

Supplementary Material to:  
Ridge estimation of the VAR(1) model and its time series  
chain graph from multivariate time-course omics data

**Viktorian Miok<sup>1,2</sup>, Saskia M. Wilting<sup>2</sup>, Wessel N. van Wieringen<sup>1,3,\*</sup>**

<sup>1</sup> Department of Epidemiology and Biostatistics, VU University Medical Center

P.O. Box 7057, 1007 MB Amsterdam, The Netherlands

<sup>2</sup> Department of Pathology, VU University Medical Center

P.O. Box 7057, 1007, MB, Amsterdam, The Netherlands

<sup>3</sup> Department of Mathematics, VU University Amsterdam

De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

---

\*Correspondence to: w.vanwieringen@vumc.nl.

## SM I: Pseudo-code of ridge ML estimation

*Box 1:* Ridge ML estimation of the parameters of the VAR(1) model.

**Initiate**

Obtain an initial estimate  $\hat{\mathbf{A}}^{(0)}(\lambda_a)$  of  $\mathbf{A}$  from the ridge OLS estimator.

**Iterate**

For  $k = 1, \dots, k_{\max}$  iterate the following steps:

- 1) Use the latest estimate of  $\mathbf{A}$ ,  $\hat{\mathbf{A}}^{(k-1)}(\lambda_a)$ , to calculate the sample error covariance matrix  $\mathbf{S}_{\varepsilon}^{(k)}$ . Given  $\mathbf{S}_{\varepsilon}^{(k)}$ , the ridge ML precision estimator yields the updated estimate of  $\mathbf{\Omega}_{\varepsilon}$ ,  $\hat{\mathbf{\Omega}}_{\varepsilon}^{(k)}(\lambda_{\omega})$ .
- 2) Get  $\hat{\mathbf{A}}^{(k)}(\lambda_a)$  from the ridge ML estimator of  $\mathbf{A}$  with  $\hat{\mathbf{\Omega}}_{\varepsilon}^{(k)}(\lambda_{\omega})$  substituted for  $\mathbf{\Omega}_{\varepsilon}$ .

**Terminate**

Iterate the previous step, until convergence.

## SM II: Comparison of computation time

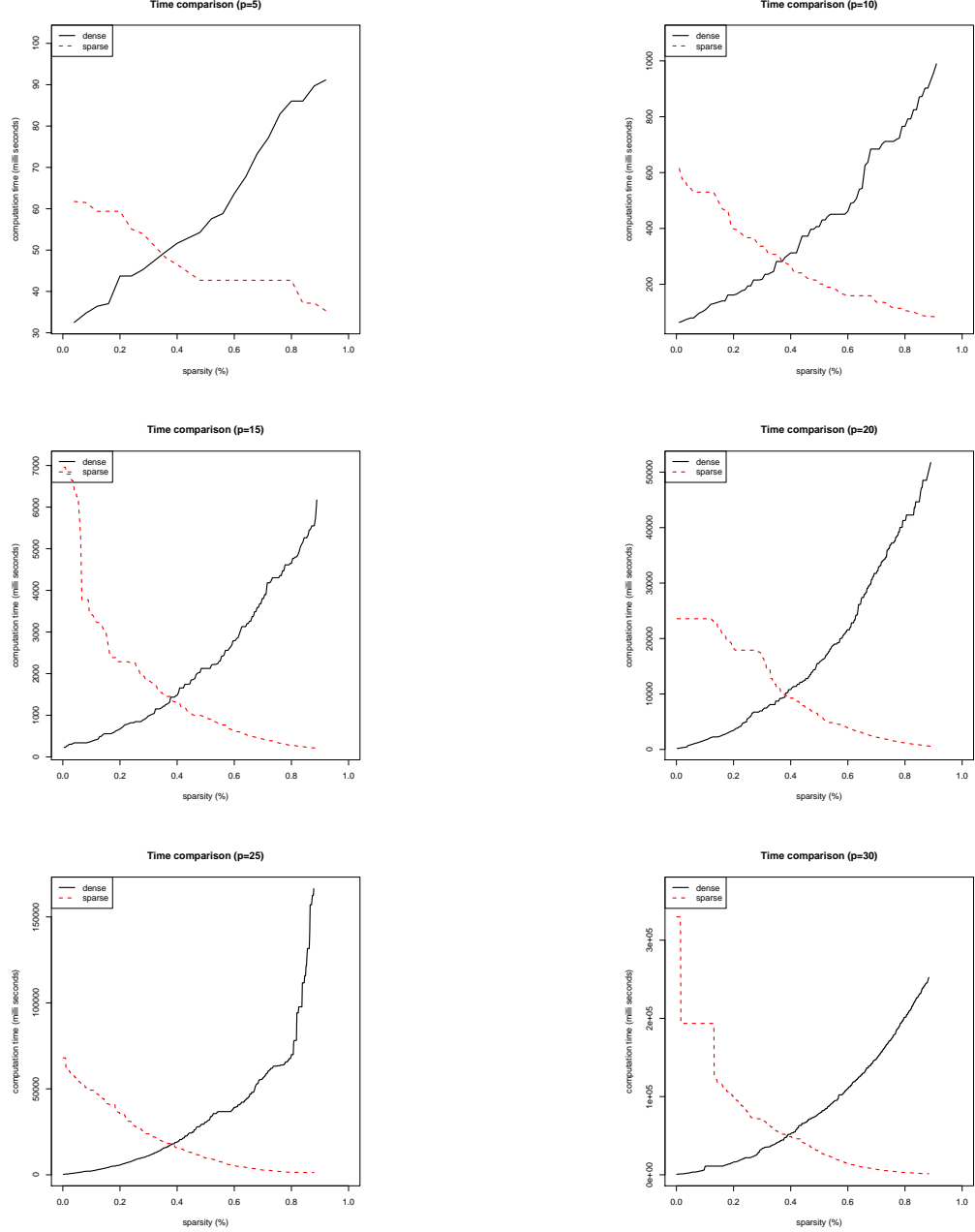
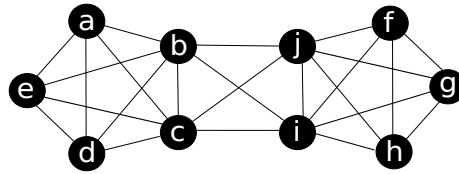


Figure 1: The computational costs of evaluating the estimator of  $\mathbf{A}$  with known support for two plotted against the sparsity percentage of the support. Two different implementations are evaluated: for a dense (few zeros; black, solid line) and sparse (many zeros; red, dashed line) support. Computation times were evaluated using the `microbenchmark`-package (?).

### SM III: Decomposable graph illustration

Figure ?? shows an example of a decomposable graph and its decomposition. Clearly, every path between the two cliques runs via the separator. Furthermore, both cliques and separator are complete subgraphs. Hence, they form a decomposition of the graph. Figure ?? containing two additional examples of decomposable graphs: the tree and the chain graph.

A decomposable graph



Its decomposition

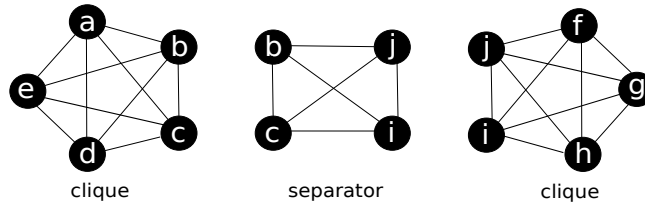


Figure 2: The top row shows a decomposable graph. The bottom row displays its decomposition in terms of the two cliques, comprising vertices  $\{a, b, c, d, e\}$  and  $\{f, g, h, i, j\}$  and the edges connecting them, and a separator with vertices  $\{b, c, i, j\}$  and the edges connecting them.

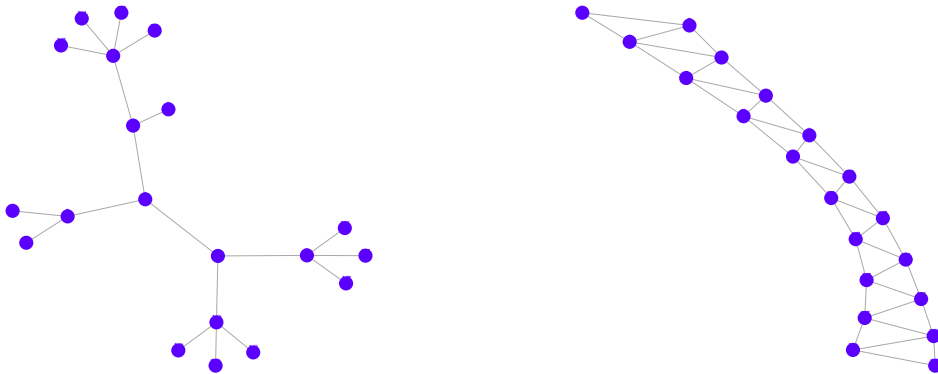


Figure 3: Two examples of decomposable graphs: the tree and a chain graph.

## SM IV: initial estimate of $\Omega$ with chordal support

Under the assumption of a decomposable conditional independence graph  $\mathcal{G}$ , the ridge ML estimator of  $\Omega_\varepsilon$  can be expressed in limiting cases ( $\lambda_\omega = 0$ ,  $\lambda_\omega = \infty$  and  $\mathcal{S} = \emptyset$ ) as a combination of the ridge ML estimators of the cliques and separators that decompose  $\mathcal{G}$ . Fully analogous to unpenalized case specified by Proposition 5.9 of (?) but with the unpenalized marginal ML covariance estimates of cliques and separators replaced by their penalized counterparts.

To derive the ridge ML estimator for the aforementioned limiting cases, note by direct observation that the decomposability of the support of  $\Omega_\varepsilon$  implies:

$$\begin{aligned} \text{tr}[(\Omega_\varepsilon - \Omega_0)^\top (\Omega_\varepsilon - \Omega_0)] &= \sum_{C \in \mathcal{C}} \text{tr}\{[(\Omega_\varepsilon)_{C,C} - (\Omega_0)_{C,C}]^\top [(\Omega_\varepsilon)_{C,C} - (\Omega_0)_{C,C}]\} \\ &\quad - \sum_{S \in \mathcal{S}} \text{tr}\{[(\Omega_\varepsilon)_{S,S} - (\Omega_0)_{S,S}]^\top [(\Omega_\varepsilon)_{S,S} - (\Omega_0)_{S,S}]\}, \end{aligned} \quad (1)$$

in which  $\Omega_0$  is assumed to share its support with  $\Omega_\varepsilon$ . For square, symmetric  $\mathbf{S}_\varepsilon \succeq$ ,  $\Omega_\varepsilon \succ 0$ , and  $\Omega_0 \succ 0$  of arbitrary but equal dimensions define

$$\mathcal{L}_\omega^{\text{pen}}(\mathbf{S}_\varepsilon, \Omega_\varepsilon, \Omega_0, \lambda_\omega) = \frac{1}{2}n(\mathcal{T} - 1)\{\log |\Omega_\varepsilon| - \text{tr}(\mathbf{S}\Omega_\varepsilon) - \lambda_\omega \text{tr}[(\Omega_\varepsilon - \Omega_0)^\top (\Omega_\varepsilon - \Omega_0)]\}.$$

Using the reformulated ridge penalty (??) together with iterative application of Lemma 5.5. of (?), which decomposes the unpenalized log-likelihood in a linear combination of those of the cliques and separators, the loss function for  $\Omega_\varepsilon$  can then be rewritten to:

$$\sum_{C \in \mathcal{C}} \mathcal{L}_\omega^{\text{pen}}[(\mathbf{S}_\varepsilon)_{C,C}, (\Omega_\varepsilon)_{C,C}, (\Omega_0)_{C,C}, \lambda_\omega] - \sum_{S \in \mathcal{S}} \nu(S) \mathcal{L}_\omega^{\text{pen}}[(\mathbf{S}_\varepsilon)_{S,S}, (\Omega_\varepsilon)_{S,S}, (\Omega_0)_{S,S}, \lambda_\omega], \quad (2)$$

where  $\nu(S)$  is the number of times separator  $S$  forms the intersection of two neighboring cliques.

Recall from the main text that, when  $\mathcal{G}$  decomposes into cliques  $C_1$ ,  $C_2$  and  $S$ , the proposed ridge ML estimator is:

$$\widehat{\Omega}_\varepsilon(\lambda_\omega) = \begin{pmatrix} [\widehat{\Omega}_\varepsilon^{(C_1)}(\lambda_\omega)]_{C_1 \setminus S, C_1 \setminus S} & [\widehat{\Omega}_\varepsilon^{(C_1)}(\lambda_\omega)]_{C_1 \setminus S, S} & \mathbf{0}_{|C_1 \setminus S| \times |C_2 \setminus S|} \\ [\widehat{\Omega}_\varepsilon^{(C_1)}(\lambda_\omega)]_{S, C_1 \setminus S} & [\widehat{\Omega}_\varepsilon^{(C_1)}(\lambda_\omega)]_{S, S} + [\widehat{\Omega}_\varepsilon^{(C_2)}(\lambda_\omega)]_{S, S} - \widehat{\Omega}_\varepsilon^{(S)}(\lambda_\omega) & [\widehat{\Omega}_\varepsilon^{(C_2)}(\lambda_\omega)]_{S, C_2 \setminus S} \\ \mathbf{0}_{|C_2 \setminus S| \times |C_1 \setminus S|} & [\widehat{\Omega}_\varepsilon^{(C_2)}(\lambda_\omega)]_{C_2 \setminus S, S} & [\widehat{\Omega}_\varepsilon^{(C_2)}(\lambda_\omega)]_{C_2 \setminus S, C_2 \setminus S} \end{pmatrix},$$

where  $\widehat{\Omega}_\varepsilon^{(C_1)}(\lambda_\omega)$ ,  $\widehat{\Omega}_\varepsilon^{(C_2)}(\lambda_\omega)$  and  $\widehat{\Omega}_\varepsilon^{(S)}(\lambda_\omega)$  are the marginal ridge ML covariance estimators for cliques  $C_1$ ,  $C_2$  and separator  $S$ . Note that each of these marginal estimator satisfy the marginal estimating equation, e.g.:

$$[\widehat{\Omega}_\varepsilon^{(C_1)}(\lambda_\omega)]^{-1} - \lambda_\omega \widehat{\Omega}_\varepsilon^{(C_1)}(\lambda_\omega) = (\mathbf{S}_\varepsilon)_{C_1, C_1} - \lambda_\omega (\Omega_0)_{C_1, C_1},$$

confer (?) for details.

To show that this estimator maximizes (in the three limiting cases) the ridge penalized log-likelihood under the assumption of a decomposable conditional independence graph, we study this penalized log-likelihood in the vicinity of the proposed estimator. Let a  $\Delta$  be a  $p \times p$  dimensional matrix and  $\gamma$  a constant close to but unequal to zero. Consider a single summand of the penalized log-likelihood (??), say, that of the first clique. It may then, after noting that  $\log(\mathbf{A} + \varepsilon \mathbf{B}) = \log(\mathbf{A}) + \varepsilon \mathbf{A}^{-1} \mathbf{B} + \mathcal{O}(\varepsilon^2)$  and  $\log(|\mathbf{A}|) = \text{tr}[\log(\mathbf{A})]$ , be approximated by:

$$\mathcal{L}_\omega^{\text{pen}}\{(\mathbf{S}_\varepsilon)_{C_1, C_1}, [\widehat{\Omega}_\varepsilon(\lambda_\omega)]_{C_1, C_1} + \gamma(\Delta)_{C_1, C_1}, (\Omega_0)_{C_1, C_1}, \lambda_\omega\}$$

$$\begin{aligned}
\approx & \log \{ |[\widehat{\boldsymbol{\Omega}}_\varepsilon(\lambda_\omega)]_{C_1, C_1}| \} - \text{tr} \{ (\mathbf{S}_\varepsilon)_{C_1, C_1} [\widehat{\boldsymbol{\Omega}}_\varepsilon(\lambda_\omega)]_{C_1, C_1} \} \\
& - \frac{1}{2} \lambda_\omega \text{tr} \left( \{ [\widehat{\boldsymbol{\Omega}}_\varepsilon(\lambda_\omega)]_{C_1, C_1} - (\boldsymbol{\Omega}_0)_{C_1, C_1} \}^\top \{ [\widehat{\boldsymbol{\Omega}}_\varepsilon(\lambda_\omega)]_{C_1, C_1} - (\boldsymbol{\Omega}_0)_{C_1, C_1} \} \right) \\
& + \gamma \text{tr} \left[ \left( \{ [\widehat{\boldsymbol{\Omega}}_\varepsilon(\lambda_\omega)]_{C_1, C_1} \}^{-1} - \{ [\widehat{\boldsymbol{\Omega}}_\varepsilon(\lambda_\omega)]^{-1} \}_{C_1, C_1} \right) (\boldsymbol{\Delta})_{C_1, C_1} \right] \\
& - \frac{1}{2} \gamma^2 \lambda_\omega \text{tr} \{ [(\boldsymbol{\Delta})_{C_1, C_1}]^\top (\boldsymbol{\Delta})_{C_1, C_1} \},
\end{aligned}$$

in which we also used the estimating equation.

Only the latter two summands of this approximation of the penalized log-likelihood of the clique involve  $\gamma$ . Now consider the three boundary cases separately:

- When  $\lambda_\omega = 0$  the summands in the approximation stemming from the penalty vanish and the claim is warranted by Lemma 5.5 of ?).
- When  $\mathcal{S} = \emptyset$ , the precision matrix is block diagonal and the blocks can be estimated marginally. This implies  $[(\boldsymbol{\Omega}_\varepsilon)_{C_1, C_1}]^{-1} = [(\boldsymbol{\Omega}_\varepsilon)^{-1}]_{C_1, C_1}$  and ensures that the first summand involving  $\gamma$  vanishes. As the second summand can only contribute negatively to the log-likelihood for any  $\gamma \neq 0$ , the proposed estimator maximizes the log-likelihood.
- When  $\lambda_\omega$  tends to infinity, the second summand involving  $\gamma$  dominates the other. For these dominant summands:

$$\text{tr}(\boldsymbol{\Delta}^\top \boldsymbol{\Delta}) = \sum_{C \in \mathcal{C}} \text{tr} \{ [(\boldsymbol{\Delta})_{C, C}]^\top (\boldsymbol{\Delta})_{C, C} \} - \sum_{S \in \mathcal{S}} \text{tr} \{ [(\boldsymbol{\Delta})_{S, S}]^\top (\boldsymbol{\Delta})_{S, S} \}.$$

The total contribution of these terms to the log-likelihood is thus negative for any choice of  $\gamma$  other than zero.

Hence, the proposed estimator indeed maximizes the penalized log-likelihood in the three limiting cases.

## SM V: Path analysis illustration

The path analysis of autocovariance is illustrated by means of a toy example. Consider a VAR(1) model parametrized by:

$$\boldsymbol{\nu} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 1/2 & 1/2 & 1/2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_\varepsilon = \begin{pmatrix} 1 & 1/5 & 1/5 \\ 1/5 & 1 & 0 \\ 1/5 & 0 & 1 \end{pmatrix}^{-1}.$$

The top row of Figure ?? displays the corresponding time series chain graph.

The autocovariance between  $Y_{1,t}$  and  $Y_{1,t+1}$  (dropping the sample index for notational clarity) is:  $\text{Cov}(Y_{1,t}, Y_{1,t+1}) = (\boldsymbol{\Sigma}_\varepsilon \mathbf{A}^\top)_{1,1} = 0.326087$ . From the time series chain graph it is immediate that  $Y_{1,t}$  and  $Y_{1,t+1}$  are connected by three paths (visualized by the bottom row of Figure ??): *i*)  $Y_{1,t} \rightarrow Y_{1,t+1}$ , *ii*)  $Y_{1,t} \rightarrow Y_{2,t} \rightarrow Y_{1,t+1}$ , and *iii*)  $Y_{1,t} \rightarrow Y_{3,t} \rightarrow Y_{1,t+1}$ . The covariance between  $Y_{1,t}$  and  $Y_{1,t+1}$  can now be decomposed into the contributions of each of these paths. Using the decomposition of the autocovariance in terms of the paths' contributions from the main text one obtains: 0.5434783 (for  $Y_{1,t} \rightarrow Y_{1,t+1}$ ),  $-0.1086957$  (for  $Y_{1,t} \rightarrow Y_{2,t} \rightarrow Y_{1,t+1}$ ) and  $-0.1086957$  (for  $Y_{1,t} \rightarrow Y_{3,t} \rightarrow Y_{1,t+1}$ ). These contributions indeed sum to 0.326087, which equals  $\text{Cov}(Y_{1,t}, Y_{1,t+1})$ .

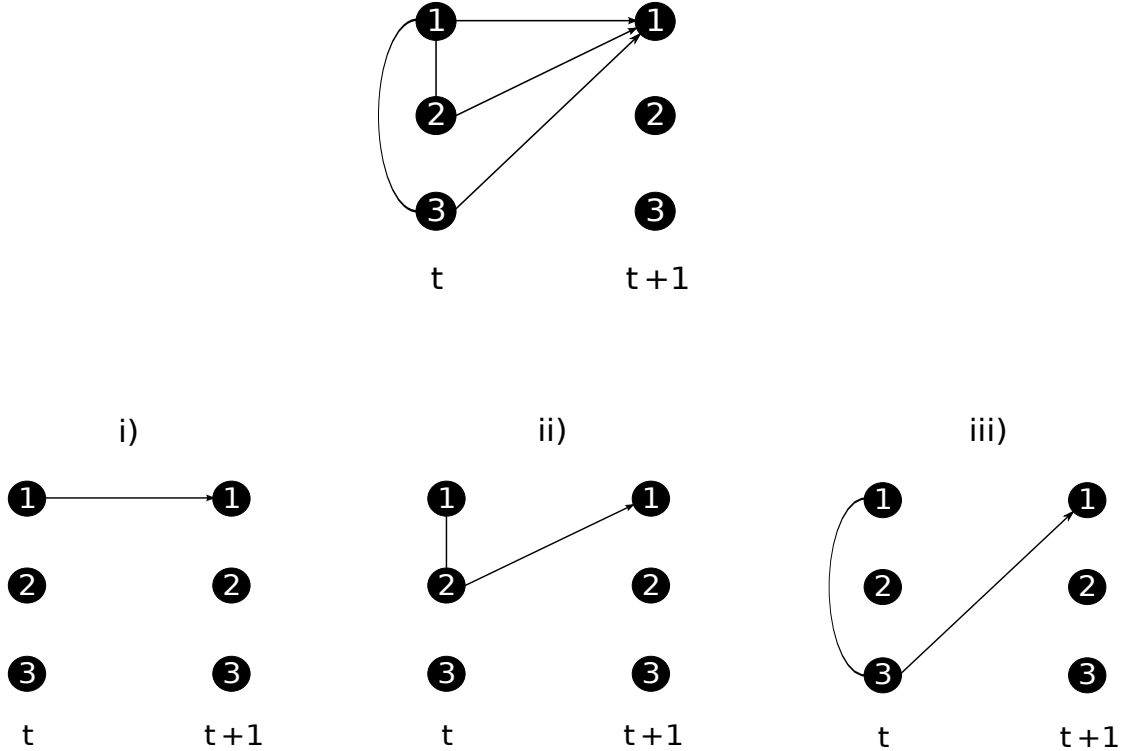


Figure 4: The top row shows the time series chain graph of the toy model. The bottom row displays the three paths in the time series chain graph connecting the first variate between to consecutive time points.

## SM VI: Design of time-course studies in GEO

?) observed that more than 80% of time-course gene expression microarray data sets have 8 or fewer number of time points. This reference being somewhat out of date, we conducted our own survey using the GEO data base ?). Using the search terms “time course”, “cancer”, “human”, “cell line”, “gene expression”, “microarray”, 85 studies were selected (at June 23, 2015). From the resulting list we excluded studies involving hematological tumors and data set GSE762 with a rather intricate experimental design (among other a time component). Finally, 63 studies satisfy these criteria, involving cell lines of various cancer types (breast, colon, prostate, ovarian, kidney and lung) and were mainly conducted using Affymetrix gene expression arrays (confer Table ??). Figure ?? shows histograms of the number of time-points  $\mathcal{T}$  and individuals  $n$  (left panels) and the scatterplot of these study characteristics. From the provided table and histograms we conclude that a reasonable optimistic and simultaneously empirically maximum number of time-points and individuals are 10 and 5, respectively.

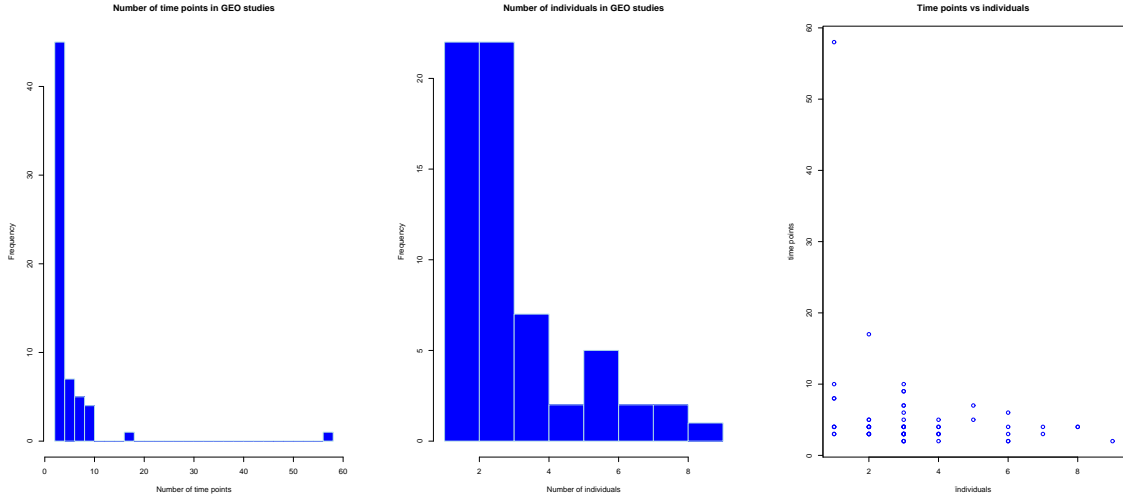


Figure 5: Number of time-points and observations in microarray time-course gene expression studies from GEO database. Left panel presents the histogram of number of time-points, while the middle panel shows the histogram of the number of observations (cell lines) in GEO studies. The right panel plots the number of time points against the number of individuals.



data set ID	# time points	# individuals	# treatments	cancer type	organism	platform
GSE7868	3	3	1	prostate	human	Affymetrix
GDS2626	4	7	2	breast	human	Affymetrix
GSE29917	7	5	3	breast	human	Agilent
GSE13009	17	2	2	breast	human	Affymetrix
GDS4367	5	3	1	colon	mouse	Affymetrix
GDS4121	3	1	1	prostate	human	Affymetrix
GDS4110	9	3	1	prostate	human	Affymetrix
GDS4063	2	9	3	breast	human	Affymetrix
GDS4052	2	6	3	breast	human	Affymetrix
GDS3910	3	2	3	breast	human	Affymetrix
GDS3887	4	4	2	bladder	mouse	Affymetrix
GDS3872	3	3	2	ovarian	human	Affymetrix
GDS3761	3	4	2	prostate	human	Affymetrix
GDS3710	9	3	1	epithelial	human	Affymetrix
GDS3517	4	3	4	melanoma	human	Affymetrix
GDS3484	2	3	2	breast	human	Affymetrix
GDS3792	3	2	2	ovarian	human	Affymetrix
GDS3358	5	2	2	prostate	human	Affymetrix
GDS3319	3	1	3	thyroid	human	Affymetrix
GDS3285	4	3	1	breast	human	Affymetrix
GDS3283	2	6	2	breast	human	Affymetrix
GDS3116	58	1	2	breast	human	Affymetrix
GDS3111	3	3	2	prostate	human	Affymetrix
GDS2975	4	2	1	ovarian	human	Affymetrix
GDS2810	3	2	2	breast	human	Affymetrix
GDS2604	6	6		lung	human	Affymetrix
GDS2323	3	3	1	breast	human	Affymetrix
GDS2097	4	1	2	breast	human	Affymetrix
GDS2096	4	1	2	breast	human	Affymetrix
GDS1942	8	1	2	colon	human	Affymetrix
GDS1736	2	4	2	prostate	human	Affymetrix
GDS1627	3	2		breast	human	oligo
GDS848	3	7	3	breast	human	oligo
GDS725	10	1	1	prostate	human	Affymetrix
GDS719	4	2	3	prostate	human	Affymetrix
GDS528	3	3	1	colon	human	Affymetrix
GSE41828	5	5	2	prostate	human	Affymetrix
GSE41680	4	1	2	melanoma	human	Affymetrix
GSE31128	4	3	4	breast	human	Illumina
GSE28305	4	2	1	breast	human	Affymetrix
GDS4853	4	3	2	stomach	human	Affymetrix
GSE26834	2	3	6	breast	human	Affymetrix
GSE23616	3	3	2	ovarian	human	Affymetrix
GSE20361	8	1	1	breast	human	Affymetrix
GSE23845	4	4	1	bladder	human	Affymetrix
GSE15947	3	4	3	prostate	human	Affymetrix
GSE18494	4	3	3	breast	human	Affymetrix
GSE17708	10	3	2	lung	human	Affymetrix
GSE11428	3	3	6	prostate	human	Affymetrix
GSE16424	4	6	2	ovarian	human	Affymetrix
GSE13525	3	4	2	ovarian	human	Affymetrix
GSE13059	4	8	1	colon	human	Affymetrix
GSE8772	3	2	1	melanoma	human	Affymetrix
GSE8702	7	3	1	prostate	human	Affymetrix
GSE9936	2	3	16	breast	human	Affymetrix
GSE9826	6	3	3	ovarian	human	Affymetrix
GSE8057	5	4	8	ovarian	human	Affymetrix
GSE6013	5	2	3	lung	human	Affymetrix
GSE6930	3	6	4	ewing sarcoma	human	Affymetrix
GSE6653	4	2	1	ovarian	human	Affymetrix
GSE5345	7	3	2	prostate	human	IQUSP
GDS2034	4	2	1	ovarian	human	Affymetrix
GSE4917	4	8	3	breast	human	Affymetrix

Table 1: GEO data sets with a time-course design.

## SM VII: Comparison by simulation

The proposed ridge estimator is compared by means of simulation to (as far as we are aware) the only other method that estimates the VAR(1) model and its associated time series chain graph from high-dimensional data: SparseTSCGM proposed by ?). SparseTSCGM too estimates parameters of the VAR(1) model in penalized fashion, but employs the smoothly clipped absolute deviation (SCAD, ?) penalty on the autoregressive coefficient matrix  $\mathbf{A}$  and precision matrix  $\mathbf{\Omega}_\varepsilon$ . As the estimation methods estimate the same VAR(1) model and only differ in the employed penalties, they can readily be compared in terms of loss of the estimates and selection of the edges of the time series chain graph.

The ridge and ‘SCAD’ estimators of the VAR(1) model are compared under various choices of the model parameters  $\mathbf{A}$  and  $\mathbf{\Omega}_\varepsilon$ , while the number of variates  $p$ , time points  $\mathcal{T}$  and the number of samples  $n$  are varied in accordance with a factorial design. We first describe the levels of the factors of the factorial design. The number of variates equals either  $p = 25$  or  $p = 50$ <sup>1</sup>, representing the size of non-trivial but well-defined pathways. Simultaneously, we set  $n \in \{5, 15\}$  and  $\mathcal{T} \in \{10, 20\}$ . We stress that the particular case of  $(n = 5, \mathcal{T} = 10)$  is the most challenging but also most relevant as its design is closest to that which is most prevalent in practice. ?) observed that more than 80% of time-course gene expression microarray data sets have 8 or fewer number of time points. This is confirmed by a more recent survey of our own (confer SM III).

We now outline parameter choices of the various VAR(1) models employed in the simulation. Autoregression matrices  $\mathbf{A}$  with a hub, cluster, random (each with two sparsity levels: approximately 5% and 25% nonzero elements), and full network structure choices are used. These matrices  $\mathbf{A}$  are used in combination with a banded and full precision matrix  $\mathbf{\Omega}_\varepsilon$  and two data-driven variants. We first detail the various choices of  $\mathbf{\Omega}_\varepsilon$ :

- **Full  $\mathbf{\Omega}_\varepsilon$ :** In the precision matrix  $\mathbf{\Omega}_\varepsilon$  existing contemporaneous conditional independencies is represented with  $\rho = 0.5$ .
- **Banded  $\mathbf{\Omega}_\varepsilon$ :** A banded precision matrix is generated with  $(\mathbf{\Omega}_\varepsilon)_{j_1, j_2} = \rho^{|j_1 - j_2|}$  with  $\rho = 0.5$  for  $|j_1 - j_2| < 3$  and  $(\mathbf{\Omega}_\varepsilon)_{j_1, j_2} = 0$  otherwise.
- **Full but data-driven  $\mathbf{\Omega}_\varepsilon$ :** The precision matrix is obtained from a VAR(1) model fitted to data from a time-course gene experiment experiment. The data of the application detailed in the main document are used. The data are limited to genes mapping to the p53 signaling pathway as defined by KEGG (?). The VAR(1) model is fitted to these pathway data by means of the proposed ridge estimation procedure using penalty parameter values  $\lambda_a = 0.3, \lambda_\omega = 0.1$ . The thus estimated  $\mathbf{\Omega}_\varepsilon$  is used in the simulation. For the various choice of  $p$ , the first  $p$  genes from the pathway are used in the comparison.

For the regression coefficient matrix  $\mathbf{A}$  the following variants are employed:

- **Hub  $\mathbf{A}$ :** All elements of  $\mathbf{A}$  are set to zero, except for  $(\mathbf{A})_{j_1, j_2} = 0.3$  for  $j_1 = \{d, 2d, 3d, \dots\}$  with  $j_1 > j_2$  and  $d = 10$  and  $d = 2$  for the less sparse case). Hence, only a few variates affect (in varying degree) the temporal variation of the others.
- **Cluster  $\mathbf{A}$ :** Eight (most sparse) or two (less sparse) equally-sized lower-triangular blocks filled with the value 0.3 are aligned along the diagonal. The remaining elements of  $\mathbf{A}$  are zero.
- **Random  $\mathbf{A}$ :** A random network is generated, selecting 10% or 50% of the total possible edges, after which the lower triangle is retained, i.e. setting  $(\mathbf{A})_{j_1, j_2} = 0$  if  $j_1 < j_2$ . Elements of  $\mathbf{A}$  are set equal to 0.3 if the corresponding element in the adjacency matrix of the random network is nonzero.
- **Full data-driven  $\mathbf{A}$ :** The same data set as for the generation of the data-driven  $\mathbf{\Omega}_\varepsilon$  is used to

---

<sup>1</sup>Initially, we also included  $p = 100$ , but the estimation with the SCAD procedure fails to converge for  $n = 5$  and  $T = 10$ , as is often the case for  $p = 100$  with larger  $n$  and  $\mathcal{T}$ .

estimate  $\mathbf{A}$ . Using these data  $\mathbf{A}$  is estimated by the ridge procedure with penalty parameters  $\lambda_a = 0.3$  and  $\lambda_\omega = 0.1$ . The resulting  $\mathbf{A}$  is used in the simulation.

- Sparse data-driven  $\mathbf{A}$  is obtained by sparsifying (as outlined in Section 4.1 of the main text) the full data-driven  $\mathbf{A}$ .

Note that the majority of choices for  $\mathbf{A}$  and  $\mathbf{\Omega}_\varepsilon$  is sparse. This in principle favors the ‘SCAD’ estimation procedure SparseTSCGM.

For (combination of) these choices for the regression parameter matrix  $\mathbf{A}$  and precision matrix  $\mathbf{\Omega}_\varepsilon$ , data are simulated in accordance with the VAR(1) model. Data for the first time point  $t = 1$  are drawn from  $\mathcal{N}(\mathbf{0}_{p \times 1}, \mathbf{\Omega}_\varepsilon^{-1})$  and for the following time points  $t = 2, \dots, T$  data are sampled from  $\mathcal{N}(\mathbf{A}\mathbf{Y}_{*,i,t-1}, \mathbf{\Sigma}_\varepsilon)$ . In total for each employed combination of the model parameters and design parameters  $p, n, \mathcal{T}$ , fifty time series data sets are generated.

For each simulated data set, the VAR(1) model is fitted using both the proposed ridge procedure and the ‘SCAD’-based method SparseTSCGM. The parameters estimates obtained from both methods are chosen in an identical fashion: using maximization of the leave K-fold-out cross-validated log-likelihood. Instead of removing a single element from the set of design points (as discussed in Section 3.4 of the main document), an individual (cell line) with all its time points is removed at each fold. This is due to the fact that the SparseTSCGM cannot deal with an unbalanced design. Formally, from the set of design points  $\mathcal{D} = \{(t, i) : t = 1, \dots, \mathcal{T}, i = 1, \dots, n\}$  we remove a single sample  $\mathcal{D} = \{(i) : i = 1, \dots, n\}$  at the time. Resulting parameter estimates of  $\mathbf{A}$  and  $\mathbf{\Omega}_\varepsilon$  obtained by SparseTSCGM are sparse, whereas their ridge counterparts are not. This is irrelevant for the loss comparison (for which both a sparse and full matrix can be used). Only comparing the methods with respect to the ability to reconstruct the time series chain graph, the ridge estimates undergo post-estimation sparsification (discussed below).

The ridge and SCAD estimators of  $\hat{\mathbf{A}}(\lambda_a)$  and  $\hat{\mathbf{\Omega}}_\varepsilon(\lambda_a, \lambda_\omega)$  are evaluated in terms of their Frobenius loss and their ability to select edges of time series chain graph. The Frobenius loss for (e.g.) a precision matrix estimate is:

$$\|\hat{\mathbf{\Omega}}_\varepsilon(\lambda_a, \lambda_\omega) - \mathbf{\Omega}_\varepsilon\|_F^2 = \sum_{j_1, j_2} \left\{ [\hat{\mathbf{\Omega}}_\varepsilon(\lambda_a, \lambda_\omega)]_{j_1, j_2} - (\mathbf{\Omega}_\varepsilon)_{j_1, j_2} \right\}^2.$$

The Frobenius loss for the estimate of  $\mathbf{A}$  is defined similarly.

The edge selection ability of the ridge and SCAD estimation methods are compared by means of sensitivity and specificity. When the ridge estimation is augmented with the post-estimation empirical Bayes procedure, the two methods may yield differing number of edges. This may hamper the comparison of their sensitivity and specificity. To facilitate a better comparison, also adhering to the *ceteris paribus* principle, of the two methods in this respect, a simple but different post-estimation edge selection procedure is applied to the ridge estimate. It comprises of selecting the same number of edges as the SCAD method, thus favouring the latter. Having fixed the number of to-be-selected edges, they are selected on the basis of the (absolute) size of the statistic derived from the elements of  $\mathbf{A}$  (as pointed out in Section 4.1 the main document) and (standardized)  $\hat{\mathbf{\Omega}}_\varepsilon(\lambda_a, \lambda_\omega)$ . This thus means that in each data set for the number of edges selected by the SCAD estimator, we select the same number of largest elements of from the estimate of  $\mathbf{A}$ .

We first discuss the Frobenius loss comparison. Figures ?? and ?? show the Frobenius loss (as boxplots) of the ridge and SCAD estimates of  $\mathbf{A}$  (upper panel) with roughly 5% and 25% nonzero elements, respectively, and  $\mathbf{\Omega}_\varepsilon$  (lower panel) from fifty data sets generated with  $p = 25, T = 10, n = 5$  (the most relevant case empirically), and the various combinations of  $\mathbf{A}$  and  $\mathbf{\Omega}_\varepsilon$ . The panels of Figures ?? and ?? reveal that the Frobenius loss of the SCAD estimates of both  $\hat{\mathbf{A}}(\lambda_a)$  and  $\hat{\mathbf{\Omega}}_\varepsilon(\lambda_\omega)$  exceeds that of its ridge counterpart. This observation is consistent over the data sets generated from different choices of the regression coefficient (with all levels of sparsity) and precision matrices. This is confirmed when the VAR(1) model is increased to include  $p = 50$  variates. The

difference in Frobenius loss between the ridge and SCAD estimators grows substantially in favour of the former (confer Figures ?? and ??). When the number of time points  $\mathcal{T}$  is increased, the ridge procedure outperforms its SCAD counterpart (as can be witnessed from Figures ?? and ?? and Figures ?? and ?? representing the ‘ $T = 20, n = 5$ ’-case with  $p = 25$  and  $p = 50$ , respectively).

Increasing the number of samples to the practically non-observed (confer ??)  $n = 15$  benefits the SCAD estimator in the cases with  $p = 25$  and the both sparse choices of  $\mathbf{A}$  (confer Figures ??, ??, ??, and ??). When the number of variates increases to  $p = 50$  the SCAD still outperforms the ridge estimator in for sparsest  $\mathbf{A}$  (Figure ??) but roles are reversed when  $\mathbf{A}$  contains roughly 25% nonzero elements (Figure ??).

In the latter the superior Frobenius loss of the SCAD estimator is consequence of the sparse  $\hat{\mathbf{A}}(\lambda_a)$  and  $\hat{\mathbf{\Omega}}_\epsilon(\lambda_\omega)$  which favours the SCAD estimator due to the sparsity of the estimates. This is verified by comparing the ridge and SCAD estimates separately for zero and non-zero elements of  $\mathbf{A}$  (not shown). The SCAD estimator performs better for the zero elements, while the ridge estimator outperforms on non-zero elements of  $\mathbf{A}$ . Only for the hub network does the SCAD estimator generally outperforms the ridge estimator.

We now turn to the sensitivity and specificity of the ridge procedure and that of the SCAD procedure (Figures ??, ??, ??, ??, ??, ??, ??, and ??). Recall, to make the sensitivity and specificity more comparable, the ridge procedure selects (from the largest elements elements of its estimate of  $\mathbf{A}$ ) the same number of edges as found by the SCAD method, thus possibly favouring the latter. In the general picture that then emerges the methods are on a par. For the lower sample sized (and thereby more prevalent) settings a difference is virtually absent. More discriminative power is present when  $n = 15$  and *mathcal{T*, but even then no clear winner appear (although then the hub case appears to favour the SCAD method).

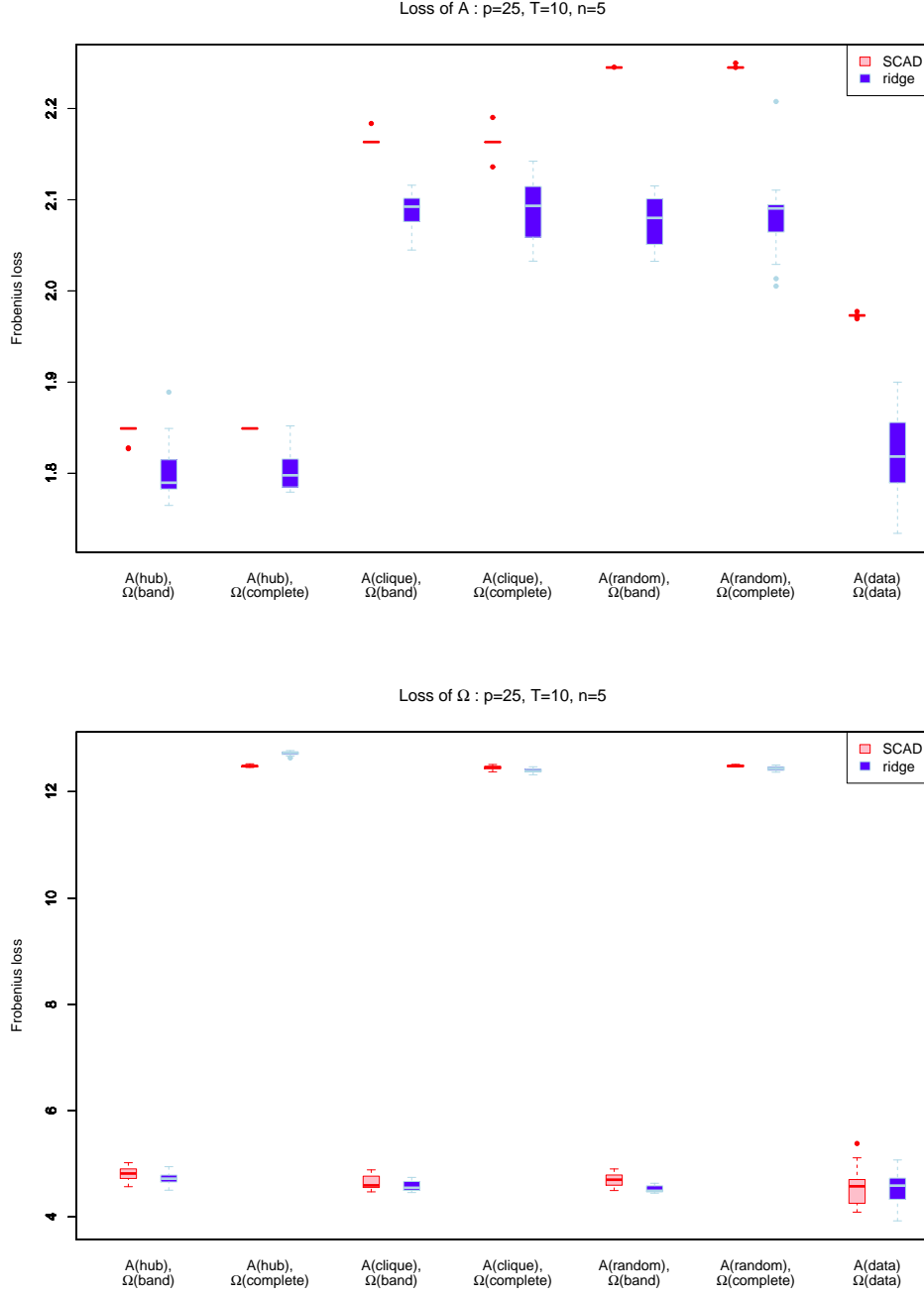


Figure 6: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

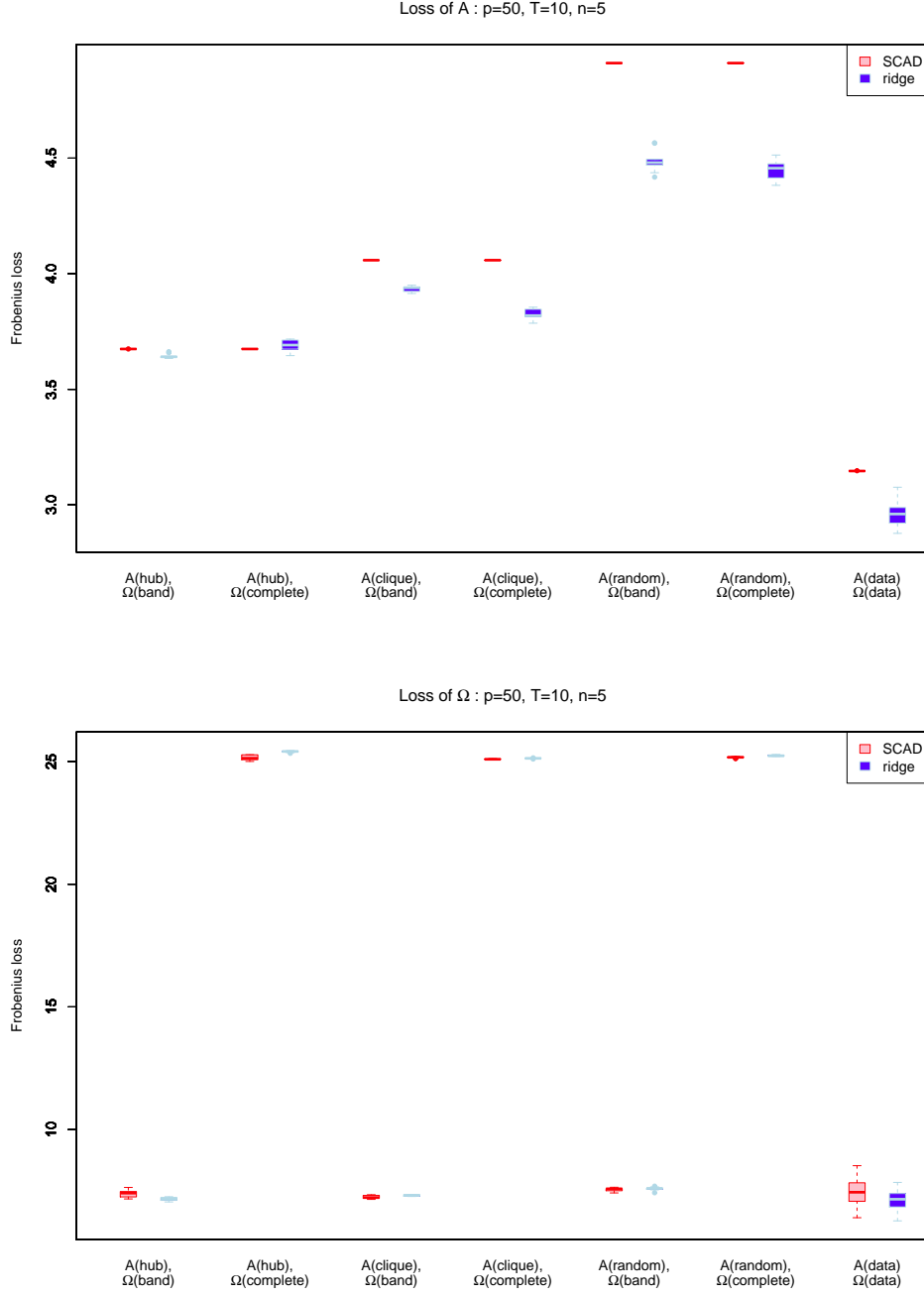


Figure 7: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

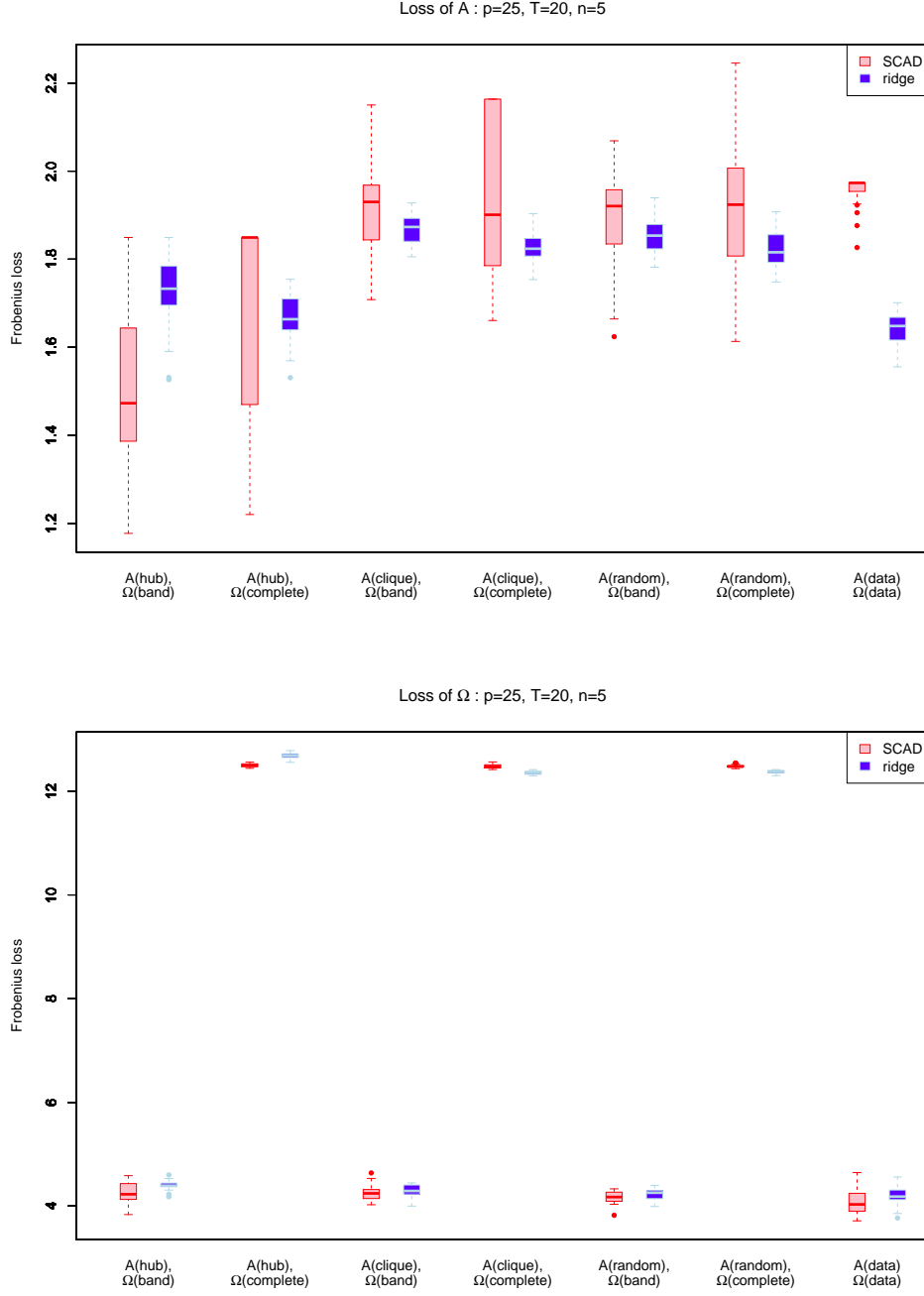


Figure 8: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25$ ,  $T=20$ ,  $n=5$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

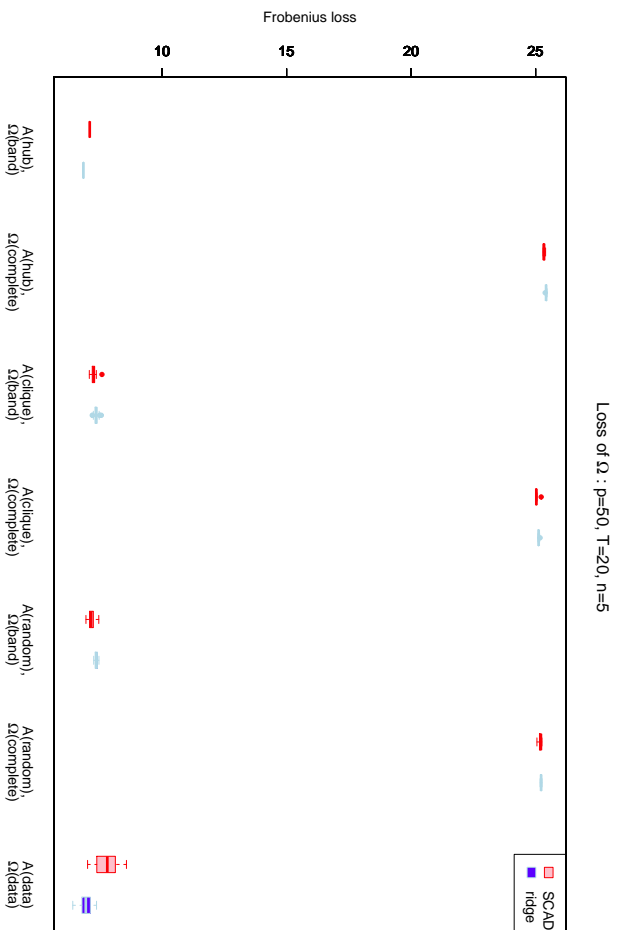
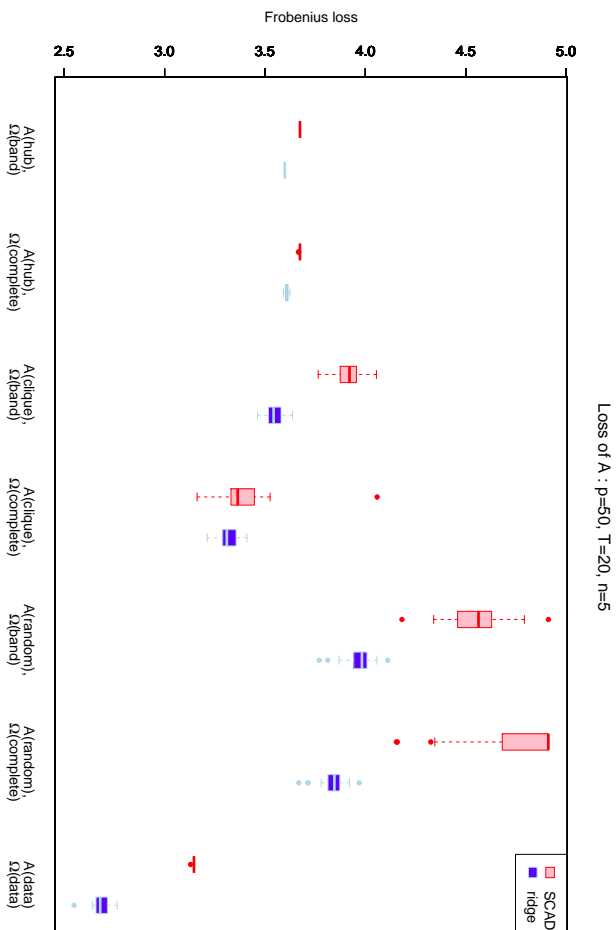


Figure 9: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50$ ,  $T=20$ ,  $n=5$  and  $\mathbf{A}$  with roughly 5% nonzero elements.



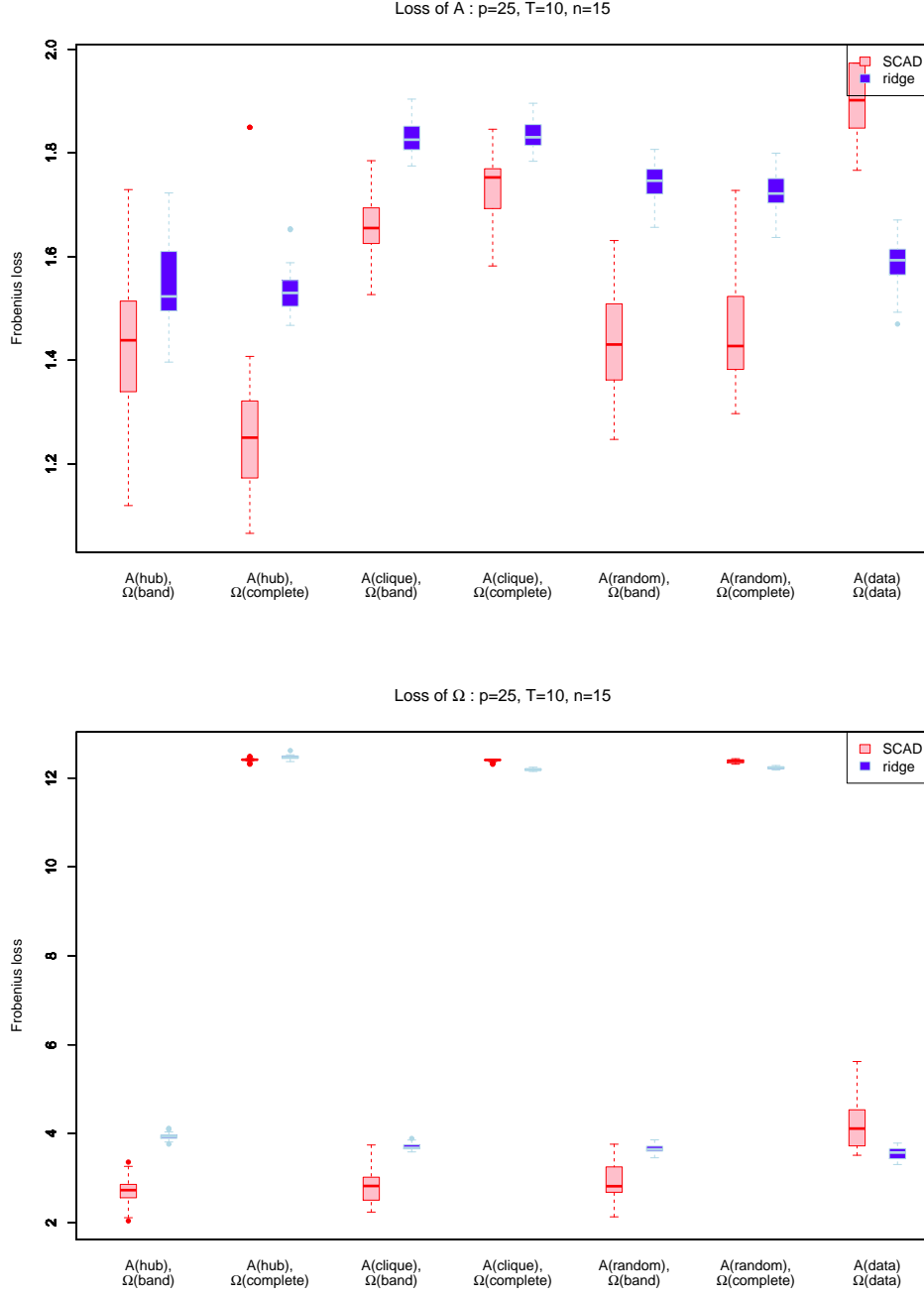


Figure 10: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25, T=10, n=15$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

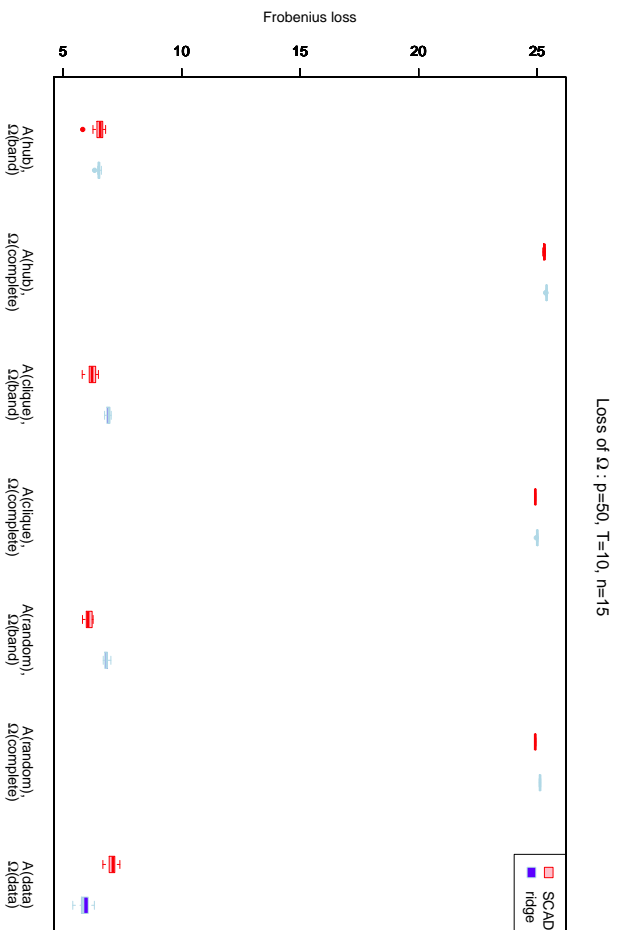
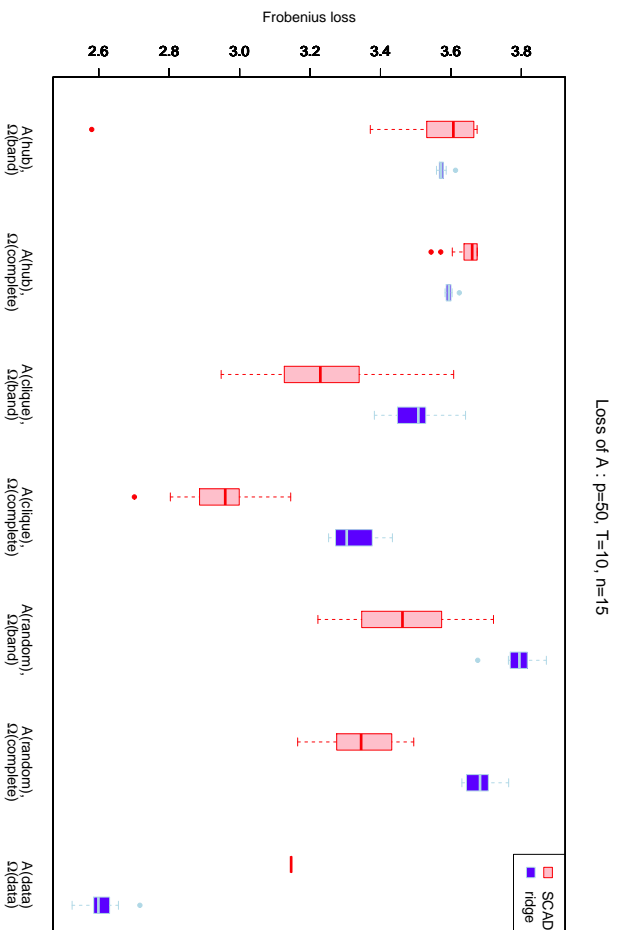


Figure 11: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50$ ,  $T=10$ ,  $n=15$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

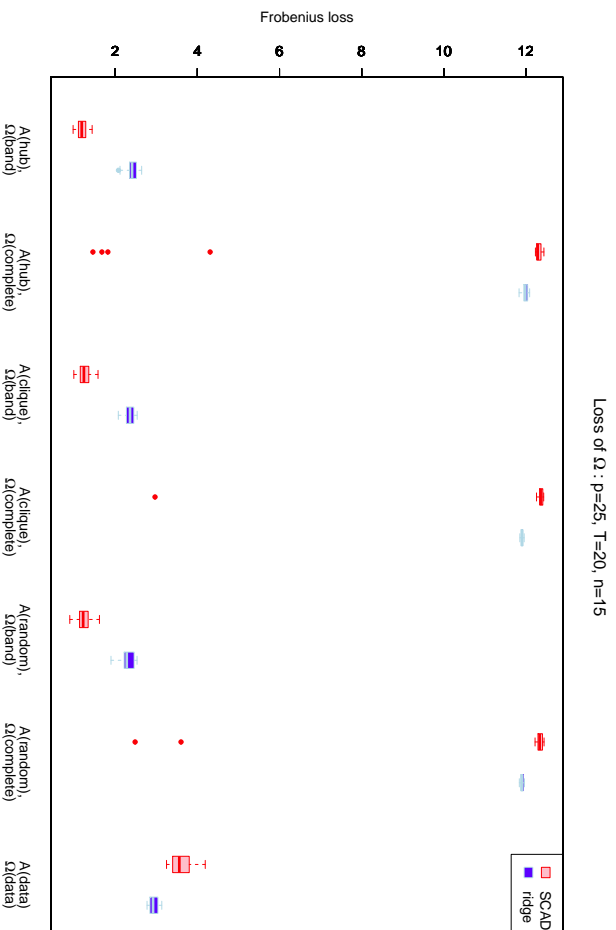
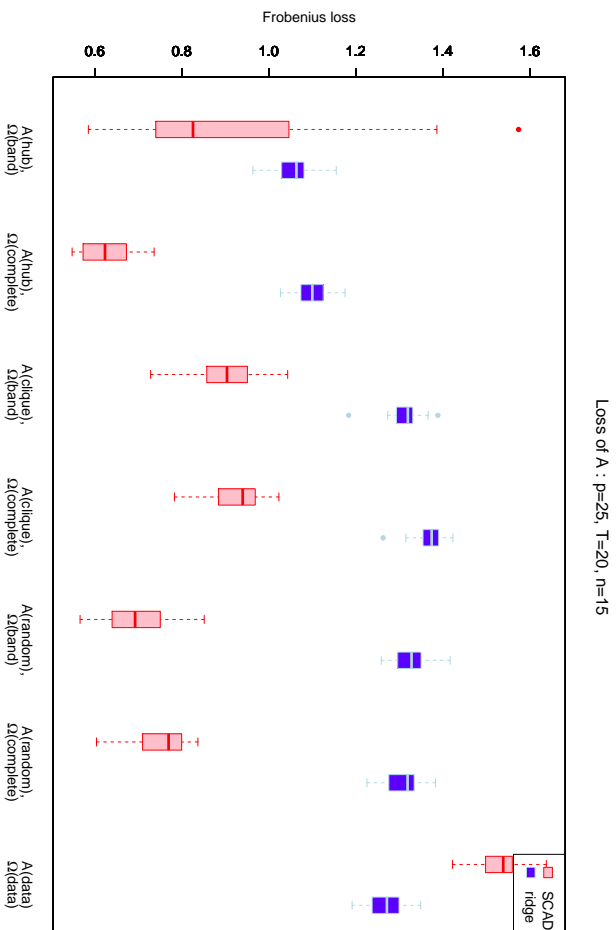


Figure 12: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25$ ,  $T=20$ ,  $n=15$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

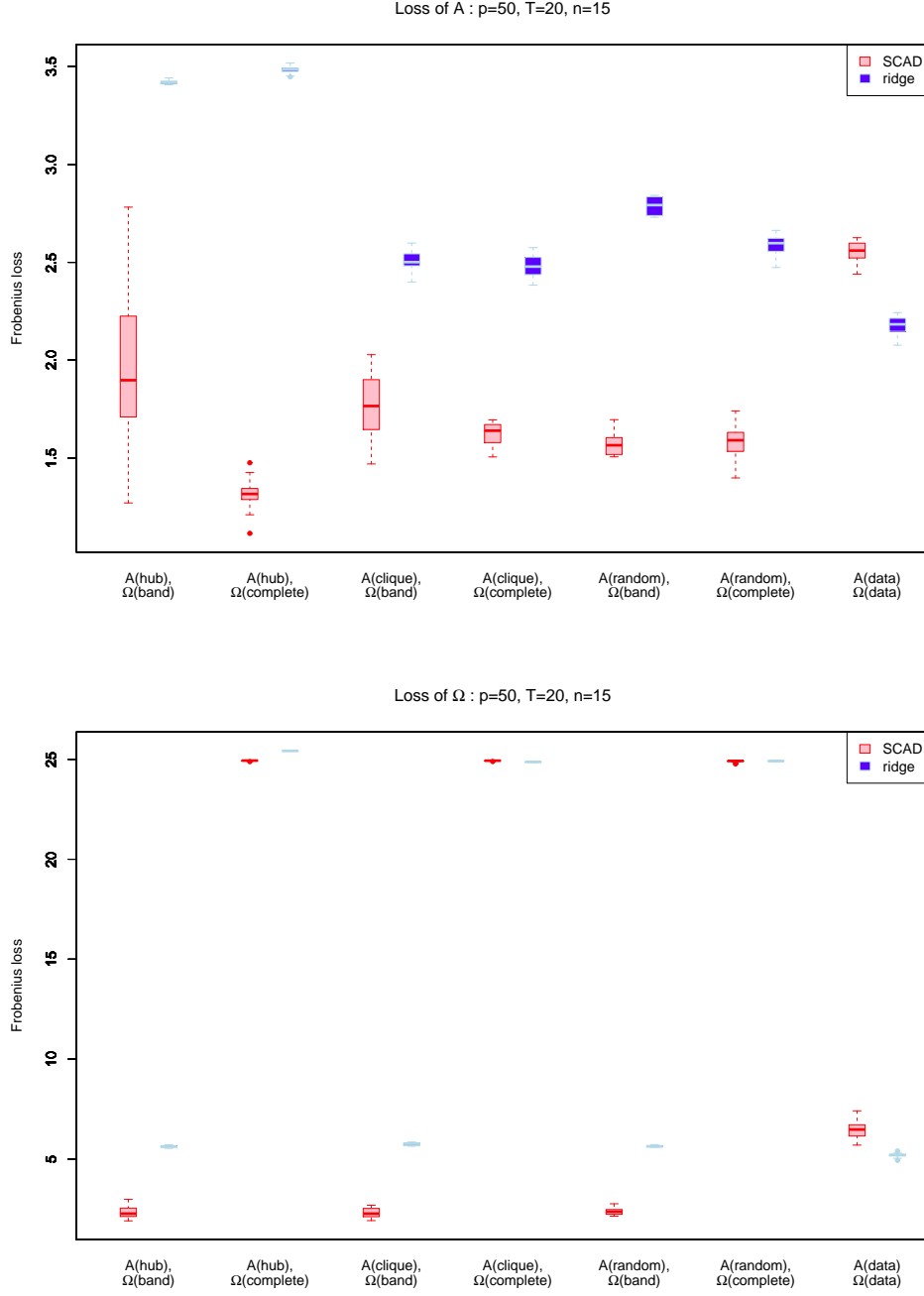


Figure 13: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50, T=20, n=15$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

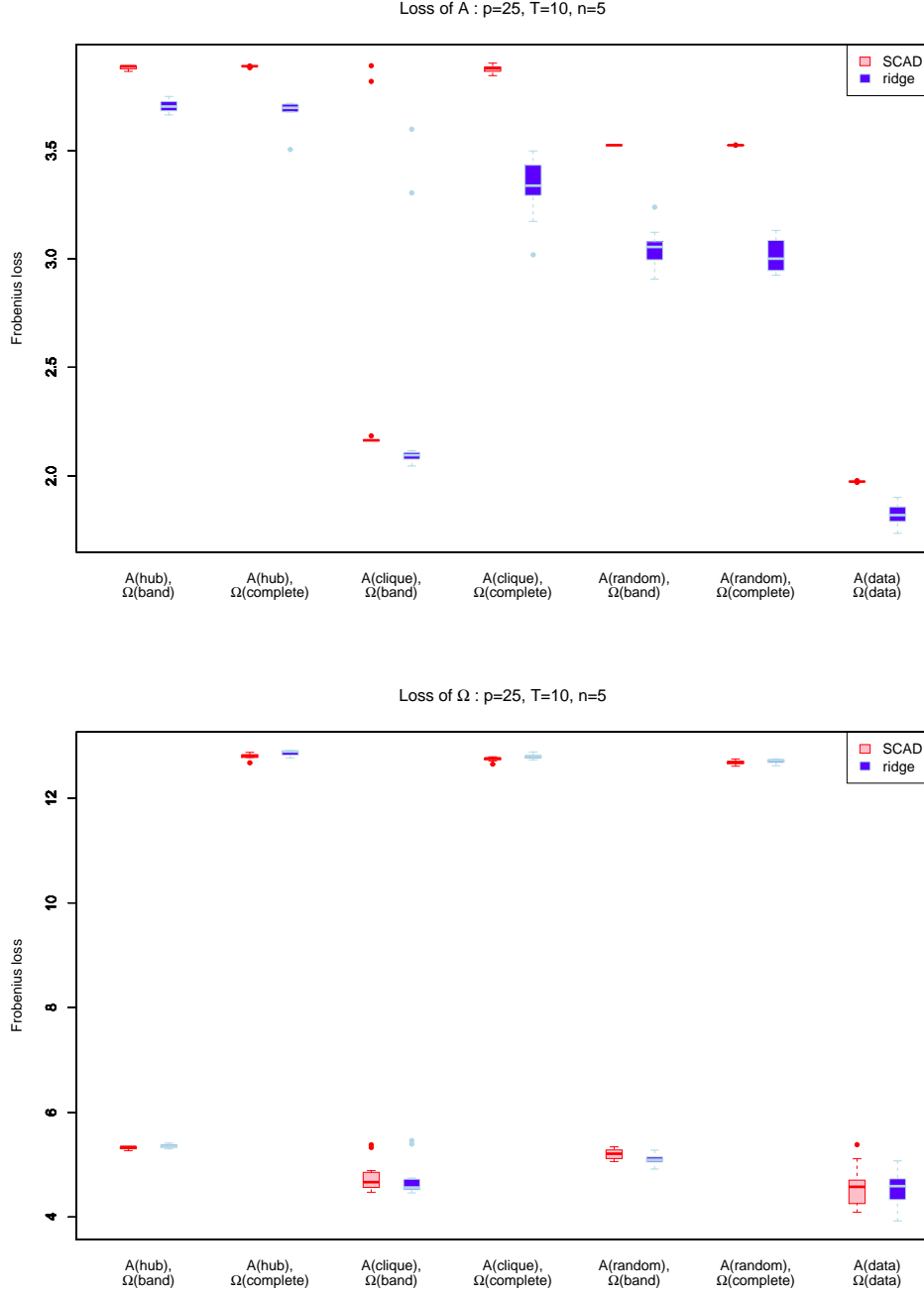


Figure 14: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

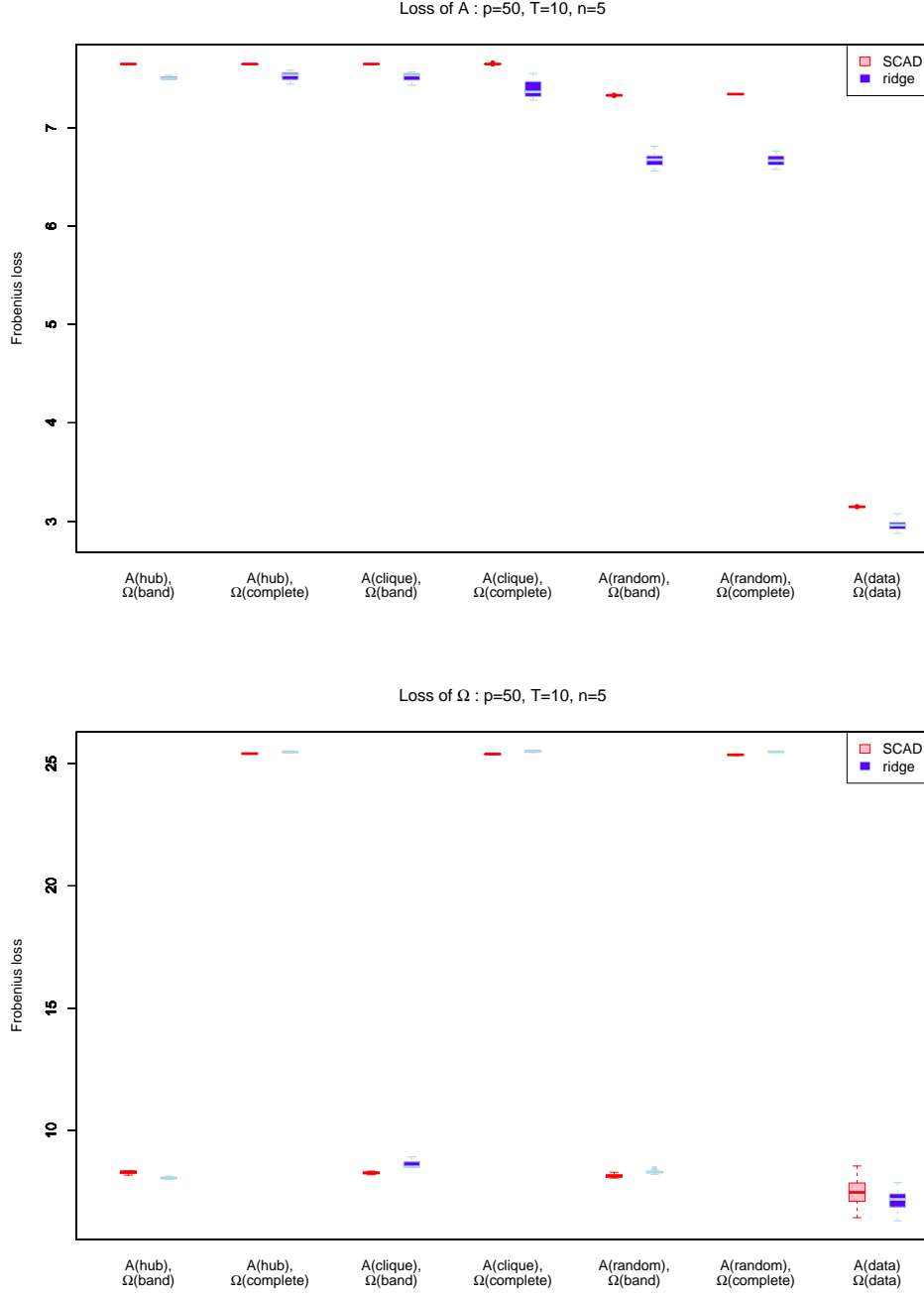


Figure 15: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

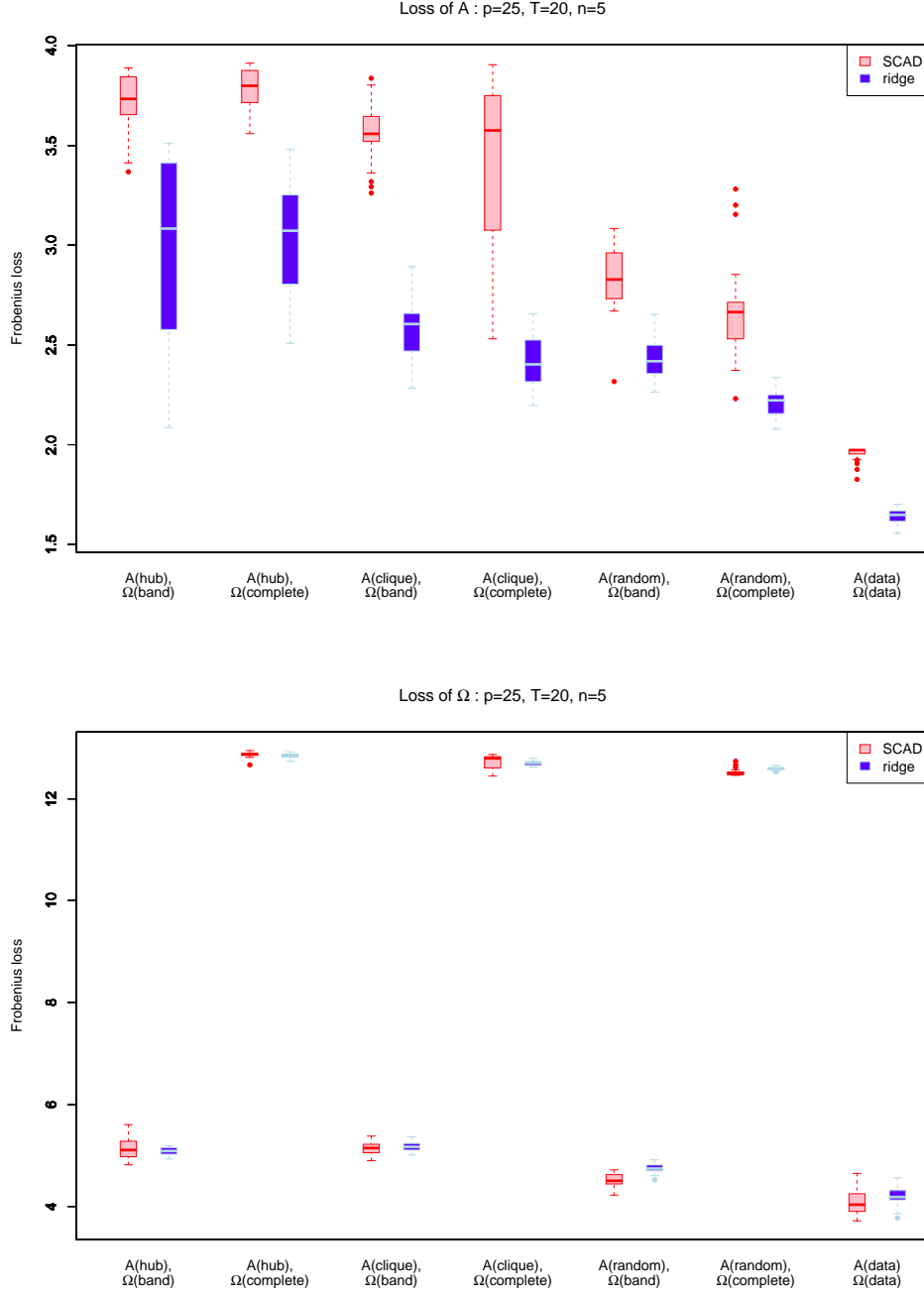


Figure 16: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25$ ,  $T=20$ ,  $n=5$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

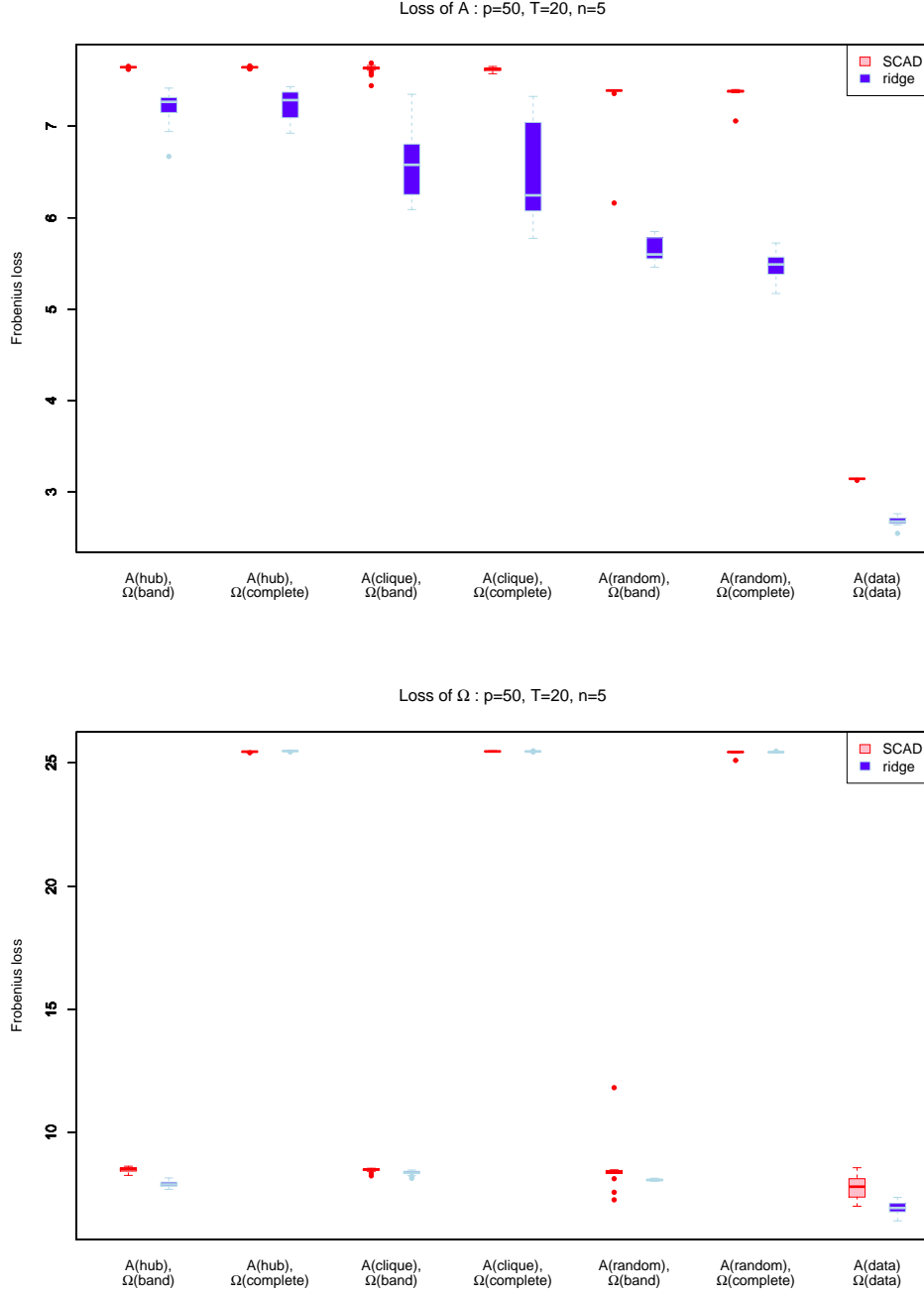


Figure 17: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50$ ,  $T=20$ ,  $n=5$  and  $\mathbf{A}$  with roughly 25% nonzero elements.



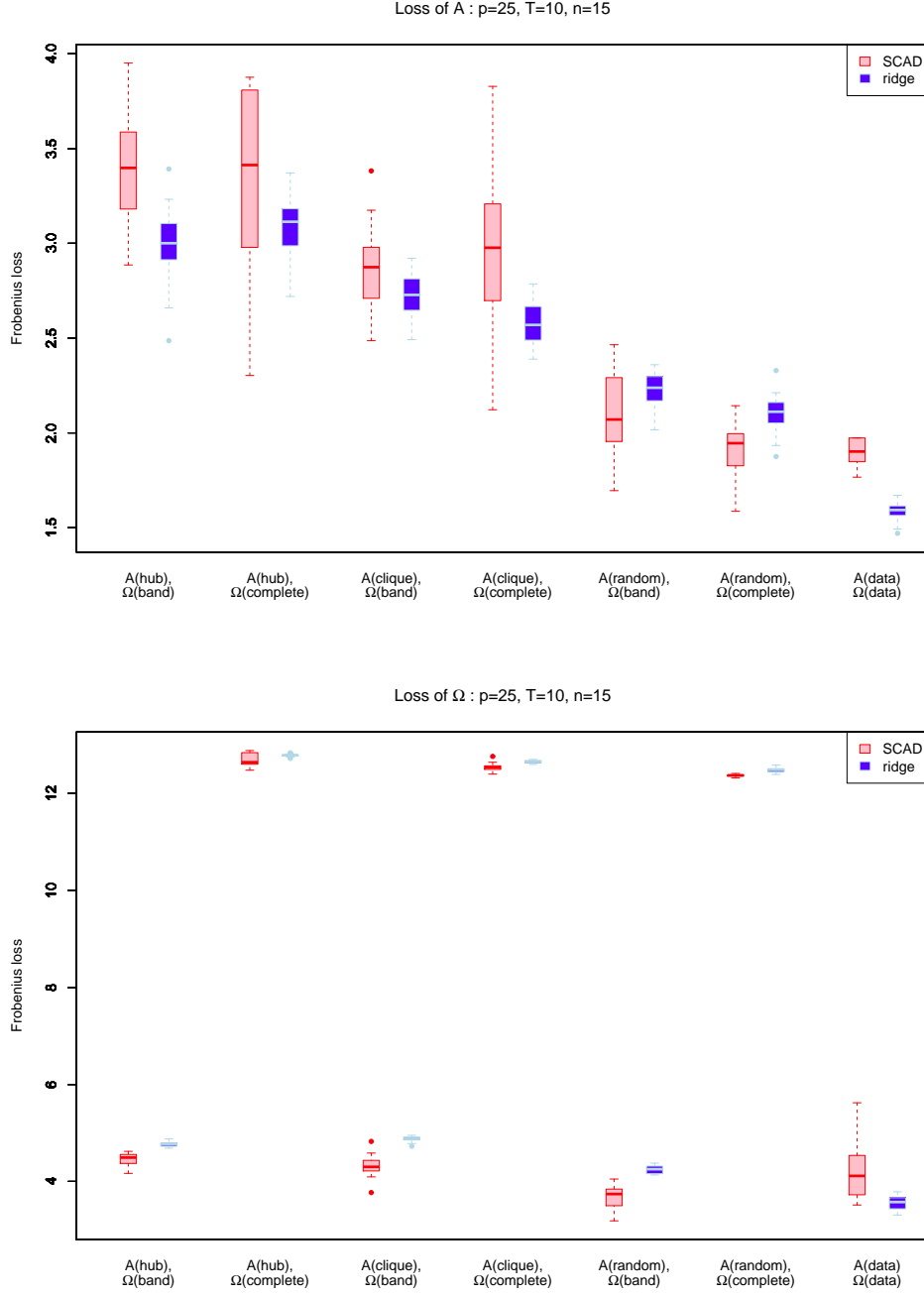


Figure 18: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25, T=10, n=15$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

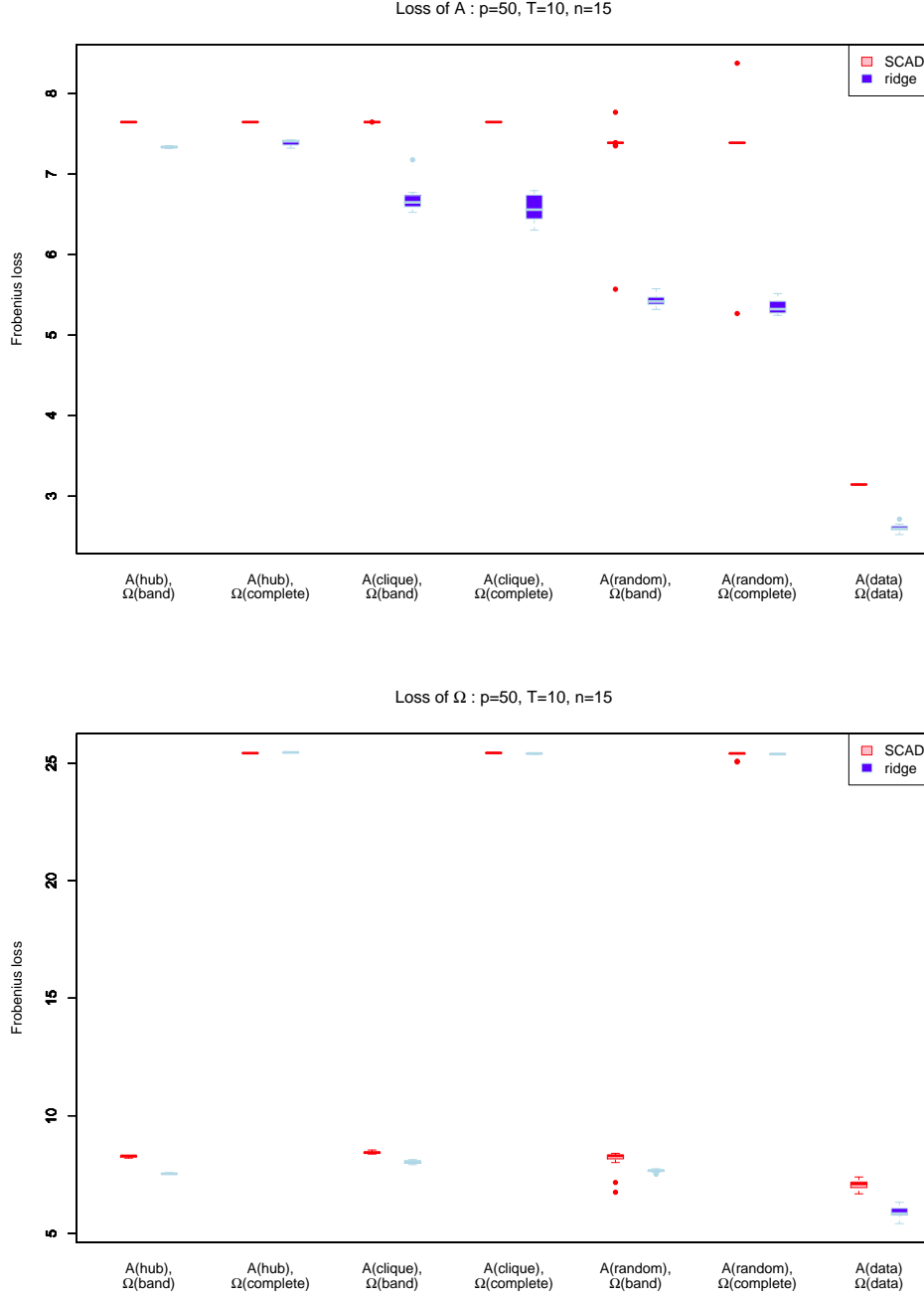


Figure 19: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50, T=10, n=15$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

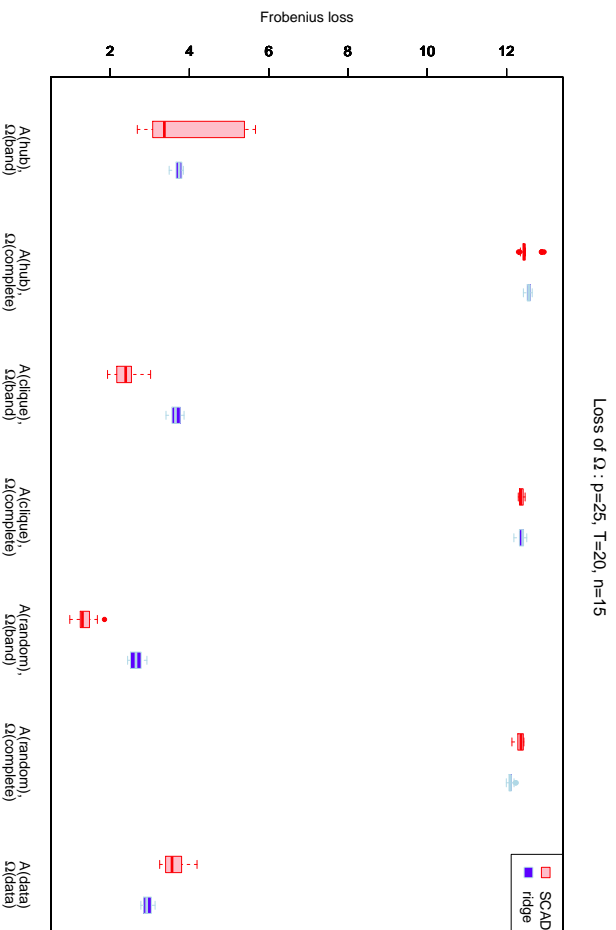
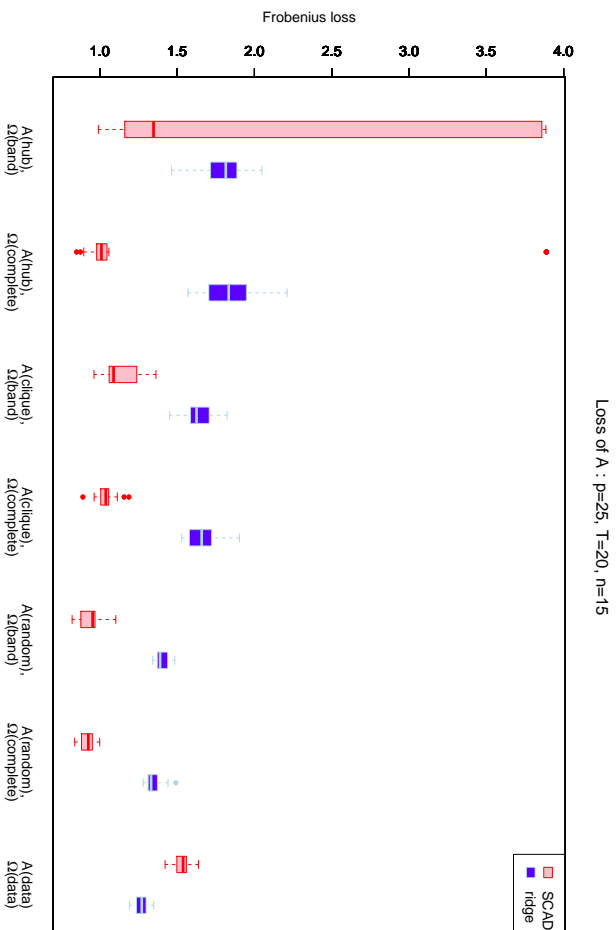


Figure 20: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=25$ ,  $T=20$ ,  $n=15$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

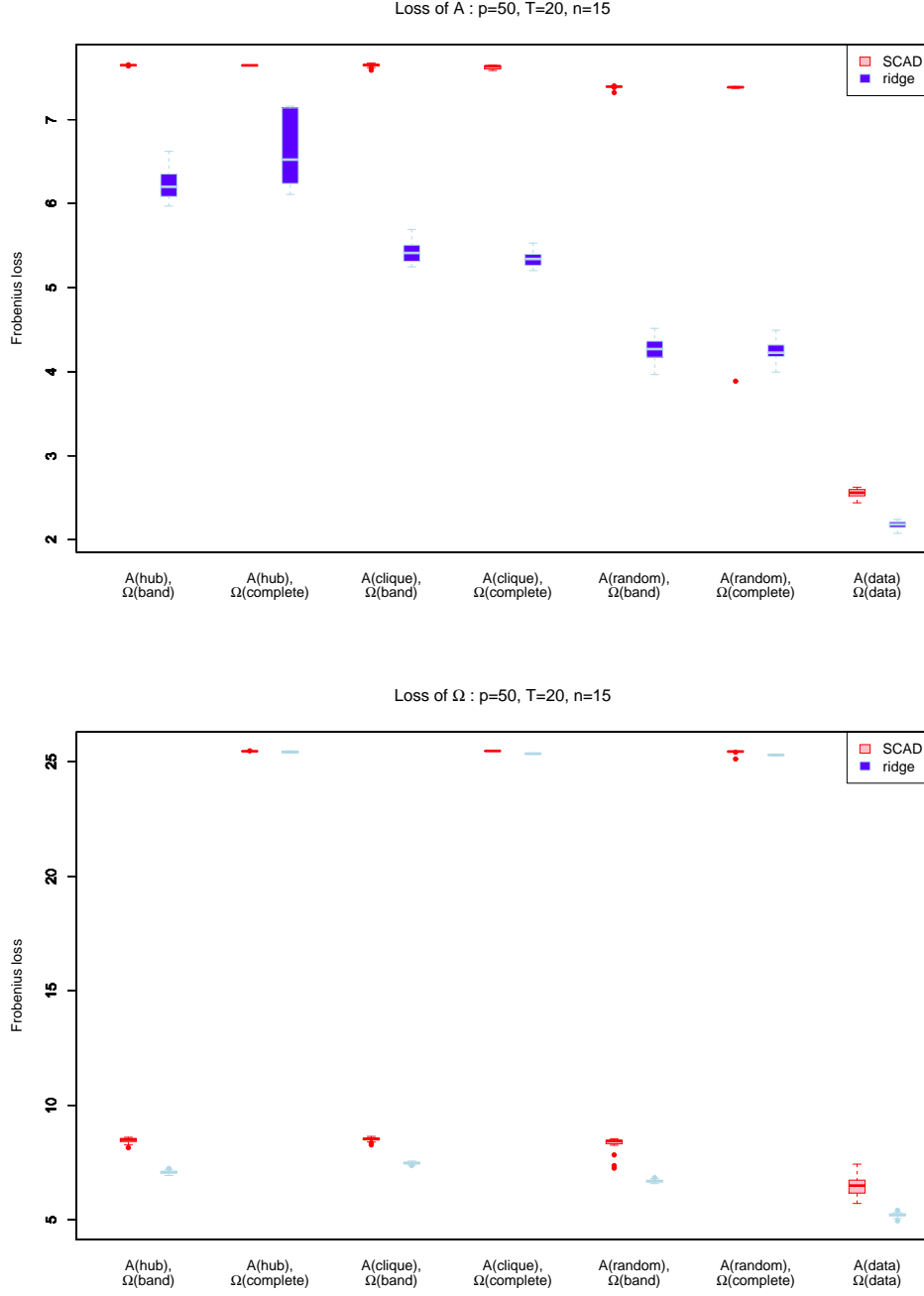


Figure 21: Frobenius loss comparison between SCAD and ridge estimators for precision and autoregressive coefficient matrix on simulated data set where  $p=50, T=20, n=15$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

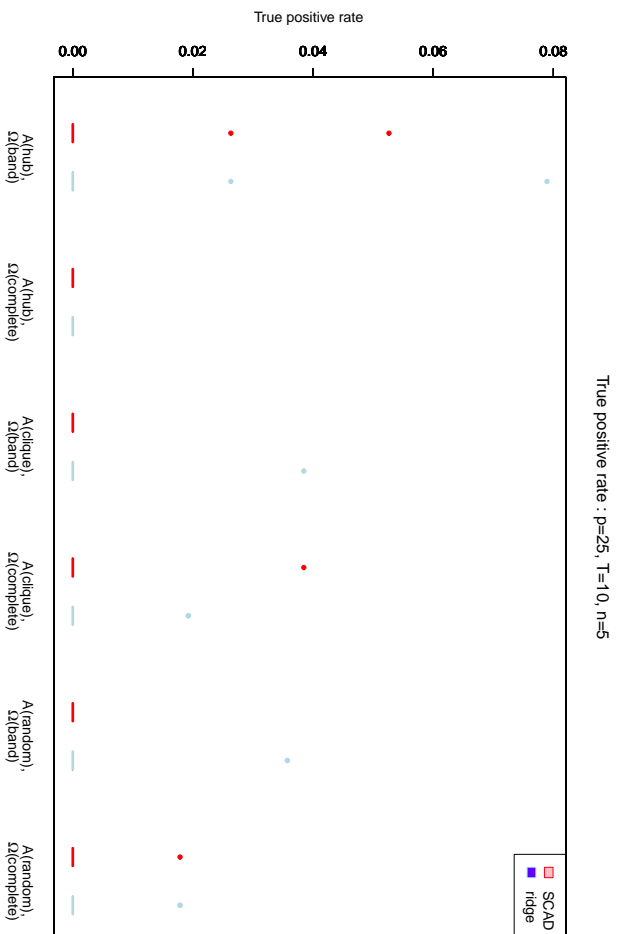
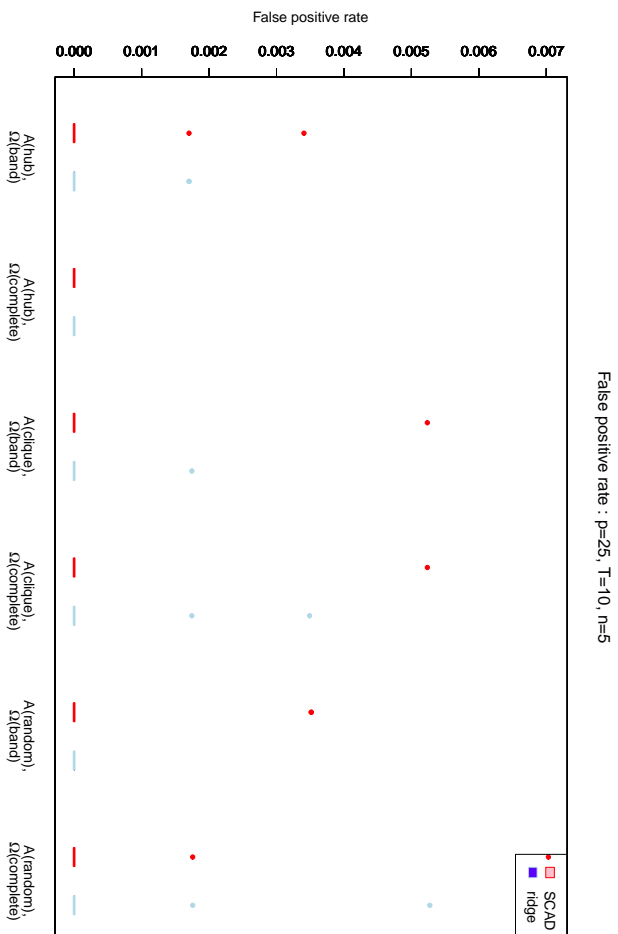


Figure 22: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=25$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

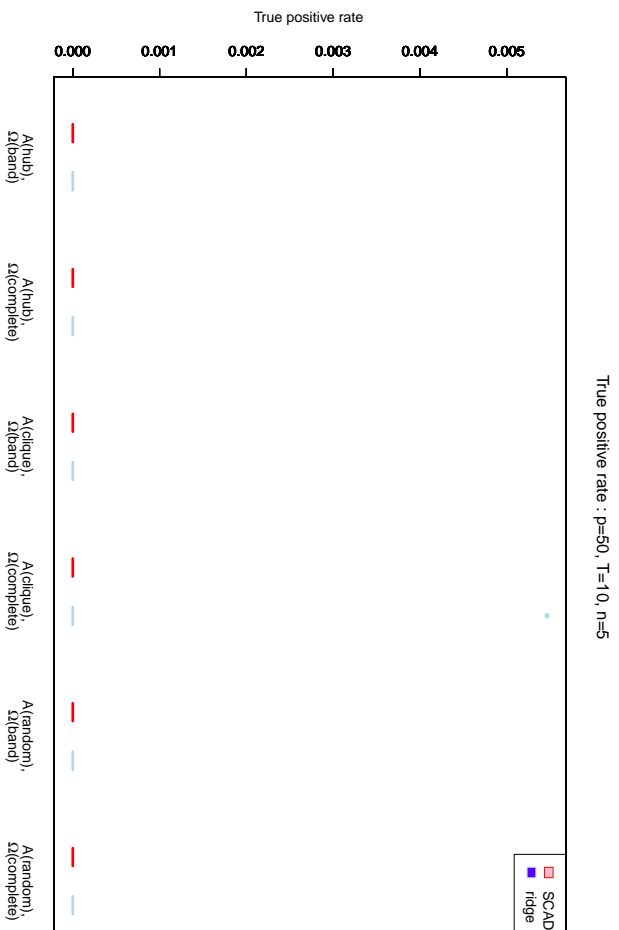
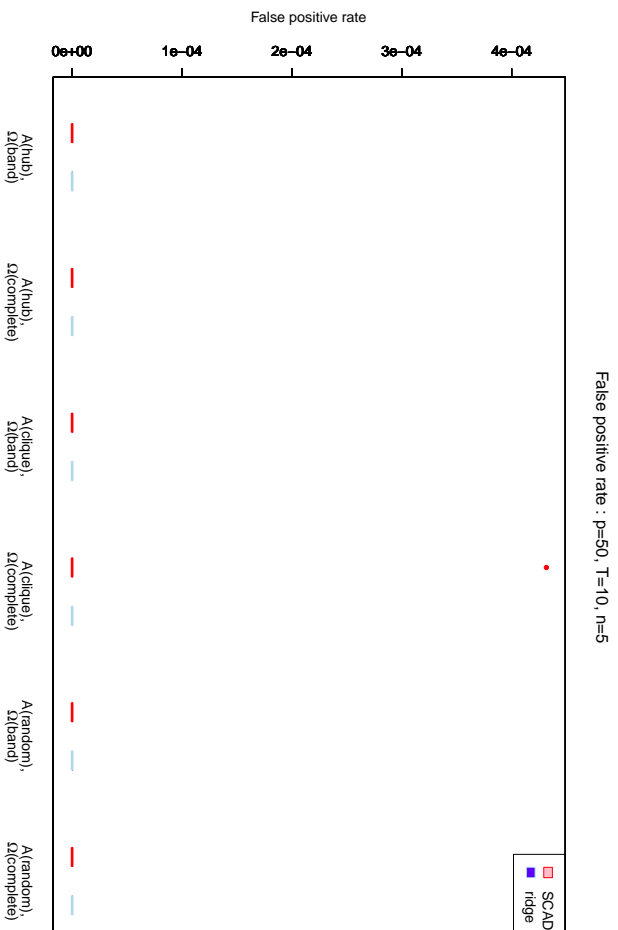


Figure 23: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=50$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

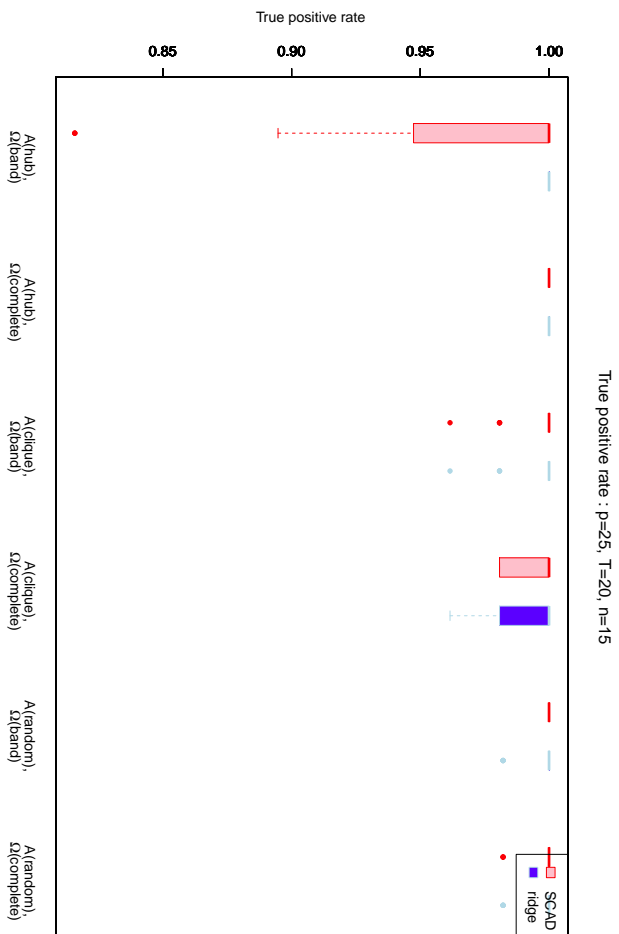
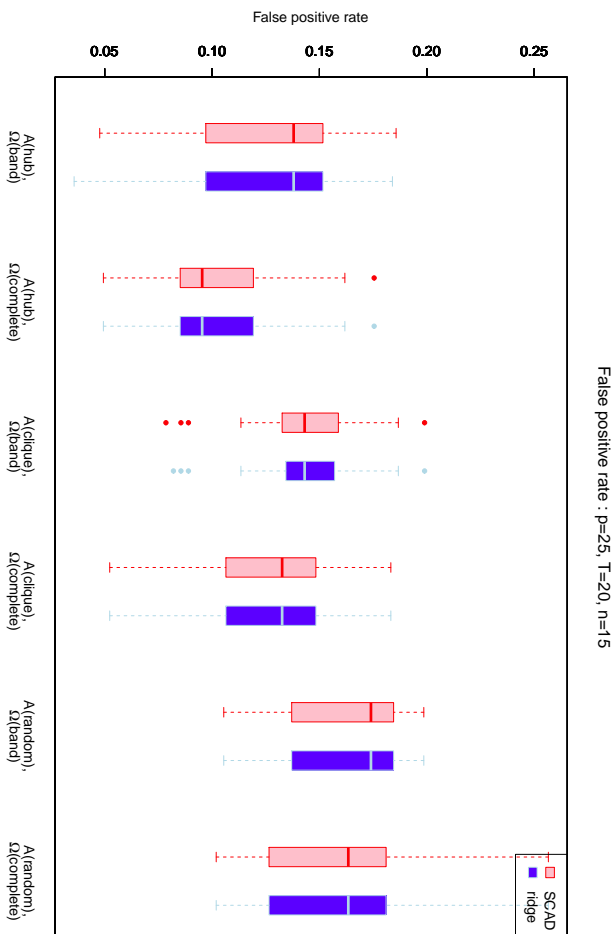


Figure 24: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=25$ ,  $T=20$ ,  $n=15$  and  $\mathbf{A}$  with roughly 5% nonzero elements.

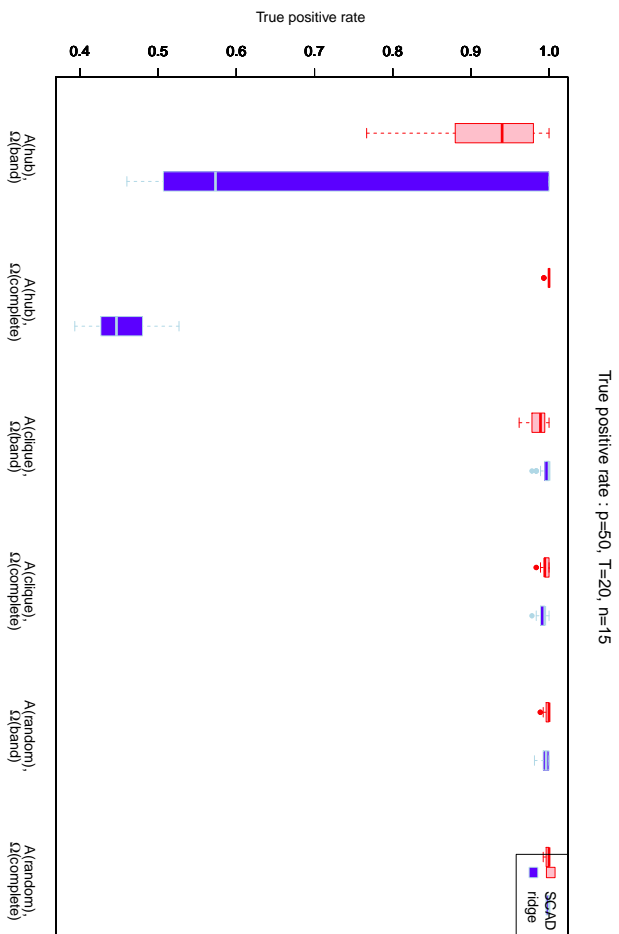
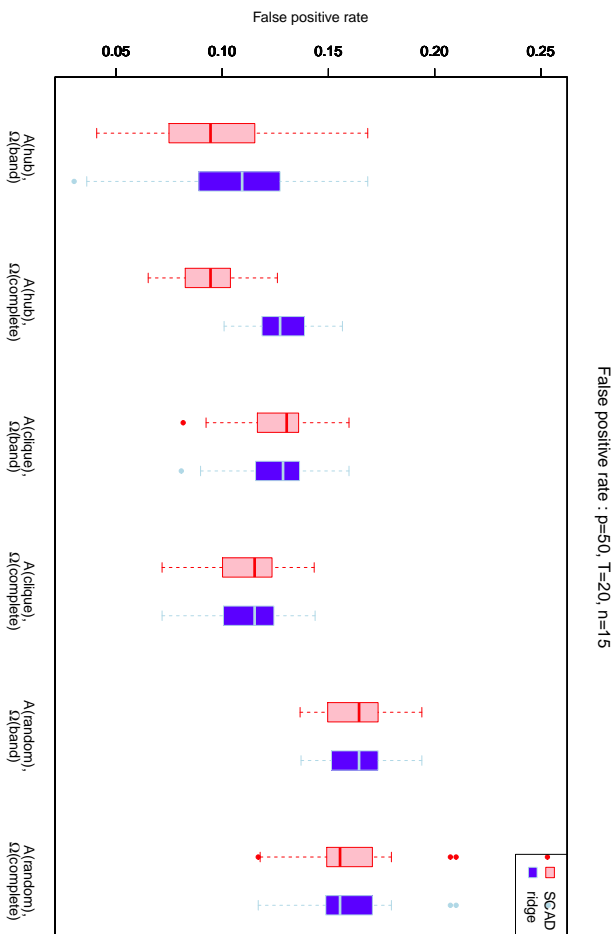


Figure 25: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=50$ ,  $T=20$ ,  $n=15$  and  $\mathbf{A}$  with roughly 5% nonzero elements.



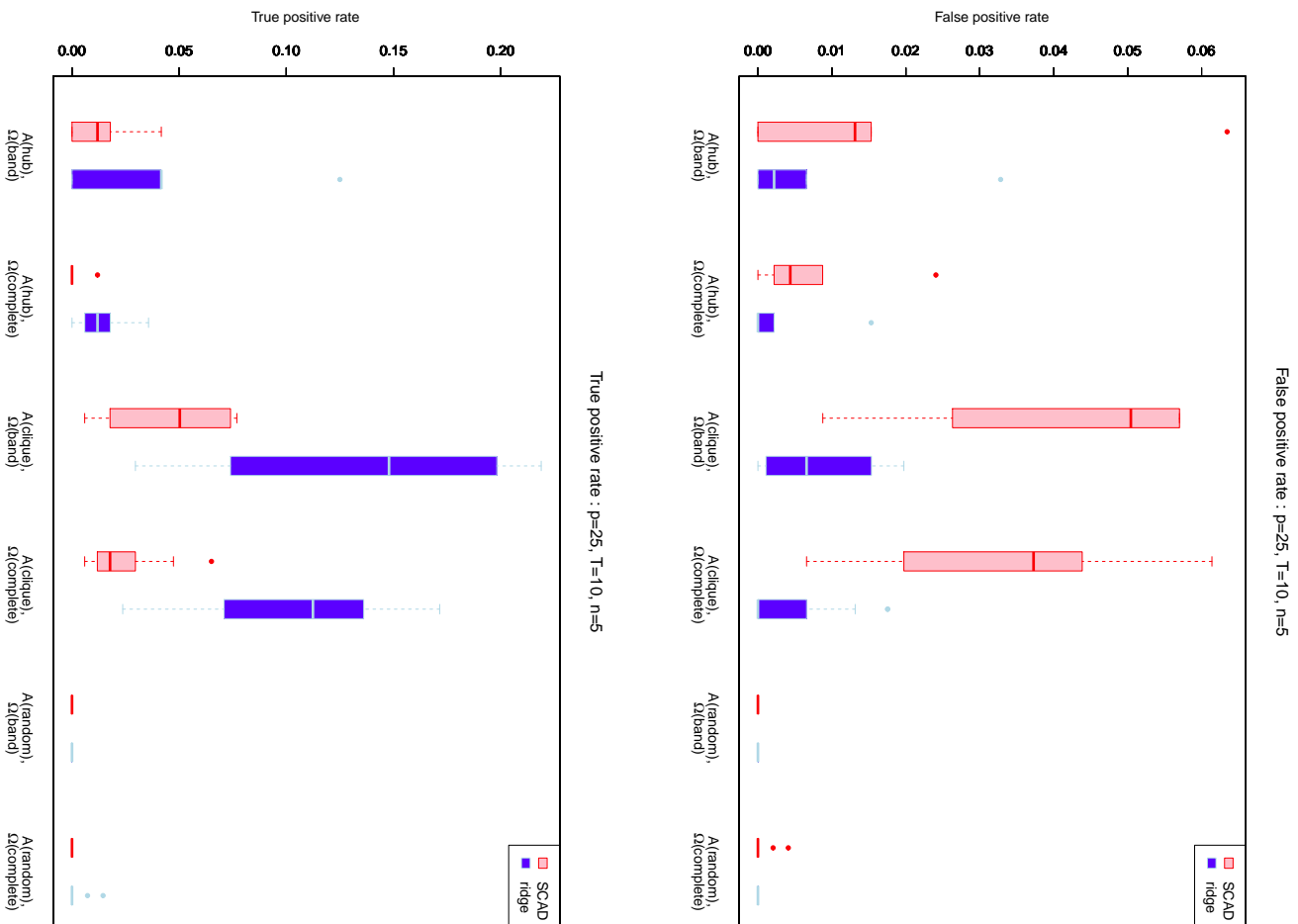


Figure 26: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=25$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

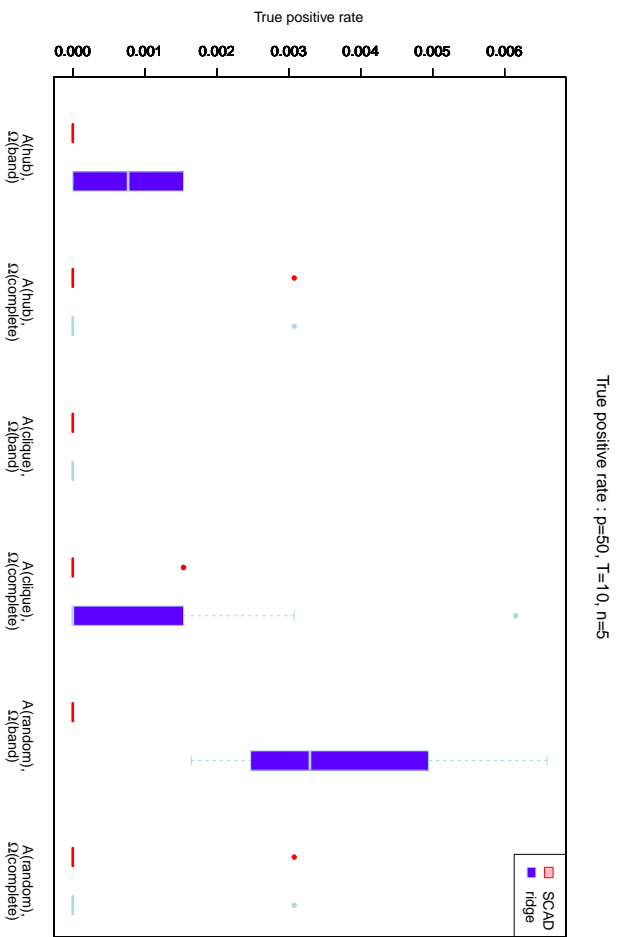
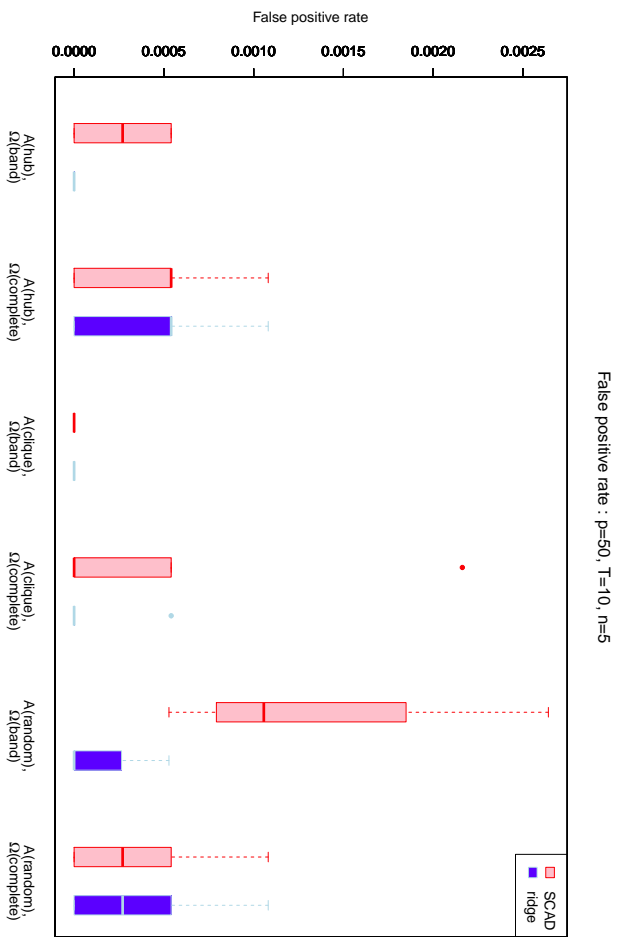


Figure 27: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=50$ ,  $T=10$ ,  $n=5$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

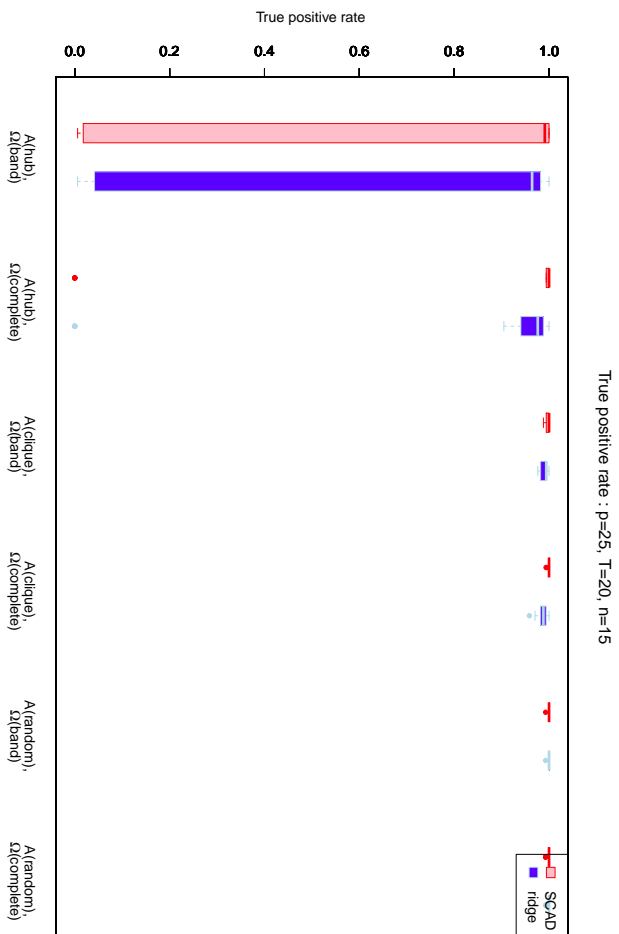
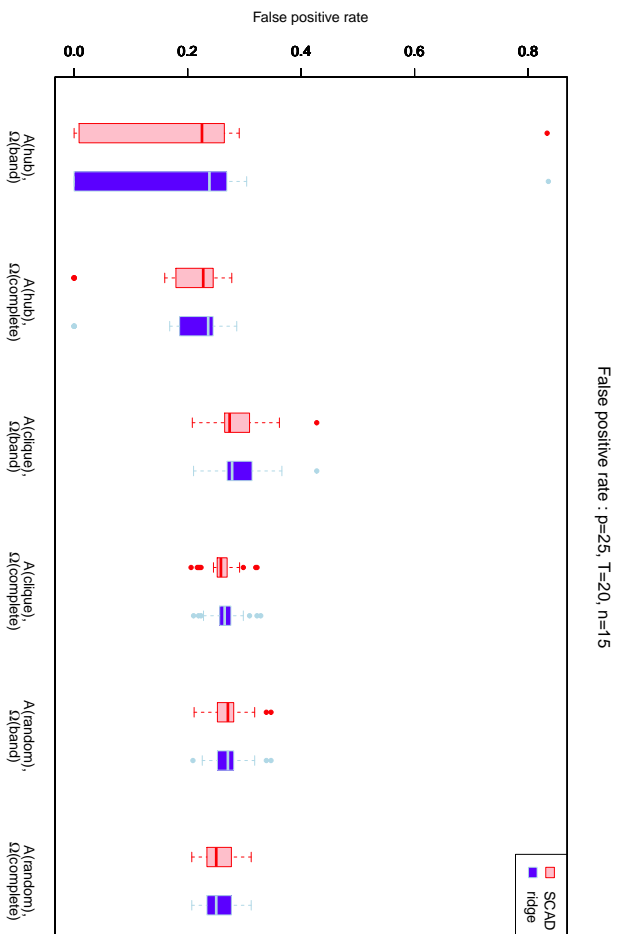


Figure 28: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=25$ ,  $T=20$ ,  $n=15$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

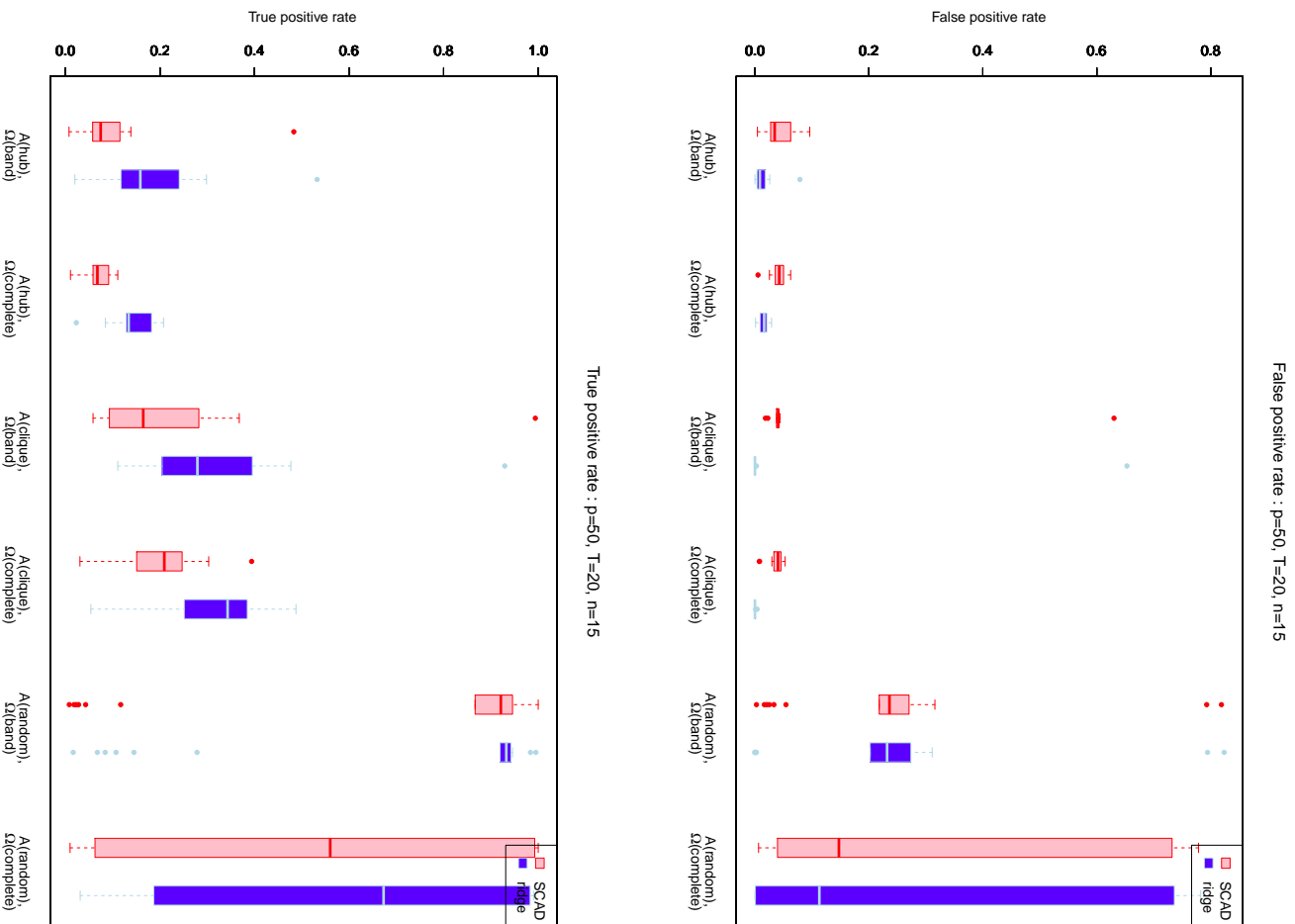


Figure 29: Box plot of specificity (false positive rate) and sensitivity (true positive rate) of the ridge and SCAD methods on simulated data where  $p=50$ ,  $T=20$ ,  $n=15$  and  $\mathbf{A}$  with roughly 25% nonzero elements.

Listing 1: R code

```
# set seed
set.seed(321)

# load libraries
library(longitudinal)
library(Biobase)
library(Matrix)
library(SparseTSCGM)
library(ragt2ridges)

# declare function
loglikLOOCVVAR1sim <- function (lambdas, Y, penalty="ridge", ...){
  loglik <- 0

  # ridge leave K-fold-out cross-validated log-likelihood
  if(penalty=="ridge"){
    for (k in 1:dim(Y)[3]) {
      # Ridge ML estimation of the VAR(1) model
      VAR1hat <- ridgeVAR1(Y[,,-k], lambdas[1], lambdas[2], ...)
      for (l in 2:dim(Y)[2]){
        res <- Y[, l, k] - VAR1hat$A %*%Y[, l-1, k]
        loglik <- loglik - t(res) %*% VAR1hat$P %*% res/2 +
          log(det(VAR1hat$P))/2
      }
    }
  }

  # SCAD leave K-fold-out cross-validated log-likelihood
  if(penalty=="scad"){
    for (k in 1:dim(Y)[3]) {
      # convert the data without k-th cell line
      Yscad <- array2longitudinal(Y[,,-k])
      Yscad1 <- as.longitudinal(Yscad, repeats=dim(Y)[3]-1)

      # SCAD estimation of VAR(1) model
      VAR1hat <- sparse.tscgm(data=Yscad1, lam1=lambdas[1],
        lam2=lambdas[2], model="ar1", optimality=NULL,
        control=list(maxit.out=5, maxit.in=10))
      Omega <- VAR1hat$theta
      for (l in 2:dim(Y)[2]){
        res <- Y[, l, k] - t(VAR1hat$gamma) %*%Y[, l-1, k]
        loglik <- loglik - t(res) %*% Omega %*% res/2 +
          log(det(Omega))/2
      }
    }
  }

  return(-loglik)
}

# set number variates, samples, time points
p=25
```

```

n=15
T=20

# generate autoregressive coefficient matrix
trueA <- createA(p, "clique", 0.3, 8)

# generate precision matrix
trueP <- createS(n, p, "banded", nonzero=0.5, banded.n=3, precision=T)

# sample data from a VAR(1) model
Y <- dataVAR1(n, T, trueA, solve(trueP))

# covariate-wise zero centering of the data
Yridge <- centerVAR1data(Y)

# optimal lambdas for ragt2ridges
lamR<-seq(4,0.01,length.out=20)
LOOCVrPrev <- loglikLOOCVVAR1sim(c(lamR[1],lamR[1]), Yridge,
                                penalty="ridge")
LOOCVr <- numeric()
for(i in 1:length(lamR)) {
  LOOCVr <- cbind(LOOCVr,loglikLOOCVVAR1sim(c(lamR[i],lamR[i]),
                                             Yridge, penalty="ridge"))
  if (LOOCVr[i]-LOOCVrPrev > 5){
    k <- which.min(LOOCVr)
    break
  } else {
    LOOCVrPrev = LOOCVr[i]
    k <- which.min(LOOCVr)
  }
}
optLr <- nlminb(c(lamR[k], lamR[k]), loglikLOOCVVAR1sim,gradient=NULL,
               lower=c(10^(-10),10^(-10)), Y=Yridge, penalty="ridge",
               control=list(rel.tol=10^(-10)))$par

# Ridge ML eximtion of the VAR(1) model
ridgeEst <- ridgeVAR1(Y=Yridge, lambdaA=optLr[1], lambdaP=optLr[2])
ridgeA <- ridgeEst$A
ridgeP <- ridgeEst$P

# convert a time-series array to a longitudinal object required for
  SparseTSCGM
Yscad <- array2longitudinal(Yridge)
Yscad1 <- as.longitudinal(Yscad,repeats=n)

# optimal lambdas for ridge method
lamS<-seq(2,0.01,length.out =20)
LOOCVs <- numeric()
LOOCVsPrev <- loglikLOOCVVAR1sim(c(lamS[1],lamS[1]), Yridge,
                                penalty="scad")
for(j in 1:length(lamS)) {
  LOOCVs <- cbind(LOOCVs,loglikLOOCVVAR1sim(c(lamS[j],lamS[j]), Yridge,
                                             penalty="scad"))
  if (LOOCVs[j]-LOOCVsPrev>5){

```

```

    l <- which.min(L00CVs)
    break
  } else{
    L00CVsPrev=L00CVs[j]
    l <- which.min(L00CVs)
  }
}
optL1 <- nlminb(c(lamS[1],lamS[1]), loglikL00CVVAR1sim, gradient=NULL,
  lower=c(10^(-10),10^(-10)), Y=Yridge, penalty="scad",
  control=list(rel.tol=10^(-10)))$par

# SCAD estimation of autoregressive coefficient and precision matrix
scadEst <- sparse.tscgm(data=Yscad1, lam1=optL1[1], lam2=optL1[2],
  optimality=NULL)
scadA <- t(scadEst$gamma)
scadP <- scadEst$theta

# Frobenius loss for A and Sigma
sqrt(sum((as.numeric(ridgeP) - as.numeric(trueP))^2))
sqrt(sum((as.numeric(scadP) - as.numeric(trueP))^2))
sqrt(sum((as.numeric(ridgeA) - as.numeric(trueA))^2))
sqrt(sum((as.numeric(scadA) - as.numeric(trueA))^2))

```

## SM VIII: Illustration, additional material

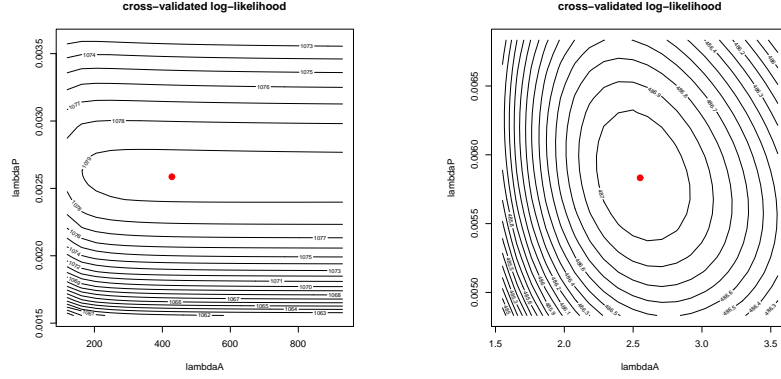


Figure 30: Contour plots of LOOCV log-likelihood of VAR(1) model for the p53 signalling pathway data. Left panel represents the contour plot of the penalty parameters original estimation (without knowledge on the support), while in the right panel that with inferred support supplied.

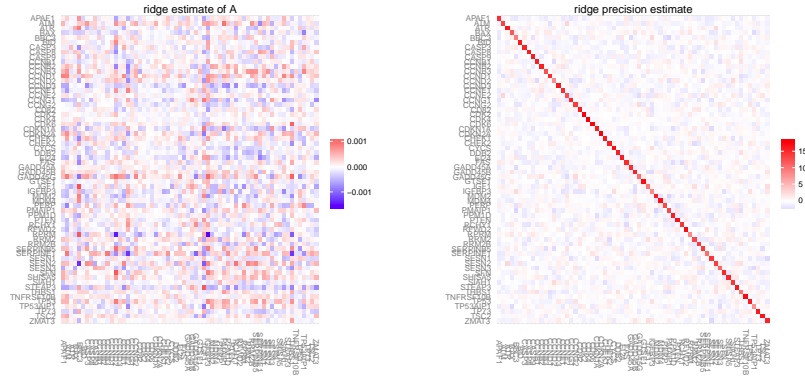


Figure 31: Heatmaps of the ridge estimated parameters for p53 signalling pathway (without knowledge of the support). The left panel contains the estimate of  $\mathbf{A}$ , while the right panel that of  $\mathbf{\Omega}_\epsilon$ . The rows and columns correspond to the 64 (alphabetically ordered) genes of the pathway. Each element (i.e. a small square) of the heatmap represents an element of the estimate. The color intensity represents the size of the estimated element.



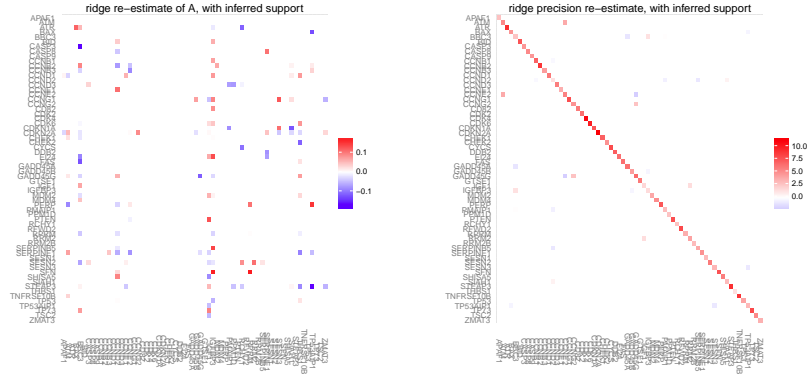


Figure 32: Heatmaps of the ridge estimated parameters for p53 signalling pathway (with inferred support supplied). The left panel contains the estimate of  $\mathbf{A}$ , while the right panel that of  $\mathbf{\Omega}_\epsilon$ . The rows and columns correspond to the 64 (alphabetically ordered) genes of the pathway. Each element (i.e. a small square) of the heatmap represents an element of the estimate. The color intensity represents the size of the estimated element.

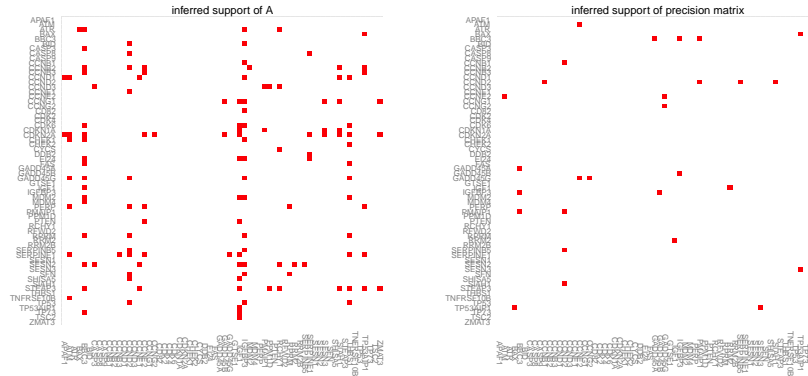


Figure 33: Heatmaps of the support of matrices  $\mathbf{A}$  and  $\mathbf{\Omega}_\epsilon$ . The rows and columns correspond to the 64 (alphabetically ordered) genes of the pathway. Each element (i.e. a small square) of the heatmap represents an element of the estimate. A red element indicates the presence ('nonzero-ness') of the edge, while white indicates its absence.

	$\deg^-(\mathbf{A})$	$\deg^+(\mathbf{A})$	between.	close.	eigen centr.	mutual info.	impulse resp.
BBC3	0	17	17	0.00192	1.00000	0.05605	0.01497
CCND2	0	12	18	0.00187	0.68783	0.01054	0.00874
IGF1	1	14	0	0.00191	0.97635	0.01472	0.01086
IGFBP3	0	16	7	0.00190	0.87513	0.04498	0.01561
THBS1	0	11	0	0.00188	0.87717	0.01172	0.00965
CCNG1	6	0	0	0.00177	0.25154	0.00000	0.00000
CDKN2A	12	0	0	0.00181	0.49508	0.00000	0.00000
SERPINE1	8	4	0	0.00185	0.70869	0.01796	0.00458
SESN2	8	0	0	0.00180	0.26759	0.00000	0.00000
STEAP3	9	0	0	0.00179	0.36285	0.00000	0.00000

Table 2: Node statistics of the inferred time series chain graph of the p53 signalling pathway. The columns contain (from left to right): the gene corresponding to the node; the in-degree of lag-one temporal dependencies; the out-degree of lag-one temporal dependencies; the betweenness in the (global) partial correlation graph; the closeness in the (global) partial correlation graph; the eigen-centrality in the (global) partial correlation graph; the mutual information between the node’s expression level at time  $t$  and that of all pathway nodes at  $t + 1$ ; the impulse response on the node’s expression level at time  $t$  on that of all pathway nodes at  $t + 1$ .

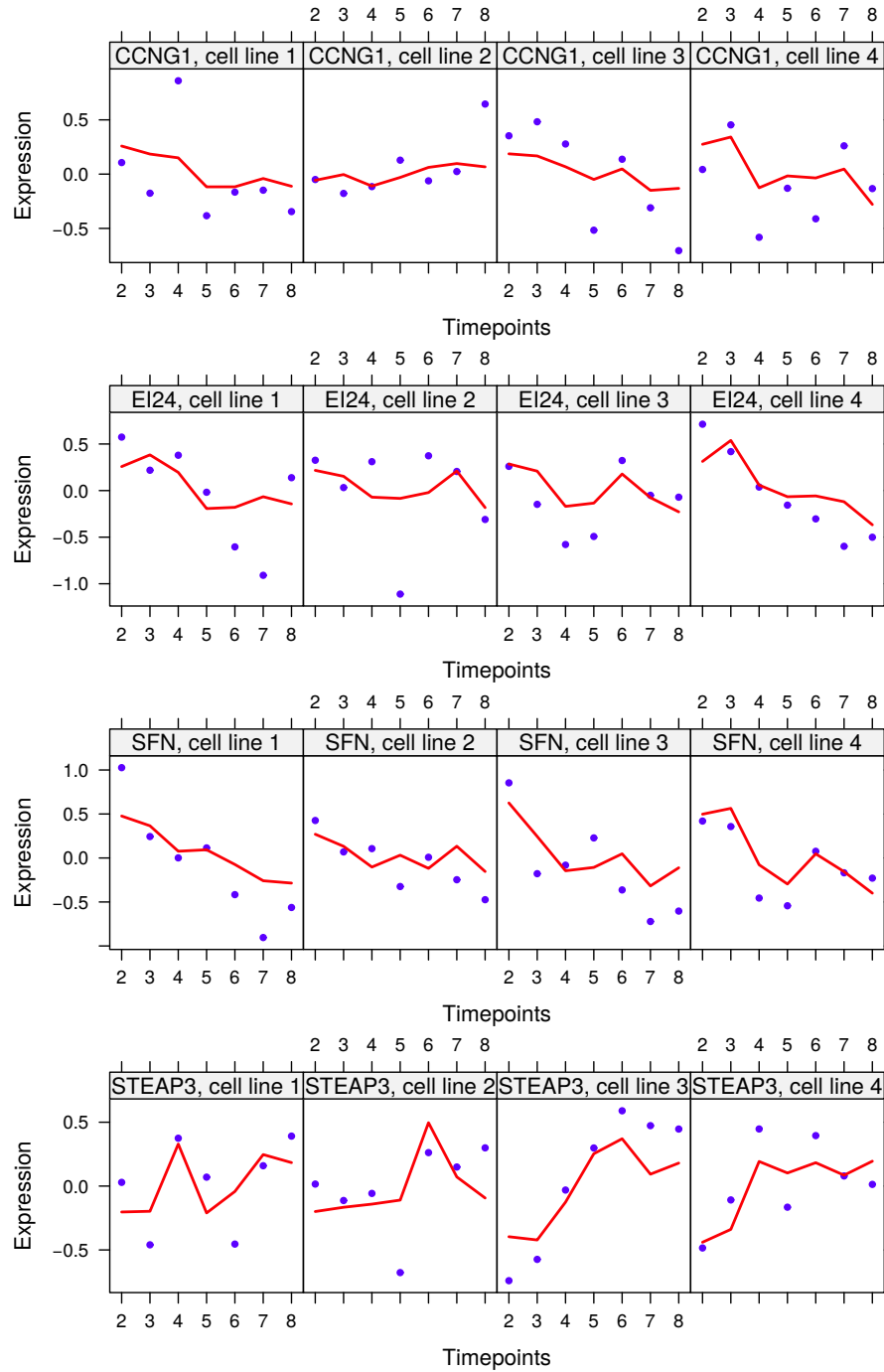


Figure 34: Fit of the expression levels of the CCNG1, EI24, SFN and STEAP3 genes over time in all cell lines. Dots represent the expression levels, while solid red lines indicated the fit of the model over time.

Listing 2: R code

```
# load libraries
library(ragt2ridges)
library(Biobase)

# load and reformat data
data(hpvP53)
Y <- longitudinal2array(t(exprs(hpvP53)))

# search for optimal penalty parameters
optLambdas1 <- optPenaltyVAR1(Y, lambdaMin=c(120, 0.0024), lambdaMax=c(930,
  0.0026), lambdaInit=c(250, 0.0025))

# specify grid for contour
lambdaAgrid <- seq(120, 930, length.out=20)
lambdaPgrid <- sort(c(seq(0.002557, .0026, length.out=100), seq(0.001557,
  .0036, length.out=20)))
L00CVres1 <- loglikL00CVcontourVAR1(lambdaAgrid, lambdaPgrid, Y)

# plot contour
contour(lambdaAgrid, lambdaPgrid, L00CVres1$llL00CV, xlab = "lambdaA", ylab
  = "lambdaP", main = "cross-validated log-likelihood", nlevels=25)
points(optLambdas1[1], optLambdas1[2], pch=20, cex=2, col="red")

# fit VAR(1) model
VAR1hat <- ridgeVAR1(Y=Y, lambdaA=optLambdas1[1], lambdaP=optLambdas1[2])
Ahat <- VAR1hat$A; Phat <- VAR1hat$P
rownames(Ahat) <- colnames(Ahat) <- rownames(Phat) <- colnames(Phat) <-
  rownames(hpvP53)
edgeHeat(Ahat, main="ridge estimate of A")
edgeHeat(Phat, main="ridge precision estimate")

# determine support for A and 0
zerosA <- sparsifyVAR1(A=Ahat, SigmaE=symm(solve(Phat)), threshold="
  localFDR", FDRcut=0.95, statistics=F)$zeros
zerosP <- sparsify(Phat, threshold = "localFDR", FDRcut = 0.95, output="
  light")$zeros

# determine optimal lambda's with inferred support
optLambdas2 <- optPenaltyVAR1(Y, lambdaMin=c(10^(-5), 10^(-5)), lambdaMax=c
  (10, 0.1), lambdaInit=c(5, 0.01), zerosA=zerosA, zerosP=zerosP, zerosAfit
  ="sparse")

# determine contour
lambdaAgrid <- seq(-1, 1, length.out=20) + optLambdas2[1]
lambdaPgrid <- seq(-0.001, 0.001, length.out=20) + optLambdas2[2]
L00CVres2 <- loglikL00CVcontourVAR1(lambdaAgrid, lambdaPgrid, Y, zerosA=
  zerosA, zerosP=zerosP, zerosAfit="sparse")

# plot contour
contour(lambdaAgrid, lambdaPgrid, L00CVres2$llL00CV, xlab = "lambdaA", ylab
  = "lambdaP", main = "cross-validated log-likelihood", nlevels=25)
points(optLambdas2[1], optLambdas2[2], pch=20, cex=2, col="red")
```

```

# re-fit of the VAR(1) model including the prior knowledge
AhatNonsparse <- Ahat
PhatNonsparse <- Phat
VAR1hat <- ridgeVAR1(Y=Y, lambdaA=optLambdas2[1], lambdaP=optLambdas2[2],
  zerosA=zerosA, zerosP=zerosP, zerosAfit="sparse")
Ahat <- VAR1hat$A; Phat <- VAR1hat$P
rownames(Ahat) <- colnames(Ahat) <- rownames(Phat) <- colnames(Phat) <-
  rownames(hpvP53)
edgeHeat(Ahat, main="ridge_re-estimate_of_A_with_inferred_support")
edgeHeat(Phat, main="ridge_precision_re-estimate_with_inferred_support")
edgeHeat(adjacentMat(Ahat), legend=FALSE, main="inferred_support_of_A")
edgeHeat(adjacentMat(Phat), legend=FALSE, main="inferred_support_of_
  precision_matrix")

# graph of the temporal interaction among the genes
graphVAR1(Ahat, Phat, nNames=rownames(Ahat), type="TSCG", vertex.label.cex
  =0.5, vertex.label.font=1, vertex.size=4, vertex.label.color.T0="
  darkblue",
  vertex.label.color.T1="darkblue", vertex.frame.color="steelblue",
  vertex.color.T0="lightblue",
  vertex.color.T1="lightblue", edge.width=1.5, main="")

# node statistics table
stats <- nodeStatsVAR1(Ahat, Phat, as.table=TRUE)
rownames(stats) <- fData(hpvP53)[,1]

```

## Acknowledgement

This research was funded by a grant from the VU University Medical Center-Cancer Center Amsterdam (VUMC-CCA, project CCA2011-5-02).

## References

- Abegaz, F. and Wit, E. (2013). Sparse time series chain graphical models for reconstructing genetic networks. *Biostatistics*, **13**(3), 586–599.
- Carey, V. (2015). *bronchialIL13: time course experiment involving il13*. R package, version 1.3.1.
- Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, **30**(1), 207–210.
- Ernst, J., Nau, G. J., and Bar-Joseph, Z. (2005). Clustering short time series gene expression data. *Bioinformatics*, **21**(Suppl 1), i159–i168.
- Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, **96**(456), 1348–1360.
- Lauritzen, S. L. (1996). *Graphical models*. Oxford University Press.
- Mersmann, O. (2014). *microbenchmark: Accurate Timing Functions*. R package version 1.4-2.
- Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, **27**, 29–34.
- Van Wieringen, W. N. and Peeters, C. F. W. (2014). Ridge estimation of the inverse covariance matrix from high-dimensional data. *under revision: <http://arxiv.org/pdf/1403.0904.pdf>*.