Estimation of covariance and precision matrix, network structure and a view towards systems biology

Markku O. Kuismin [*], Mikko J. Sillanpää [*†]

Article Type:

Overview

Abstract

Covariance matrix and its inverse, known as the precision matrix, have many applications in multivariate analysis because their elements can exhibit the variance, correlation, covariance and conditional independence between variables. The practice of estimating the precision matrix directly without involving any matrix inversion has obtained significant attention in the literature. We review the methods that have been implemented in R and their R packages, particularly when there are more variables than data samples and discuss ideas behind them. We describe how sparse precision matrix estimation methods can be used to infer network structure. Finally, we discuss methods that are suitable for gene co-expression network construction.

[*]Department of Mathematical Sciences, University of Oulu, Finland
[†]Biocenter Oulu, University of Oulu, Finland

INTRODUCTION

A covariance matrix is an essential part in several multivariate analysis methods such as pattern recognition using linear (quadratic) discriminant analysis, principal component analysis and cluster analysis to name a few (see e.g Mardia[1] and McLachlan[2]). Because the inverse of the covariance matrix plays an essential role in Gaussian graphical models, there is also interest in directly estimating the inverse of the covariance matrix, known as the *precision matrix*. The obvious aim of these analyses is to produce reliable and easy to interpret information over the characteristics of interest. It is natural to assume that when the sample size increases we gain better and reliable estimates. However, in some research fields, such as in biology, the number of observations available may by very limited compared to the number of variables of interest. In particular, when there are more variables than data points, estimation problem of the precision matrix is ill-posed.

Let $Y_i = (Y_{1i}, \ldots, Y_{pi})^\top$ be a vector of length p. The symbol $Y_i^\top$ is used to designate the transpose of Y_i. The entries in Y_i are independent random samples from a multivariate Gaussian distribution $N(\mu, \Sigma)$, where μ is the mean vector of length p and Σ is the $p \times p$ symmetric and positive definite covariance matrix. In some cases it may be more convenient to express the Gaussian distribution in the form of $N(\mu, \Theta^{-1})$, where Θ is the inverse of the covariance matrix Σ (the precision matrix). We can assume that μ is a zero vector.

A covariance matrix Σ is always symmetric and all of its diagonal elements need to be positive. One important property of covariance matrix is that it has to be positive semi-definite, which means that there cannot be any linear dependencies between rows or columns of the matrix (i.e., it has to be full rank) and for every non-zero vector x of the length p, $x\Sigma x \geq 0$. When the matrix is positive semi-definite, it is non-singular and it has a unique inverse that can be calculated. For a positive definite matrix it holds that $x\Sigma x > 0$.

Usually the covariance matrix is estimated with the sample covariance matrix, denoted by S, from a sample of the size n

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (Y_i - \overline{Y})(Y_i - \overline{Y})^\top, \tag{1}$$

where $\overline{Y}$ is the sample mean vector. Remembering the assumption that the mean vector

is a zero vector or that the variables have been centered at zero, $\overline{Y}$ can be omitted from equation (1). The maximum likelihood estimate of the covariance matrix can be calculated by replacing $n-1$ in the equation (1) (also known as the unbiased estimate) with n.

Usually in any statistical analysis one assumes to have relatively large number of independent and unbiased random samples from the appropriate distribution. According to the law of large numbers, the sample covariance (1) is approximately equal to the real covariance matrix. It can be shown that when the sample size is large compared to the number of variables p, then the sample covariance matrix is actually positive definite and always non-singular. In this case one can estimate the precision matrix Θ by inverting S.

Nevertheless, in many real life data analyses the number of samples is not substantially large compared to the number of variables p and sometimes there may be more variables of interest than samples to estimate them. When the dimension p is larger (or even much larger) than the sample size n, the sample covariance matrix S cannot be inverted. This is due to the fact that the number of non-zero eigenvalues of the sample covariance is always $\min(n, p)$. The number of non-zero eigenvalues of S has to be equal to p for S to be non-singular. This case is known as "large p small n" in the literature about high-dimensional covariance matrix estimation, usually denoted by $p \gg n$. There are covariance matrix estimation methods which ensure that the resulting matrix is automatically positive semi-definite but some of them require application of ad-hoc adjustments afterwards to obtain positive semi-definite estimate. More intuitively, because the sample covariance matrix depends on the number of sample values, data with small sample sizes cannot be used to accurately estimate the true covariance matrix even if it would not be singular. In the next section, we present a selection of methods for more reliable covariance and the precision matrix estimation.

Alternative estimators for covariance and precision matrix

We restrict our discussion to estimators that always produce symmetric and positive definite estimates, even when the sample size is smaller than the number of variables. This restriction rules out some estimators, such as the Stein-type estimator proposed by Stein[3] and multiple testing procedures of Drton and Perlman[4,5] because they do not always produce positive

definite estimates. Readers interested in these methods should check the original papers and R packages `ShrinkCovMat` (version 1.1.2) and `SIN` (version 0.6) respectively. See also a recent article of Naul and Taylor[6].

In practice it seems that methods with ready to use software are the ones gaining more attention and thus are applied to practical data analyses. Therefore, we review methods that are implemented in R (available from CRAN - Comprehensive R Archive Network) (Table 1). We also discuss the network estimation methods introduced by Meinshausen and Bühlmann[7] and Zhang and Horvath[8] to give better insight into the diverse world of network estimation. In the statistical literature, there is an extensive discussion about regularized covariance and precision matrix estimation, which is covered very computation-centric way such that the estimation of each likelihood and penalty combination is considered separately and very deeply from computational point of view. This creates very scattered literature[9–15]. We emphasize that our review contains purely a selective setting of estimators. In the next subsections we briefly describe the methods presented in Table 1. We end the review discussing some methods which do not fit in the framework described in Table 1.

Method	$\widehat{\Sigma}$	$\widehat{\Theta}$	**Tuning parameter selection methods**	**Package name**	**Package version**
LW-estimators	Yes	No	Not needed	`nlshrink`	1.0.1
Glasso	Yes	Yes	eBIC, RIC, StARS	`glasso` and `huge`	1.8 and 1.2.7
QUIC	No	Yes	Not available	`QUIC`	1.1
BIGQUIC	No	Yes	StARS	`BigQuic`	1.1-7
ROPE	No	Yes	(approximate) leave-one-out CV, k-fold CV	`rags2ridges`	2.0
CLIME	No	Yes	k-fold CV, StARS	`clime`, `fastclime` and `camel`	0.4.1, 1.4.1 and 0.2.0
SCIO	No	Yes	CV	`scio`	0.6.1
spcov	Yes	No	Not available	`spcov`	1.01
TIGER	No	Yes	StARS, k-fold CV	`camel`	0.2.0
CondReg-estimator	Yes	Yes	k-fold CV	`CondReg`	0.20
MB-approximation	No	No	RIC, StARS	`glasso` and `huge`	1.8 and 1.2.7

Table 1: An eclectic collection of different estimators either for the covariance ($\widehat{\Sigma}$) or for the precision matrix ($\widehat{\Theta}$) and their R packages, what tuning parameter selection methods are readily available in the package, the package name and the current package version.

Convex combination estimators

The easiest way to circumvent the problem that the sample covariance matrix is singular, is to add a positive constant to the matrix diagonal: $\widehat{\Sigma}_\alpha = S + \alpha I$, where α is a positive coefficient and I is the $p \times p$ identity matrix. This estimator is so called *ridge estimator* (see, e.g. Warton[16]) and it is a special case of estimators of the form

$$\widehat{\Sigma} = \alpha_1 S + \alpha_2 T, \tag{2}$$

where α_1, α_2 are positive coefficients and T is $p \times p$ target matrix. Note that $\widehat{\Sigma}$ is a convex combination of S and T. The valuable property of the estimators in (2) is that they are quite robust, in that they do not assume the data to be Gaussian.

Ledoit and Wolf[17,18] examined the convex combination (2) of the form

$$\widehat{\Sigma} = \alpha S + (1 - \alpha) T \tag{3}$$

and referred their estimator as a *linear shrinkage estimator*. Namely, linear shrinkage estimator shrinks the sample covariance matrix towards a scaled identity matrix. The linear shrinkage estimator can be computed from (3) by choosing $T = \mathrm{tr}(S)/pI$, where $\mathrm{tr}(S)$ is the matrix trace and the coefficient α is determined empirically from the data.

The starting point of the shrinkage estimator is that the sample covariance matrix seems to overestimate the large eigenvalues and underestimate the small eigenvalues compared with the true covariance matrix eigenvalues[18]. The valuable property of the linear shrinkage estimator is that it will decrease the large eigenvalues of the sample covariance matrix and at the same time increase the small eigenvalues of the sample covariance matrix in a "linear" manner.

Ledoit and Wolf[19,20] extended this approach and proposed a *nonlinear shrinkage estimator*. Ledoit and Wolf computed this estimator by applying a nonlinear shrinkage formula to the sample covariance matrix eigenvalues. The nonlinear shrinkage estimator "corrects" the eigenvalues in a more complicated manner than the linear shrinkage estimator; the nonlinear shrinkage estimator has very convoluted formulation and it is discussed here. The practical starting point behind the nonlinear shrinkage estimator is that one can gain more

reliable information of the total variation explained by the principal component in principal component analysis. Because the large sample covariance matrix eigenvalues are biased upwards compared to the true eigenvalues of the covariance matrix Σ, the variances of the largest principal components are overestimated. A nonlinear shrinkage estimator will reduce this overestimation. Both linear and nonlinear estimators (LW-estimators in Table 1) are implemented in the R package nlshrink.

Graphical lasso and alternative ridge estimator

The joint distribution function of multivariate Gaussian data $Y = (Y_1, \ldots, Y_n)^\top$ is

$$p(Y|\Sigma) = (2\pi)^{-pn/2}|\Sigma|^{-n/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{n} Y_i^\top \Sigma^{-1} Y_i\right), \tag{4}$$

where $|\Sigma|$ is the determinant of the covariance matrix. The log-likelihood function (4) is more convenient to express as the function of the $n \times p$ data matrix Y and the precision matrix Θ up to a constant,

$$\log p(Y|\Theta) \propto \log|\Theta| - \mathrm{tr}(S\Theta). \tag{5}$$

There is increasing interest in performing precision matrix estimation in a penalized likelihood framework. In this framework, evidence of the data measured by the likelihood (object function) is combined with the penalty function which can be interpreted as a constraint from optimization theory or as a prior from Bayesian inference viewpoint. This joint expression of likelihood and penalty is then optimized together to find pareto-optimum of this expression.

Graphical lasso (Glasso)[9] is probably the best known precision matrix estimation method (with more than 914 citations to the paper of Friedman et al.[9] alone, Web of Science database, 2 August 2017). Potential solutions to the Glasso problem have been proposed in numerous research articles[9–13,21–23]. The Glasso problem can be expressed as a maximization problem of the *penalized* log-likelihood of the form

$$\log|\Theta| - \mathrm{tr}(S\Theta) - \lambda||\Theta||_1, \tag{6}$$

over positive definite matrices Θ. Here $||\Theta||_1$ is the L_1-norm of the matrix, computed as the sum of the absolute value of the elements of Θ and λ is a positive *tuning parameter* which controls the number of zero entries in the final estimate of the precision matrix[10]. Three valuable properties of the L_1-norm as the penalty function in (6) are: (i) it shrinks the elements of the precision matrix towards zero (ii) it simultaneously tests if the elements of the precision matrix could be set to zero (iii) the final (sparse) estimate will be positive definite even in high-dimensional setting.

Solving the Glasso problem is not straightforward and numerous optimization methods are available for the maximization of the L_1-penalized log-likelihood (6). Here different algorithms are mentioned briefly to clarify the differences between these methods, though the original papers provide more detailed descriptions. The first methods to solve the Glasso problem were proposed by Banerjee et al.[22] and Friedman et al.[9]. Banerjee et al. and Friedman et al. examined the dual form of the Glasso problem

$$\widehat{\Sigma} = \max \left\{ \log |W| : ||W - S||_\infty \leq \lambda \right\}, \tag{7}$$

where W is the estimate of the covariance matrix Σ and $||W - S||_\infty$ denotes the maximum absolute value element of the symmetric matrix $W - S$. Banerjee et al.[22] uses a block-coordinate descent algorithm to solve the dual problem (7) by updating one row and column of W at a time; these lower dimensional problems can be written as LASSO problems[24]. Finally, Nesterov's first order method is used to solve LASSO subproblems[22]. Friedman et al.[9] developed Banerjee et al's algorithm further and proposed the use of the coordinate descent method. Witten et al.[10] showed that a block diagonal screening rule can reduce computational time to determine a sparse precision matrix estimate; this method is implemented in the original R package `glasso` (version 1.8). In addition to the aforementioned algorithms, Hsieh et al.[13] introduced an algorithm which combines quadratic approximation, Newton's method and coordinate descent. This algorithm is implemented in the R package `QUIC` (version 1.1). Furthermore, the algorithm called BIGQUIC is implemented in the R package `BigQuic` (version 1.1-7) which uses Newton's method, coordinate descent[9], and METIS clustering[25] to approximate a sparse precision matrix with up to one million variables[12]. There are also MATLAB implementations available for QUIC and BIGQUIC

(see Hsieh et al.[13] and Hsieh et al.[12]).

In covariance and precision matrix estimation it is useful to generate a heatmap from the computed estimate to gain an insight into the (true) covariance or precision matrix structure. We simulated data with 95 samples ($p = 100$) from a Gaussian distribution $N(0, \Sigma)$ and compared the ground truth with the sample covariance and Glasso estimate (Figure 1). The covariance matrix Σ considered in this small example is a sparse matrix with a special block-structure; the non-zero off-diagonal elements are set to the value 0.75 and diagonal elements to the value 1.

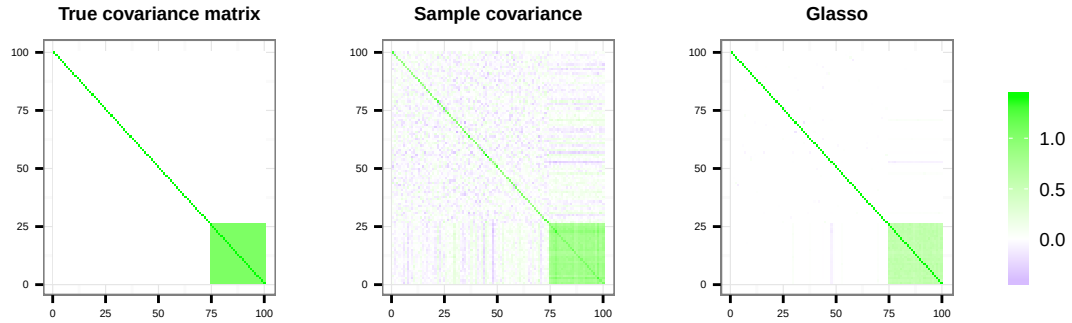

Figure 1: Heatmaps of the estimated covariance matrix when the ground truth is known. We have used $p = 100$ and $n = 95$ as a representing example. Note that the non-zero elements of the covariance matrix estimated with Glasso are somewhat smaller than their counterparts in the sample covariance matrix, due to the shrinkage effect of the L_1-penalty.

The inclusion of penalty function to the likelihood expression makes parameter estimation also possible in oversaturated or ill-posed situations where classic maximum likelihood estimate does not exist. By including constraints or prior information to the oversaturated or ill-posed problems, we make a specific statement about which solution(s), out of many possible ones, we are interested in obtaining.

Recently, there has been critique that the ridge estimator $\widehat{\Sigma} = S + \alpha I$ does not resemble the original ridge penalty[26,27] but is determined from different kind of penalized log-likelihood function[16]. Therefore, a novel alternative ridge estimator, the Ridge Operated Precision Matrix Estimator (ROPE), which uses penalty function more consistent with the L_2-penalty function of the ridge-regression, has been introduced for precision matrix estimation[26,27].

The proposed estimate maximizes the penalized log-likelihood of the form

$$\log|\Theta| - \mathrm{tr}(S\Theta) - \lambda||\Theta||_{\mathrm{F}}^2, \tag{8}$$

where $||\Theta||_{\mathrm{F}}$ is the Frobenius norm of the matrix, computed as the squared root of the sum of squared absolute value of the elements of Θ. Using a more consistent "ridge-penalty" does not substantially complicate the computation of the final estimate and, moreover, the estimate has a simple closed form solution[26,27]. In addition, one can utilize a special target matrix T possibly carrying some prior information. This target matrix makes the estimator potentially more adaptable to some special data-analysis problems,

$$\log|\Theta| - \mathrm{tr}(S\Theta) - \lambda||\Theta - T||_{\mathrm{F}}^2. \tag{9}$$

One difference of (alternative) ridge estimators compared with Glasso is that they only shrink the elements of the precision or the covariance matrix, but do not produce a sparse estimate; this may be a desired property for some data applications such as linear discriminant analysis (LDA) and portfolio optimization[17]. Another difference between ridge-type estimators, alternative ridge estimators and Glasso is that the estimators are *rotation equivariant* (when the target matrix T is a diagonal or a zero matrix). The eigenvectors of the rotation equivariant estimators are the same as the eigenvectors of the sample covariance matrix (see, e.g. Ledoit and Wolf[18], Kuismin and Sillanpää[28] and Kuismin et al.[27]).

Constrained estimators

Alternative estimators based on constrained optimization, rather than the penalized log-likelihood, have been proposed. For example, Cai et al.[29] proposed the Constrained L_1-minimization for Inverse Matrix Estimation (CLIME)

$$\text{Minimize } ||\Theta||_1 \text{ subject to: } ||S\Theta - I||_\infty \leq \rho, \tag{10}$$

where ρ is a positive tuning parameter. The main idea behind CLIME is to estimate each column of the precision matrix at a time using a sparse linear regression to reduce the dimensionality of the computational problem. Related to the CLIME estimator (10), Liu

and Luo[14] proposed a novel method called Sparse Column-wise Inverse Operator (SCIO) as a fast method suited for large scale ($p \geq 800$) precision matrix estimation. In addition, Liu and Wang[15] proposed another method called TIGER (Tuning-Insensitive Graph Estimation and Regression) to solve the same L_1-constrained problem. The only difference between CLIME and TIGER is how they solve the sparse linear regression problem: CLIME uses the Dantzig selector[30], SCIO uses an iterative coordinate descent algorithm[14] and TIGER uses the SQRT-LASSO[31]. Readers interested in using a method similar to SCIO, CLIME and TIGER with MATLAB should refer to Yuan[32].

All constrained estimators produce a sparse estimator which may not be symmetric but this can be fixed with a simple symmetrization step[14,29]. A valuable property of constrained estimators among Glasso and ROPE is that one can determine a direct estimate for the precision matrix rather than having to first compute an estimate for the covariance matrix and then invert it. Direct estimation of the precision matrix lowers the complexity of the estimator and reduces potential numerical error induced by the matrix inversion.

We have illustrated the precision matrices estimated with CLIME, SCIO and TIGER in Figure 2 and compared the results with simulated ground truth. The precision matrix considered here corresponds to a cluster graph structure produced with the R package `huge`[33]. For sensible graphical representation, we plotted the heatmaps of the adjacency matrices $A = (a_{i,j})$, where $a_{i,j}$ is equal to one if the corresponding precision matrix element $\theta_{i,j}$ is non-zero, zero if the corresponding $\theta_{i,j}$ is also zero and all diagonal elements are equal to zero.

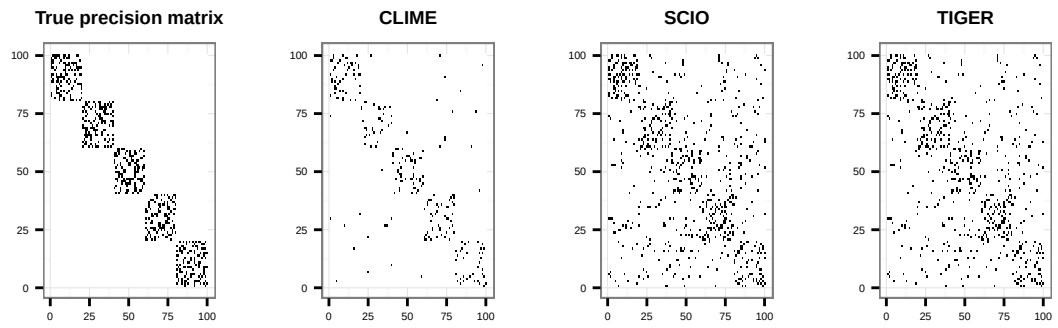

Figure 2: Heatmaps of the estimated adjacency matrices for CLIME, SCIO and TIGER when the ground truth is known. We have used $p = 100$ and $n = 95$ as a representive example. White means zero adjacency matrix (precision matrix) element and black is a non-zero element.

Other estimators

In addition to the convex combination, Glasso, alternative ridge and constrained estimators there are numerous other estimators available for covariance matrix estimation. For example sparse estimator of a covariance matrix is obtained by minimizing penalized log-likelihood

$$\log|\Sigma| + \mathrm{tr}(\Theta S) + \lambda||P \circ \Sigma||_1, \tag{11}$$

where $\circ$ denotes elementwise multiplication and P is a symmetric matrix with non-negative elements inducing weighted penalty for each element of Σ separately[34]. An algorithm to minimize (11) is available in the R package spcov. Use of a weighted penalties makes a direct link to adaptive LASSO in regression context[35].

Deng and Tsui[36] considered a special penalized log-likelihood. This method utilizes the matrix logarithm transformation $A = \log(\Sigma)$ and minimizes the expression

$$l(A) + \lambda||A^2||_{\mathrm{F}}^2, \tag{12}$$

where $l(A)$ is an approximation of the negative log-likelihood $\mathrm{tr}(A) + \mathrm{tr}\{\exp(-A)S\}$. The expression of $l(A)$ is quite complicated and is not presented here. The valuable property of the matrix logarithm transformation is that the penalty function will non-linearly regularize

both the largest and smallest eigenvalue of the covariance matrix. MATLAB code for this method is available in the supplementary materials of Deng and Tsui[36].

Won et al.[37] proposed a condition number constrained estimator (CondReg-estimator in Table 1) for the covariance matrix by solving

$$\text{Maximize } l(\Sigma) \text{ subject to: } \text{cond}(\Sigma) \leq \kappa_{\max}, \tag{13}$$

where $l(\Sigma)$ is the log-likelihood of (4), $\text{cond}(\Sigma)$ is the condition number of a positive definite covariance matrix Σ that is computed by dividing the largest eigenvalue of the matrix with the smallest eigenvalue. Threshold parameter $\kappa_{\max}$ which is smaller than the condition number of the sample covariance matrix S ($\kappa_{\max} < \text{cond}(S)$) can be chosen by using k-fold cross-validation (see the next subsection). A solution to the condition number constrained maximization problem (13) is implemented in the R package `CondReg` (version 0.20)[37] and it can be used for better conditioned precision and covariance matrix estimation.

Choosing the tuning parameter

The obvious problem with penalized likelihood and other regularized optimization problems is the choice of proper value for the tuning parameter λ. Tuning parameter is usually selected using a so called cross-validation scheme to determine which control parameter value can give the best performance for the model to predict future observations (see, e.g. Fang et al.[38]). One can use either some matrix norm as the cross-validation criterion[38], use some loss function[14,29] or use the log-likelihood (5)[27,34,36].

On the other hand, in instances where a sparse model is appropriate, cross-validation can be suboptimal because it tends to favor dense models[21]. An alternative to cross-validation is the *stability selection scheme* to determine which control parameter value can give the maximal stability to the model for small changes in the composition of the data[39].

In the statistical literature, several alternative methods have been developed for cross-validation, particularly to produce sparse network estimates. The most notable of these are extended Bayesian Information Criterion (eBIC)[23], Stability Approach to Regularization Selection (StARS)[21] and Rotation Information Criterion (RIC)[33]. eBIC, StARS and RIC

are implemented in the R package huge and can be used with Glasso, or with Meinshausen and Bühlmann approximation, which we discuss in the next section.

It is important to note that each selection scheme optimizes performance with respect to one criterion (e.g., predictive performance or stability of the model) and the apparently best parameter value may not be optimal with respect to some other criterion. Cross-validation is also very time consuming. Overall, selecting the tuning parameter is a challenging problem with no one-size-fits-all solution. In our experience one should consider the special characteristics of the application while selecting the tuning parameter value.

Network structure estimation

Estimation of a sparse precision matrix can be seen as a subproblem of the selection of Gaussian graphical models. The undirected graphical model G is usually defined as a set $G = (V, E)$, where $V = \{1, \ldots, p\}$ is the set of nodes and E set of edges (i, j), $i, j = 1, \ldots, p$. The pair (i, j) belongs to the set E if, and only if, the corresponding precision matrix element (i, j) is nonzero. That the precision matrix element (i, j) is zero implies conditional independence between variables i and j, given the rest of the variables. This conditional independence follows actually from the relation between the partial correlation matrix Q and the precision matrix Θ, $Q = -\text{diag}(\Theta)^{-1/2}\Theta\text{diag}(\Theta)^{-1/2}$, where $\text{diag}(\Theta)$ is a diagonal matrix constructed from the diagonal elements of Θ. From the equation of the partial correlation matrix one can see, that the connection between the elements of $Q = (q_{i,j})$ and $\Theta = (\theta_{i,j})$ can be expressed as $q_{i,j} = -\theta_{i,j}(\theta_{i,i}\theta_{j,j})^{-1/2}$. If the element $q_{i,j}$ is zero, the variable i is conditionally independent from j given the rest of the variables and vice versa. Information stored in Q can be changed to the binary conditional independence indicator form and arranged into a matrix, which is called the *adjacency matrix*.

Inducing sparseness to the precision matrix (network) is usually done in two ways; either by using (i) such penalty functions that are able to induce sparseness to the precision matrix by shrinking individual precision matrix elements towards zero, or (ii) by hypothesis testing[26,40–42] to decide which non-zero precision matrix elements could be set to zero. If latter method is applied, the sparse precision matrix is not guaranteed to be positive semi-definite.

Sparse precision matrix estimates computed with Glasso, QUIC, BIGQUIC, TIGER, CLIME and SCIO are always positive definite and can be used directly for network estimation.

With ROPE, the hypothesis testing approach of Wieringen and Peeters[26] utilizes the local false discovery rate procedure to make the network estimate sparse and the procedure is implemented in the R package ragas2ridges. Ha and Sun[42] used the ridge estimator to compute a positive definite estimate for the sample partial correlation matrix. In hypothesis testing, Ha and Sun used the Efrons's central matching method[43] to estimate the null distribution of the Fisher's z-statistic $0.5 \log (1 + q_{i,j})/(1 - q_{i,j})$; their method is implemented in the R package `GGMRidge` (version 1.1).

Additionally, there is a viable method available for sparse network estimation and it can be interpreted as an early approximation of the Glasso problem[9,22] which we will discuss next. Meinshausen and Bühlmann[7] (858 citations, Web of Science database 2 August 2017) provide an approximate way to estimate network structure by detecting zero entries in the precision matrix in column-by-column fashion. The approximation of Meinshausen and Bühlmann (hereafter MB-approximation) utilizes the connection between the elements of the precision matrix and LASSO-regression although MB-approximation does not provide reasonable numerical estimates for the (sparse) precision matrix elements. The same column-by-column evaluation method is utilized in CLIME, SCIO and TIGER for more efficient estimation of the high dimensional precision matrix. In contrast to CLIME, SCIO and TIGER, MB-approximation only provides an estimate for an adjacency matrix. MB-approximation can be seen as a way to transfer the network estimation problem to a standard problem of variable selection in linear regression, which means that regression-based variable selection methods can be used to solve the network topology estimation problem. We have computed two gene-expression network estimates using MB-approximation when both RIC and StARS are used to select the optimal tuning parameter value (Figure 3). The data considered here is a subsample of the riboflavin data set available in the R package hdi (version 0.1-6)[44] which consists of 200 genes with the largest empirical variance (see also Bühlmann et al.[45]).

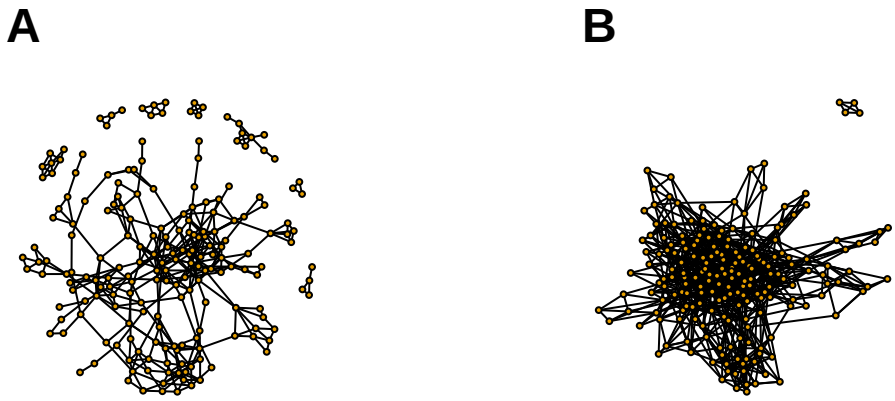

Figure 3: Graphs by Meinshausen and Bühlmann approximation estimated in a subsample of the riboflavin data set. **A** The optimal graph when the tuning parameter is chosen with RIC. **B** The optimal graph when the tuning parameter is chosen with StARS.

A special application to gene co-expression network estimation

As mentioned in the Section *Choosing the tuning parameter*, all previously mentioned estimators of Gaussian networks and tuning parameter selection methods may have theoretically pleasing properties which are not relevant to all data analyses. For example, beneficial properties hold for Gaussian data but often fail to account for certain characteristics or the design of biological data. One special case of this problem is when the data arise from a covariance structure commensurate to *scale-free network*. A scale-free network is characterized by few nodes that are highly connected to other nodes; these highly connected nodes are known as *hub nodes*. Finding hub nodes is very challenging problem (see, e.g. Krzakala et al.[46]). We have faced this challenge in practice while trying to utilize some of the above mentioned methods in gene network analysis. A small scale-free network is illustrated in Figure 4 along with Glasso and SCIO network estimates computed from a simulated data set.

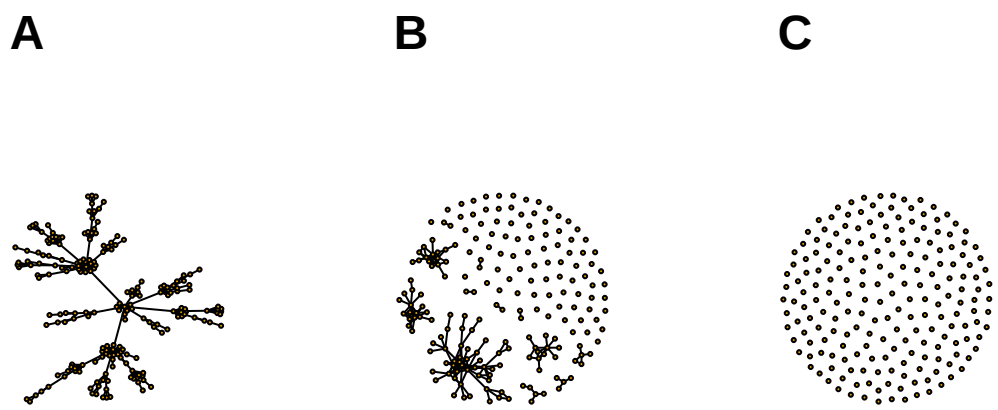

Figure 4: Estimated graph of simulated data set compared to the true graph structure. We have used $n = 100$ and $p = 200$ as a representing example. **A** The optimal graph. **B** The optimal graph estimated with Glasso when the tuning parameter is chosen with RIC. **C** The optimal graph estimated with SCIO when the tuning parameter is chosen with cross-validation.

It is apparent that networks estimated with Glasso and SCIO can have many free floating nodes, inconsistent of the underlying scale-free network structure (Figure 4). Examining too sparse network is problematic because some of the clusters and hub nodes remain undetected if the estimated network is too sparse; hub nodes may possess biologically meaningful information, for example, about diseases and disease genes[47,48]. In addition, many genes are co-expressed, meaning that there is always some type of inter-dependency between genes (see, e.g. Äijö and Bonneau[49]). The above mentioned tuning parameter selection methods do not take into account these special characteristics and may be ill-suited for gene network estimation.

When paying more attention to the role of scale-free network and co-expression between genes, Zhang and Horvath[8] proposed to estimate the adjacency matrix using a so called scale-free topology criterion. In particular, scale-free topology criterion is based on the

assumption that the degree distribution of the scale-free network follows a *power law*. The degree distribution is the probability distribution of the connections of each node.

Zhang and Horvath did not examine the covariance or precision matrix but the absolute values of the correlation matrix R,

$$R = (r_{i,j}) = (|\text{cor}(i,j)|), \tag{14}$$

where $\text{cor}(i,j)$ is the Pearson correlation between variables i and j. The elements of the adjacency matrix $A = (a_{i,j})$ were determined via so called hard-thresholding,

$$a_{i,j} = \begin{cases} 1, & \text{if } r_{i,j} \geq \tau. \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where τ is so called *hard threshold parameter*. As a side-note, similar hard-thresholding can be used to compute the solution to the Glasso problem[10,11] (see also Bickel and Levina[50]). Figure 5 contains a graphical representation of the network computed with the hard thresholding when the hard threshold parameter τ is chosen with the scale-free topology criterion. See also the tutorial of Zhang and Horvath[8].

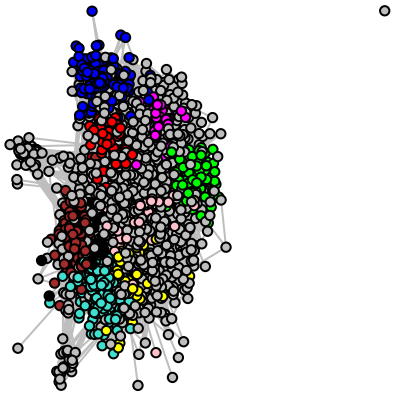

Figure 5: Gene co-expression network estimated with the hard-thresholding in a yeast microarray data[51] with 2001 different genes. Nodes are colored according to the modules represented in Figure 6.

After computing an adjacency matrix, hierarchical clustering is used to identify different modules (clusters) with coherent expression profiles. Hierarchical clustering is done based on a dendrogram by choosing a suitable height cutoff. We have illustrated module construction in Figure 6. Readers more interested in the interpretation of different modules should check the original paper of Zhang and Horvath[8] and the homepages of the WGCNA (https://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/).

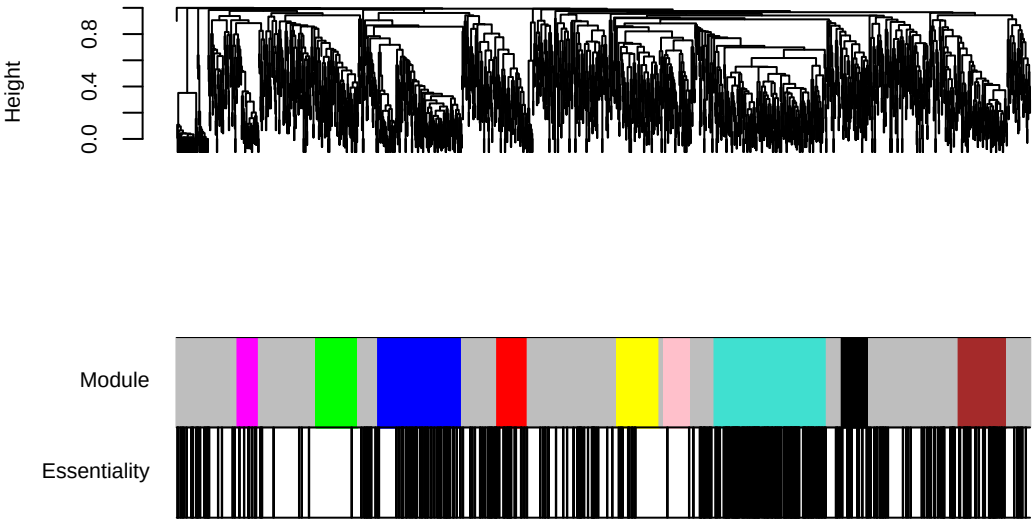

Figure 6: A dendrogram obtained from hierarchical clustering in a yeast microarray data with 2001 different genes (top). Distinct modules represented here with different colors are based on a height cutoff of the branches (middle). The bottom panel indicates essential genes in the data. Essential genes are more likely to be genes with high connectivity in the graph.

In addition, Zhang and Horvath[8] examined two soft-thresholding functions, from which the most commonly used one is determined as follows: The non-zero elements of the adjacency matrix $A = (a_{i,j})$ are computed with exponentiation,

$$a_{i,j} = |\mathrm{cor}(i,j)|^{\beta}, \tag{16}$$

where β is so called *soft-threshold parameter*. Zhang and Horvath[8] discovered that the soft-thresholding function (16) will produce networks carrying more biological information over the gene co-expression network than the corresponding sparse networks when the parameter β is chosen using the scale-free topology criterion. Soft-thresholding method has been used widely. For example, Zhang and Horvath[8] and Langfelder and Horvath[52] have been cited 1018 and 1341 times respectively (Web of Science, 2 August 2017) and used in diverse applications, such as cancer and yeast cell-cycle microarray samples[8] and mouse liver gene

expression data[53]. Weighted gene co-expression network analysis method of Horvath et al. is implemented in the R package wgcna (version 1.51) for efficient estimation of large gene co-expression networks[52].

Taking account of all the aspects mentioned above, methods for Gaussian graphical models should be modified to some extent before applying them to gene network analysis. For example, one can use some *a priori* information with Glasso algorithm to make it more adaptable for gene network analysis, as discussed by Li and Jackson[54]. Using external *a priori* knowledge Li and Jackson proposed a weighted graphical lasso (wglasso) method that uses different penalty values for different elements of the precision matrix similar to the penalty function in (11) and which makes Glasso more adaptable for network analysis in systems biology. In their simulation and a real data analysis, the weighted Glasso showed improved performance in network estimation compared to Glasso even with inaccurate *a priori* information. Figure 7 contains a network estimated in an *Arabidopsis thaliana* data set[55] when the optimal value for the wglasso tuning parameter is selected based on the minimum value of the corresponding Bayesian information criterion (BIC), defined as

$$BIC = -n \log p(Y|\Theta) + |E| \log(n), \tag{17}$$

where $\log p(Y|\Theta)$ is defined in (5) and $|E|$ is the number of edges in the estimated network.

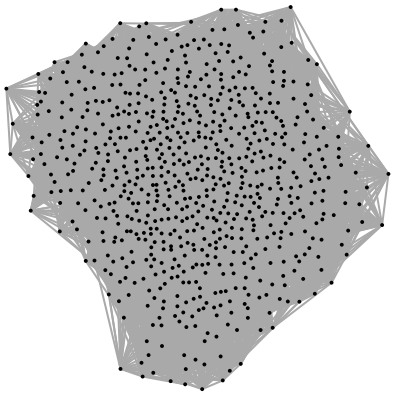

Figure 7: Gene co-expression network estimated with the weighted Glasso in expression data from the *Arabidopsis thaliana* with 739 different genes.

For other alternatives for WGCNA , see, e.g. Shimamura et al.[56] method similar to MB-approximation using weighted LASSO-regression and a recent article of Jokipii-Lukkari et. al.[57]. Readers interested in the adaptive LASSO procedure[35] as a part of a MB-approximation, see Krämer et al.[58] and R package `parcor` (version 0.2-6). Readers more familiar with the MATLAB, see Ruan et al.[59].

DISCUSSION

In addition to the frequentist methods presented in this review, there are some fully Bayesian methods to compute a posterior estimate for both the precision and covariance matrix and undirected Gaussian graphical models.

Khondker et al.[60] and Wang[61] both developed a Bayesian method for solving the graphical LASSO problem (6), where the posterior shrinkage estimate of the precision matrix is

computed with a random walk Metropolis-Hastings algorithm and a block Gibbs sampler, respectively. Bhadra and Mallick[62] introduced a Bayesian hierarchical framework to infer an undirected graph. The beauty of the approach of Bhadra and Mallick is in their adjacency matrix representation which was obtained by the analytic integration of numerical values away from the precision matrix. Kubokawa and Srivastava[63] proposed an empirical Bayesian approach to compute a ridge-type estimators for the precision matrix. Bouriga and Féron[64] utilized a hierarchical inverse-Wishart priors and used Metropolis-Hastings-within-Gibbs sampling scheme to estimate the posterior quantities in the posterior distribution of the covariance matrix. Huang and Wand[65] introduced also the inverse-Wishart distribution for the covariance matrix and examined the noninformative properties of both standard deviation and correlation parameters.

Readers interested in Bayesian tools should check the R package BDgraph (version 2.33)[66] that contains many tools to compute posterior estimates of the precision matrix and undirected graphs. Readers interested in co-expression networks should check the homepage of WGCNA with extensive tutorials and other materials https://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/.

Books and reviews dedicated to covariance and precision matrix estimation and graphical models, are presented by Hastie et al.[67,68], Pourahmadi[69], Tong et al.[70], Fan et al.[71].

ACKNOWLEDGMENTS

We would like to thank the Associate Editor, two anonymous referees, Phillip Watts and Markus Harju for their valuable comments, which helped us to improve the presentation of this paper. We thank Yupeng Li for providing the data and procedures from his paper. This work was supported by the University of Oulu's doctoral programme of Technology and Natural Sciences.

References

1. K. Mardia, J. Kent, and J. Bibby. *Multivariate Analysis*. London; New York : Academic Press, 1979.

2. G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, 1992.

3. C. Stein. Estimation of a covariance matrix. *Rietz Lecture*, 1975.

4. M. Drton and M. D. Perlman. Multiple testing and error control in Gaussian graphical model selection. *Statist. Sci.*, **22**(3):430–449, 2007.

5. M. Drton and M. D. Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, **138**(4):1179 – 1200, 2008.

6. B. Naul and J. Taylor. Sparse Steinian covariance estimation. *Journal of Computational and Graphical Statistics*, **26**(2):355–366, 2017.

7. N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the LASSO. *The Annals of Statistics*, **34**(3):1436–1462, 2006.

8. B. Zhang and S. Horvath. A general framework for weighted gene coexpression network analysis. In *Statistical Applications in Genetics and Molecular Biology 4: Article 17*, 2005.

9. J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**(3):432–441, 2008.

10. D. M. Witten, J. H. Friedman, and N. Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, **20**(4):892–900, 2011.

11. R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, **13**:723–736, 2012.

12. C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack. BIG & QUIC: Sparse inverse covariance estimation for a million variables. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3165–3173. Curran Associates, Inc., 2013.

13. C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. QUIC: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, **15**:2911–2947, 2014.

14. W. Liu and X. Luo. Fast and adaptive sparse precision matrix estimation in high dimensions. *Journal of Multivariate Analysis*, **135**:153–162, 2015.

15. H. Liu and L. Wang. TIGER: A tuning-insensitive approach for optimally estimating Gaussian graphical models. *Electron. J. Statist.*, **11**(1):241–294, 2017.

16. D. I. Warton. Penalized normal likelihood and ridge regularization of correlation and covariance matrices. *Journal of the American Statistical Association*, **103**(481):340–349, 2008.

17. O. Ledoit and M. Wolf. Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management*, **30**(4):110–119, 2004.

18. O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, **88**:365–411, 2004.

19. O. Ledoit and M. Wolf. Spectrum etimation: A unified framework for covariance matrix estimation and PCA in large dimensions. *Journal of Multivariate Analysis*, **139**:360–384, 2015.

20. O. Ledoit and M. Wolf. Numerical implementation of the QuEST function. *ArXiv e-prints*, January 2016.

21. H. Liu, K. Roeder, and L. Wasserman. Stability approach to regularization selection (StARS) for high dimensional graphical models. In *Proceedings of the 23rd International*

Conference on Neural Information Processing Systems, NIPS'10, pages 1432–1440, USA, 2010. Curran Associates Inc.

22. O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, **9**(Mar):485–516, 2008.

23. R. Foygel and M. Drton. Extended Bayesian information criteria for Gaussian graphical models. *Advances in Neural Information Processing Systems*, **23**:604–612, 2010.

24. R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society - Series B*, **58**(1):267–288, 1996.

25. G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, **20**(1):359–392, 1998.

26. W. van Wieringen and C. Peeters. Ridge estimation of inverse covariance matrices from high-dimensional data. *Computational Statistics and Data Analysis*, **103**:284–303, 2016.

27. M. Kuismin, J. T. Kemppainen, and M. J. Sillanpää. Precision matrix estimation with ROPE. *Journal of Computational and Graphical Statistics*, in press.

28. M. Kuismin and M. J. Sillanpää. Use of Wishart prior and simple extensions for sparse precision matrix estimation. *PLOS ONE*, **11**(2):e0148171, 2016.

29. T. Cai, W. Liu, and X. Luo. A constrained l_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, **106**(494):594–607, 2011.

30. E. Candés and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Ann. Statist.*, **35**(6):2313–2351, 2007.

31. A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, **98**(4):791–806, 2011.

32. M. Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, **11**(Aug):2261–2286, 2010.

33. T. Zhao, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman. The huge package for high-dimensional undirected graph estimation in R. *Journal of Machine Learning Research*, **13**(Apr):1059–1062, 2012.

34. J. Bien and R. J. Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, **98**(4):807–820, 2011.

35. H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**(476):1418–1429, 2006.

36. X. Deng and K.-W. Tsui. Penalized covariance matrix estimation using a matrix-logarithm transformation. *Journal of Computational and Graphical Statistics*, **22**(2):494–512, 2013.

37. J.-H. Won, J. Lim, S.-J. Kim, and B. Rajaratnam. Condition-number-regularized covariance estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **75**(3):427–450, 2013.

38. Y. Fang, B. Wang, and Y. Feng. Tuning parameter selection in regularized estimations of large covariance matrices. *ArXiv e-prints*, 2013.

39. N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(4):417–473, 2010.

40. J. Whittaker. *Graphical Models*. John Wiley & Sons, West Sussex, England, 1990.

41. D. Edwards. *Introduction to Graphical Modelling*. Springer-Verlag, New York, 2 edition, 2000.

42. M. J. Ha and W. Sun. Partial correlation matrix estimation using ridge penalty followed by thresholding and re-estimation. *Biometrics*, **70**:765–773, 2014.

43. B. Efron. Large-scale simultaneous hypothesis testing: the choice of a null distribution. *Journal of American Statistical Association*, **99**(465):96–104, 2004.

44. R. Dezeure, P. Bühlmann, L. Meier, and N. Meinshausen. High-dimensional inference: Confidence intervals, p-values and R-software hdi. *Statist. Sci.*, **30**(4):533–558, 2015.

45. P. Bühlmann, M. Kalisch, and L. Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Applications*, **1**:255–278, 2014.

46. F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborov, and P. Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, **110**(52):20935–20940, 2013.

47. J. M. Ranola, P. Langfelder, K. Lange, and S. Horvath. Cluster and propensity based approximation of a network. *BMC Systems Biology*, **7**(1):21, 2013.

48. P. Langfelder, P. S. Mischel, and S. Horvath. When is hub gene selection better than standard meta-analysis? *PLOS ONE*, **8**(4):1–16, 2013.

49. T. Äijö and R. Bonneau. Biophysically motivated regulatory network inference: Progress and prospects. *Human Heredity*, **81**(2):62–77, 2016.

50. P. J. Bickel and E. Levina. Covariance regularization by thresholding. *Ann. Statist.*, **36**(6):2577–2604, 2008.

51. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, **95**(25):14863–14868, 1998.

52. P. Langfelder and S. Horvath. WGCNA: An R package for weighted correlation network analysis. *BMC Bioinformatics*, **9**(1):559, 2008.

53. A. Ghazalpour, S. Doss, B. Zhang, S. Wang, C. Plaisier, R. Castellanos, A. Brozell, E. E. Schadt, T. A. Drake, A. J. Lusis, and S. Horvath. Integrating genetic and network analysis to characterize genes related to mouse weight. *PLOS Genetics*, **2**(8):1–11, 2006.

54. Y. Li and S. A. Jackson. Gene network reconstruction by integration of prior biological knowledge. *G3: Genes—Genomes—Genetics*, **5**(6):1075–1079, 2015.

55. A. Wille, P. Zimmermann, E. Vranová, A. Fürholz, O. Laule, S. Bleuler, L. Hennig, A. Prelić, P. von Rohr, L. Thiele, E. Zitzler, W. Gruissem, and P. Bühlmann. Sparse

graphical Gaussian modeling of the isoprenoid gene network in *Arabidopsis thaliana*. *Genome Biology*, **5**(11):R92, 2004.

56. T. Shimamura, S. Imoto, R. Yamaguchi, and S. Miyano. Weighted lasso in graphical Gaussian modeling for large gene network estimation based on microarray data. *Genome Informatics*, **19**:142–153, 2007.

57. S. Jokipii-Lukkari, D. Sundell, O. Nilsson, T. R. Hvidsten, N. R. Street, and H. Tuominen. Norwood: a gene expression resource for evo-devo studies of conifer wood development. *New Phytologist*, in press.

58. N. Krämer, J. Schäfer, and A.-L. Boulesteix. Regularized estimation of large-scale gene association networks using graphical Gaussian models. *BMC Bioinformatics*, **10**(1):384, 2009.

59. J. Ruan, A. K. Dean, and W. Zhang. A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC Systems Biology*, **4**(1):8, 2010.

60. Z. S. Khondker, H. Zhu, H. Chu, W. Lin, and J. G. Ibrahim. The Bayesian covariance lasso. *Statistics and Its Interface*, **6**:243–259, 2013.

61. H. Wang. Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, **7**(4):867–886, 2012.

62. A. Bhadra and B. K. Mallick. Joint high-dimensional Bayesian variable and covariance selection with an application to eQTL analysis. *Biometrics*, **69**(2):447–457, 2013.

63. T. Kubokawa and M. S. Srivastava. Estimation of the precision matrix of a singular Wishart distribution and its application in high-dimensional data. *Journal of Multivariate Analysis*, **99**(9):1906–1928, 2008.

64. M. Bouriga and O. Féron. Estimation of covariance matrices based on hierarchical inverse-Wishart priors. *Journal of Statistical Planning and Inference*, **143**(4):795–808, 2013.

65. A. Huang and M. P. Wand. Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, **8**(2):439–452, 2013.

66. A. Mohammadi and E. C. Wit. BDgraph: An R Package for Bayesian Structure Learning in Graphical Models. *ArXiv e-prints*, January 2015.

67. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics Springer, Berlin, 2009.

68. T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC press, 2015.

69. M. Pourahmadi. *High-Dimensional Covariance Estimation*. John Wiley & Sons, New York, 2013.

70. T. Tong, C. Wang, and Y. Wang. Estimation of variances and covariances for high-dimensional data: a selective review. *Wiley Interdisciplinary Reviews: Computational Statistics*, **6**(4):255–264, 2014.

71. J. Fan, Y. Liao, and H. Liu. An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal*, **19**(1):C1–C32, 2016.