GGM <- function(Y, X, nu, pen = "ridge", ridgecut = 0.1) {
  lambda <- (1/(1 - nu))^2 - 1
  covW <- cov(cbind(Y, X))
  n <- nrow(Y)
  pen <- match.arg(pen, c("ridge", "lasso"))
  if (pen == "lasso") {
    Rho <- glasso(covW, lambda)$wi != 0
  } else {
    a.eig <- eigen(lambda^2 * diag(p + q) + covW)
    a.sqrt <- a.eig$vectors %*% diag(sqrt(a.eig$values))
solve(a.eig$vectors)
Rho <- abs(-cov2cor(solve(a sqrt + 0.5 covW))) >
ridgecut
}
Btest.GGM <- Rho[1:p, 1:p]
Gtest.GGM <- Rho[1:p, (p + 1):(p + q)]
c(Btest.GGM[lower.tri(Btest.GGM)], c(Gtest.GGM))
}
### OLS
coeff.reg <- function(Y, X, nu, pen = "ridge", ridgecut = 0.1) {
  lambda <- ((1/(1 - nu))^2 - 1) * 100
  v <- matrix(Y)
  index <- combn(1:p, 2)
  R <- matrix(0, n * p, npar)
  for (icomb in 1:ncomb)
  {
    k <- index[1, icomb]
    l <- index[2, icomb]
    R[((k - 1) * n + 1):(k * n), icomb] <- Y[, l]
    R[((l - 1) * n + 1):(l * n), icomb] <- Y[, k]
  }
  R[, (ncomb + 1):npar] <- diag(p) %x% X
  pen <- match.arg(pen, c("ridge", "lasso"))
  if (pen == "lasso") {
    lassofit <- penalized(v, R, unpenalized = ~0,
      lambda1 = lambda)
    bvec <- round(coef(lassofit, "penalized"),
      3)
    G.reg <- matrix(bvec[(ncomb + 1):npar], p,
      q, byrow = T) # from byrow to bycolumn
    B.reg <- matrix(0, p, p)
    B.reg[lower.tri(B.reg)] <- bvec[1:ncomb]
    B.reg[upper.tri(B.reg)] <- t(B.reg)[upper.tri(B.reg)]
    Nonzero.YY <- (t(diag(p) - B.reg) \%\% (diag(p) -
      B.reg)) != 0
    Nonzero.YX <- (t(diag(p) - B.reg) \%\% G.reg) !=
      0
  } else {
    avec <- numeric(p * (p - 1)/2 + p * q)
    bvec <- numeric(p * (p - 1)/2 + p * q)
    cvvec <- crossprod(R, v)
    tRR <- crossprod(R)
    tL <- chol(tRR + diag(rep(lambda, p * (p -
      1)/2 + p * q)))
    avec <- forwardsolve(tL, cvvec, upper = TRUE,
      transpose = TRUE)
    bvec <- backsolve(tL, avec)
    Psi.inv <- diag(1/diag(cov(matrix(v - R \%\%
      bvec, , p))))
    CovX <- (tRR[(npar - q + 1):npar, (npar - q +
      1):npar] + diag(rep(lambda, q)))/n
    G.reg <- matrix(bvec[(ncomb + 1):npar], p,
q, byrow = T) # from byrow to bycolumn
B.reg <- matrix(0, p, p)
B.reg[lower.tri(B.reg)] <- bvec[1:ncomb]
B.reg[upper.tri(B.reg)] <- t(B.reg)[upper.tri(B.reg)]
Omega.YY <- t(diag(p) - B.reg) %*% Psi.inv %*%
(diag(p) - B.reg)
Omega.YX <- -t(diag(p) - B.reg) %*% Psi.inv %*%
G.reg
Omega.XY <- t(Omega.YX)
Omega.XX <- solve(CovX) + t(G.reg) %*% Psi.inv %*%
G.reg
Rho <- -cov2cor(rbind(cbind(Omega.YY, Omega.YX),
               cbind(Omega.XY, Omega.XX)))
Nonzero.YY <- abs(Rho[1:p, 1:p]) > ridgecut
Nonzero.YX <- abs(Rho[1:p, (p + 1):(p + q)]) > ridgecut
#
}
c(Nonzero.YY[lower.tri(Nonzero.YY)], c(Nonzero.YX))

## ML-SEM
coeff.ML <- function(Y, X, nu, pen = "ridge", ridgecut = 0.1) {
  lambda <- (1/(1 - nu))^2 - 1
  covW <- cov(cbind(Y, X))
  n <- nrow(Y)
pen <- match.arg(pen, c("ridge", "lasso"))
  if (pen == "lasso") {
    glassofit <- glasso(covW, lambda)
    covmat <- glassofit$w
    parcor0 <- glassofit$wi == 0
  } else {
    a.eig <- eigen(lambda^2 * diag(p + q) + covW %*% covW/4)
    a.sqrt <- a.eig$vectors %*% diag(sqrt(a.eig$values)) %*%
              solve(a.eig$vectors)
    ridgefit0 <- abs(-cov2cor(solve(a.sqrt + 0.5 *
                                covW))) < ridgecut
    glassofit <- glasso(covW, zero = which(ridgefit0,
                      arr.ind = T), rho = 0)
    covmat <- glassofit$w
    parcor0 <- ridgefit0
  }
  index_y <- combn(1:p, 2)
  index_x <- combn(1:q, 2)
  manifests <- c(paste("y", 1:p, sep = ""), paste("x",
               1:q, sep = ""))
  rownames(covmat) <- colnames(covmat) <- manifests
  parcor0B <- parcor0[1:p, 1:p]
  parcor0Bvec <- parcor0B[lower.tri(parcor0B)]
  parcor0G <- parcor0[1:p, (p + 1):(p + q)]
  parcor0Gvec <- c(parcor0G)
  model <- rbind(cbind(1, t(index_y), 1:ncomb, NA),
                 cbind(1, t(index_y)[, 2:1], 1:ncomb, NA),
                 cbind(1, rep(1:p, q), rep(p + 1:q, each = p), (ncomb +

n.orig <- nrow(model)
model <- model[!parcorOBvec & !parcorOBvec & !parcorOBvec,
rep(T, n.orig - (ncomb + npar)), ]
model[, 4] <- as.numeric(as.factor(model[, 4]))
param.names <- c(paste("b", index_y[1, ], "_", index_y[2, ], sep = ""), paste("g", rep(1:q,
each = p), "_", rep(1:p, q), sep = ""), paste("var",
manifests, sep = ""), paste("cov_x", index_x[1,
], "_", index_x[2, ], sep = ""))[!parcorOBvec & !parcorOBvec & rep(T, n.orig - (ncomb + npar))]
SEMfitsym <- sem(model, covmat, N = n, param.names = param.names,
var.names = manifests, optimizer = optimizerOptim)
K.sym <- SEMfitsym$A
B.sym <- K.sym[1:p, 1:p]
G.sym <- K.sym[1:p, (p + 1):(p + q)]
round(c(B.sym[lower.tri(B.sym)], c(G.sym)), 1) != 0
}