The Author(s) *BMC Bioinformatics* 2016, **17**(Suppl 16):448
DOI 10.1186/s12859-016-1309-x

BMC Bioinformatics

RESEARCH Open Access

# On the inconsistency of $\ell_1$-penalised sparse precision matrix estimation

Otte Heinävaara[1], Janne Leppä-aho[1], Jukka Corander[2,3] and Antti Honkela[1*]

*From* The 10th International Workshop on Machine Learing in Systems Biology (MLSB)
Den Haag, The Netherlands. 3-4 September 2016

## Abstract

**Background:** Various $\ell_1$-penalised estimation methods such as graphical lasso and CLIME are widely used for sparse precision matrix estimation and learning of undirected network structure from data. Many of these methods have been shown to be consistent under various quantitative assumptions about the underlying true covariance matrix. Intuitively, these conditions are related to situations where the penalty term will dominate the optimisation.

**Results:** We explore the consistency of $\ell_1$-based methods for a class of bipartite graphs motivated by the structure of models commonly used for gene regulatory networks. We show that all $\ell_1$-based methods fail dramatically for models with nearly linear dependencies between the variables. We also study the consistency on models derived from real gene expression data and note that the assumptions needed for consistency never hold even for modest sized gene networks and $\ell_1$-based methods also become unreliable in practice for larger networks.

**Conclusions:** Our results demonstrate that $\ell_1$-penalised undirected network structure learning methods are unable to reliably learn many sparse bipartite graph structures, which arise often in gene expression data. Users of such methods should be aware of the consistency criteria of the methods and check if they are likely to be met in their application of interest.

**Keywords:** Gaussian graphical model, Structure learning, Inconsistency, Graphical lasso

## Background

Networks are ubiquitous in biology and inference of network structure from observed data is a common learning task. Many important biological networks have specific structural properties affecting this task. Gene regulatory networks, for instance, are nearly bipartite graphs with a small set of transcription factors regulating all the other genes. This structure has been successfully incorporated in gene regulatory network inference, often assuming a linear dependence between the regulators and targets, in both static (e.g. [1, 2]) as well as dynamic (e.g. [3, 4]) models. These fundamental assumptions form the basis for even very recent successful network inference projects (e.g. [5]).

The simplest and possibly the most widely used generic approaches for network inference are based on estimating the sparse precision matrix, i.e. the inverse covariance matrix, from data. The motivation for the approach stems from the fact that for a Gaussian Markov random field model, zeros in the precision matrix translate exactly to absent edges in the corresponding undirected Gaussian graphical model, thus being informative about the marginal and conditional independence relationships among the variables.

The full $p$-dimensional covariance matrix contains $p(p+1)/2$ parameters, making its accurate estimation from limited data difficult. Additionally, the structure learning requires the inverse of the covariance, and matrix inversion is in general a very fragile operation. To make the problem tractable, some form of regularisation is typically needed. Direct optimisation of the sparse structure would easily lead to very difficult combinatorial optimisation problems. To avoid these computational

*Correspondence: antti.honkela@helsinki.fi
[1] Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Helsinki, Finland
Full list of author information is available at the end of the article

difficulties, several convex $\ell_1$-penalty-based approaches have been proposed. Popular examples include $\ell_1$-penalised maximum likelihood estimation [6], which also forms the basis for the highly popular graphical lasso (glasso) algorithm [7]. $\ell_1$ regularisation has also been used for example in a non-probabilistic alternative with linear-programming-based constrained $\ell_1$ minimisation (CLIME) algorithm [8].

At the heart of the optimisation problems considered by all these methods is a term depending on the $\ell_1$ norm of the estimated precision matrix. $\ell_1$-penalisation-based approaches such as lasso are popular for sparse regression, but they have a known weakness: in addition to promoting sparsity they also push true non-zero elements toward zero [9]. In the context of precision matrix estimation this effect would be expected to be especially strong when some elements of the precision matrix are large, which happens for scaled covariance matrices when the covariance matrix becomes ill-conditioned. This phenomenon occurs frequently under the circumstances where some of the variables are nearly linearly dependent.

In this paper we demonstrate a drastic failure of the $\ell_1$-penalised sparse covariance estimation methods for a class of models that have a bipartite structure where some variables depend linearly on others, such as in the commonly used and very successful gene regulatory network models. For such models even in the limit of infinite data, popular $\ell_1$-penalised methods cannot yield results that are significantly better than based on random guessing on any setting of the regularisation parameter. Yet these models have a very clear sparse structure that becomes obvious from the empirical precision matrix with an increasing $n$. Motivated by our discovery, we also explore the inconsistency of $\ell_1$-penalised methods on models derived from real gene expression data and find the methods poorly suited for such applications.

### Structure learning of Gaussian graphical models

We start with a quick recap on the basics of Gaussian graphical models in order to formulate the problem of structure learning. For a more comprehensive treatment of the subject, we refer to [10, 11]. Let $\mathbf{X} = (X_1, \ldots, X_p)^T$ denote a random vector following a multivariate normal distribution with zero mean and a covariance matrix $\boldsymbol{\Sigma}, \mathbf{X} \sim N_p(\mathbf{0}, \boldsymbol{\Sigma})$. Let $G = (V, E)$ be an undirected graph, where the $V = \{1, \ldots, p\}$ is the set of nodes and $E \subset V \times V$ stands for the set of edges. The nodes in the graph represent the random variables in the vector $\mathbf{X}$ and absences of the edges in the graph correspond conditional independence assertions between these variables. More in detail, we have that $(i, j) \notin E$ and $(j, i) \notin E$ if and only if $X_i$ is conditionally independent of $X_j$ given the remaining variables in $\mathbf{X}$.

In the multivariate normal setting, there is a one-to-one correspondence between the missing edges in the graph and the off-diagonal zeros of the precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, that is, $\omega_{ij} = 0 \Leftrightarrow X_i \perp\!\!\!\perp X_j \,|\mathbf{X} \setminus \{X_i, X_j\}$ (see, for instance, [11], p. 129). Given an undirected graph $G$, a Gaussian graphical model is defined as the collection of multivariate normal distributions for $\mathbf{X}$ satisfying the conditional independence assertions implied by the graph $G$.

Assume we have a complete (no missing observations) i.i.d. sample $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$ from the distribution $N_p(\mathbf{0}, \Sigma)$. Based on the sample $\mathbf{x}$, our goal in structure learning is to find the graph $G$, or equivalently, learn the zero-pattern of $\Omega$. The usual assumption is that the underlying graph is sparse. A naive estimate for $\Omega$ by inverting the sample covariance matrix is practically never truly sparse for any real data. Furthermore, if $n < p$ the sample covariance matrix is rank-deficient and thus not even invertible.

One common approach to overcome these problems is to impose an additional $\ell_1$-penalty on the elements of $\boldsymbol{\Omega}$ when estimating it. This kind of regularisation effectively forces some of the elements of $\boldsymbol{\Omega}$ to zero, thus resulting in sparse solutions. In the context of regression models, this method applied on the regression coefficients goes by the name of *lasso* [12]. There exists a wide variety of methods making use of $\ell_1$-regularisation in the setting of Gaussian graphical model structure learning [6–8, 13–16].

## Methods

### $\ell_1$-regularised methods for Gaussian graphical model structure learning

In this section we provide a brief review of selected examples of different types of $\ell_1$-penalised methods.

#### *Glasso*

We begin with the widely used graphical lasso-algorithm (glasso) [7]. Glasso-method maximises an objective function consisting of the Gaussian log-likelihood and an $\ell_1$ penalty:

$$\log\det(\Omega) - \operatorname{trace}(\Omega\mathbf{S}) - \lambda|\Omega|_1, \qquad (1)$$

where $\mathbf{S}$ denotes the sample covariance matrix and $\lambda > 0$ is the regularisation parameter controlling the sparsity of the solution. The $\ell_1$ penalty, $|\boldsymbol{\Omega}|_1 = \sum_{i,j} |\omega_{ij}|$, is applied on all the elements of $\boldsymbol{\Omega}$, but the variant where the diagonal elements are omitted is also common. (We use the notation $|\cdot|_p$ for the vector norm over matrix elements to avoid confusion with the matrix norm $\|\cdot\|$). The objective function (1) is maximised over all positive definite matrices $\Omega$ and the optimisation is carried out in practice using block-wise coordinate descent.

#### *CLIME*

The CLIME method (Constrained $\ell_1$ minimisation for Inverse Matrix Estimation) [8] approaches the problem of sparse precision matrix estimation from a slightly different perspective. It seeks matrices $\mathbf{\Omega}$ with a minimal $\ell_1$ norm under the constraint

$$|\mathbf{S\Omega} - \mathbf{I}|_\infty \leq \lambda, \tag{2}$$

where $\lambda$ is the tuning parameter and $|\mathbf{A}|_\infty = \max_{i,j} |a_{ij}|$ is the element-wise maximum. The optimisation problem $\min_{\mathbf{\Omega}} |\mathbf{\Omega}|_1$ subject to the constraint (2) does not explicitly force the solution to be symmetric, which is resolved by picking from estimated values $\omega_{ij}$ and $\omega_{ji}$ the one with a smaller magnitude into the final solution. In practice, the optimisation problem is decomposed over variables into $p$ sub-problems which are then efficiently solved using linear programming.

#### *SCIO*

The recently introduced Sparse Column-wise Inverse Operator (SCIO) [17] method decomposes the estimation of $\mathbf{\Omega}$ into the following smaller problems

$$\min_{\boldsymbol{\beta}_i \in \mathbb{R}^p} \left\{ \frac{1}{2} \boldsymbol{\beta}_i^T \mathbf{S} \boldsymbol{\beta}_i - \mathbf{e}_i^T \boldsymbol{\beta}_i + \lambda |\boldsymbol{\beta}_i|_1 \right\},$$

where $\mathbf{S}$ and $\lambda$ are defined as before and $\mathbf{e}_i$ is an $i$:th standard unit vector. The regularisation parameter $\lambda$ can in general vary with $i$ but this is omitted in our notation. The solutions $\hat{\boldsymbol{\beta}}_i$ form the columns for the estimate of $\mathbf{\Omega}$. SCIO does not guarantee the symmetry of the resulting precision matrix, which is resolved as in the case of CLIME.

### Alternative methods

#### *The naive approach*

In addition to the above-mentioned $\ell_1$-penalised methods, we consider two alternative approaches. In a "naive" approach, we simply take the sample covariance matrix, invert it, and then threshold the resulting matrix to obtain a sparse estimate for the precision matrix. The threshold value is chosen using the ground truth graph so that the naive estimator will have as many non-zero entries as there are edges in the true graph. Setting the threshold value according to the ground truth is of course unrealistic, however, it is nevertheless interesting to compare the accuracy of this simple procedure to the performance of the more refined $\ell_1$ methods, when also their tuning parameters are chosen in a similar fashion.

#### *FMPL*

Lastly, we consider an approximate Bayesian approach which is based on finding a graph with the highest fractional marginal pseudo-likelihood (FMPL) [18]. Seeking the graph that maximises the marginal likelihood is equivalent with finding the maximum a posteriori graph, assuming a uniform prior over different graphs. However, computing the marginal likelihood is computationally challenging for a general graph, even in the Gaussian setting with conjugate priors. The FMPL method aims at circumventing this problem by replacing the true likelihood in the marginal likelihood with pseudo-likelihood. This leads to a convenient factorisation of marginal likelihood over variables and the resulting expression can be evaluated in closed form using previous results regarding objective comparison of Gaussian directed acyclic graphs [19, 20]. In practice, the factorisation allows the method to identify optimal Markov blankets independently for each of the variables using a greedy hill-climbing algorithm. The found Markov blankets are then combined into a proper undirected graph using any of the three different schemes commonly employed in graphical model learning: OR, AND and greedy hill-climbing (HC) [21].

### Model selection consistency

The assumptions required for a consistent model selection with an $\ell_1$-penalised Gaussian log-likelihood have been studied, for instance, in [22]. The authors provide a number of conditions in the multivariate normal model that are sufficient for the recovery of the zero pattern of the true precision matrix $\mathbf{\Omega}^*$ with a high probability when the sample size is large. For our purposes, the most relevant condition is the following:

**Assumption 1** *There exists* $\alpha \in (0, 1]$, *such that*

$$\gamma := \|\Gamma_{S^C S}(\Gamma_{SS})^{-1}\|_\infty \leq 1 - \alpha. \tag{3}$$

Here $S \subset V \times V$ is a set defining the support of $\mathbf{\Omega}^*$, that is, the non-zero elements of $\mathbf{\Omega}^*$ (diagonal and the elements corresponding to the edges in the graphical model) and $S^C$ refers to the complement of $S$ in $V \times V$. The $\Gamma$ term is defined via Kronecker product $\otimes$ as $\Gamma = (\mathbf{\Omega}^*)^{-1} \otimes (\mathbf{\Omega}^*)^{-1} \in \mathbb{R}^{p^2 \times p^2}$ and $\Gamma_{AB}$ refers to the specific rows and columns of $\Gamma$ indexed by $A \subset V \times V$ and $B \subset V \times V$, respectively. The norm in the equation is defined as $\|A\|_\infty = \max_j \sum_i |a_{ij}|$.

The above result applies to glasso. However, a quite similar result was presented for SCIO in [17]:

**Assumption 2** *There exists* $\alpha \in (0, 1)$, *such that*

$$\max_{1 \leq i \leq p} \|\mathbf{\Sigma}^*_{\mathbf{s}_i^C \mathbf{s}_i} \left(\mathbf{\Sigma}^*_{\mathbf{s}_i \mathbf{s}_i}\right)^{-1}\|_\infty \leq 1 - \alpha.$$

Here $\mathbf{\Sigma}^* = (\mathbf{\Omega}^*)^{-1}$ and $\mathbf{s}_i = \{j \in \{1, \ldots, p\} \mid (\mathbf{\Omega}^*)_{ij} \neq 0\}$. Assumption 2 under the multivariate normality guarantees that the support of $\mathbf{\Omega}^*$ is recovered by SCIO with a high probability as the sample size gets large.

### Bipartite graphs inducing inconsistency with $\ell_1$ penalisation

Methods for sparse precision matrix estimation generally depend on an objective function (such as log-likelihood) and a penalty function or regulariser, which in a Bayesian setting is usually represented by the prior. The ideal penalty function for many problems would be the $\ell_0$ "norm" counting the number of non-zero elements: $|x|_0 = \#\{i|x_i \neq 0\}$. This $\ell_0$ function is not a proper norm, but it provides a very intuitive notion of sparsity. The main problem with its use is computational: using $\ell_0$-penalisation leads to very difficult non-convex combinatorial optimisation problems. The most common approach to avoid the computational challenges is to use $\ell_1$ penalisation as a convex relaxation of $\ell_0$. As mentioned above this works well in many cases but it comes with a price, since in addition to providing the sparsity, $\ell_1$ also regularises large non-zero values. Depending on the problem, as we demonstrate here, this effect can be substantial and may cause $\ell_1$-regularised methods to return totally meaningless results.

Intuitively, $\ell_1$-regularised methods are expected to fail when some elements of the true precision matrix become so large that their contribution to the penalty completely overwhelms the other parts of the objective and the penalty. One example where this happens is when some set of variables depends linearly on another set of variables. In such situation the covariance matrix can become ill-conditioned and the elements of its inverse, the precision matrix, grow. One example of when this happens is models with a linear latent variable structure.

Let us consider a model for $\mathbf{x} \in \mathbb{R}^{d_1}, \mathbf{y} \in \mathbb{R}^{d_2}$, where $\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon$. The graphical structure of the model and the corresponding precision matrix structure are illustrated in Fig. 1. Assuming $\mathbf{x} \sim \mathcal{N}(0, \sigma_x^2 I), \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2 I)$, the covariance of the concatenated vectors $\left(\mathbf{x}^T, \mathbf{y}^T\right)^T$ is given by the block matrix

$$\mathrm{Cov}\left(\left(\mathbf{x}^T, \mathbf{y}^T\right)^T\right) = \mathbf{C} = \sigma_x^2 \begin{pmatrix} I & \mathbf{A}^T \\ \mathbf{A} & \mathbf{A}\mathbf{A}^T + \sigma_\epsilon^2 I \end{pmatrix}. \quad (4)$$

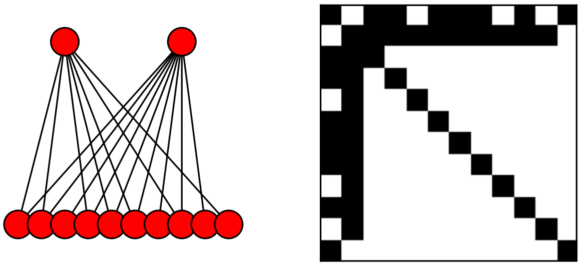

**Fig. 1** *Left*: Graphical representation of a latent variable model as an undirected graphical model for a case with somewhat sparse **A**. *Right*: The adjacency matrix of the graph showing the sparse pattern of non-zero elements in the corresponding precision matrix

The covariance matrix has an analytic block matrix inverse [23]

$$\mathbf{C}^{-1} = \sigma_x^{-2} \begin{pmatrix} I + \sigma_\epsilon^{-2}\mathbf{A}^T\mathbf{A} & -\sigma_\epsilon^{-2}\mathbf{A}^T \\ -\sigma_\epsilon^{-2}\mathbf{A} & \sigma_\epsilon^{-2} I \end{pmatrix}. \quad (5)$$

This precision matrix recapitulates the conditional independence result for Gaussian Markov random fields: the lower right block is diagonal because the variables in $\mathbf{y}$ are conditionally independent of each other given $\mathbf{x}$. The matrix is clearly sparse, so we would intuitively assume sparse precision matrix estimation methods should be able to recover it. The non-zero elements do, however, depend on $\sigma_\epsilon^{-2}$ which can make them very large if the noise $\sigma_\epsilon^2$ is small.

It is possible to evaluate and bound the different terms of Eq. (1) evaluated at the ground truth for these models:

$$\log\det(\mathbf{C}^{-1}) = -(d_1 + d_2)\log\sigma_x^2 - d_2 \log\sigma_\epsilon^2 \quad (6)$$

$$-\mathrm{trace}(\mathbf{C}\mathbf{C}^{-1}) = -(d_1 + d_2) \quad (7)$$

$$-\lambda|\mathbf{C}^{-1}|_1 < -\lambda\sigma_x^{-2}\sigma_\epsilon^{-2}(d_2 + 2|\mathbf{A}|_1). \quad (8)$$

The magnitude of the penalty term (8) clearly grows very quickly as $\sigma_\epsilon^2$ decreases while the magnitudes of the two first log-likelihood terms (6) and (7) grow much more slowly as they only depend on $\log\sigma_\epsilon^2$. Thus the total value of Eq. (1) decreases without bound as $\sigma_\epsilon^2$ decreases.

Ignoring the ground truth, it is easy to see that one can construct an estimate $\Omega$ for which the objective remains bounded. If we assume $|\mathbf{C}|_\infty = \max|c_{ij}| \leq 1$ (after normalisation), then

$$\mathrm{trace}(\mathbf{C}\Omega) \leq |\Omega|_1.$$

As the other terms only depend on $\Omega$ it is easy to choose $\Omega$ so that they remain bounded. The estimate $\Omega$ that yields these values will in many cases not have anything to do with $\mathbf{C}^{-1}$, as seen in the experiments below.

Here Eq. (6) follows from the block matrix determinant identity [24]

$$\det\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} = \det(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})\det(\mathbf{D}),$$

while Eq. (8) is based on a lower bound of the $\ell_1$ norm as the sum over all except the top-left block of the block matrix in Eq. (5).

## Results

### Synthetic example

We tested the performance of glasso, SCIO and CLIME as well as FMPL using the model structure corresponding to the bipartite graph introduced above. The performance of the methods was investigated by varying the noise variance $\sigma_\epsilon^2$, and the sample size $n$. The model matrix $\mathbf{A}$ was created as a $(d_2, d_1)$-array of independent normal random variables with mean 0 and variance 1. The majority of the tests were run using input dimensionality $d_1 = 2$, output dimensionality $d_2 = 10$ and noise variance $\sigma_\epsilon^2 = 0.1^2$ but we also tested varying these settings. For each individual choice of noise and sample size, $k = 50$ different matrices $\mathbf{A}$ were generated and the results were averaged.

Generating $n$ samples using model described, data were normalised and analysed using the five different methods. We calibrated the methods in a way that number of edges in the resulting graph would match the true number. Similarly, we thresholded the naive method by taking inverse matrix directly to output the correct number of edges. The FMPL method has no direct tuning parameters so we used its OR mode results as such. Similar tuning is not possible in a real problem where the true number of edges is now known. The tuning represents the best possible results the methods could obtain with an oracle that provides an optimal regularisation parameter.

We evaluated the results using the Hamming distance between the ground truth and the inferred sparsity pattern, i.e. the number of incorrect edges and non-edges which were treated symmetrically. For methods returning the correct number of edges, this value is directly related to the precision $pr$ through

$$d_{\text{Hamming}} = 2(1 - pr)N_{\text{true positives}}$$

or conversely

$$pr = 1 - \frac{d_{\text{Hamming}}}{2N_{\text{true positives}}}.$$

We will nevertheless use the Hamming distance as it enables fair comparison with FMPL that sometimes returns a different number of edges.

Figures 2 and 3 show the Hamming distance obtained by the different methods as a function of the noise level when using 100 and 1000 samples, respectively. The results show that especially for low but also for high noise levels, the $\ell_1$-based methods all perform very poorly with especially glasso and CLIME performing very close to random guessing level for low noise levels $\sigma_\epsilon \leq 0.1$. The naive inverse and FMPL work much better up to moderate noise levels of $\sigma_\epsilon \approx 2$ after which the noise starts to dominate the signal and the performance of all methods starts to drop. SCIO is a little better than the other $\ell_1$-based methods but clearly worse than FMPL and naive in the low noise regime.

Figure 4 shows the results when changing the output dimensionality $d_2$ from 10. The results show that the performance of all $\ell_1$-based methods is very poor across all $d_2$. Glasso performance is close to random guessing level across the entire range considered, while CLIME is slightly better for $d_2 \geq 18$ and SCIO slightly better across the entire range. Both FMPL and naive are significantly better than any of the $\ell_1$-based methods.

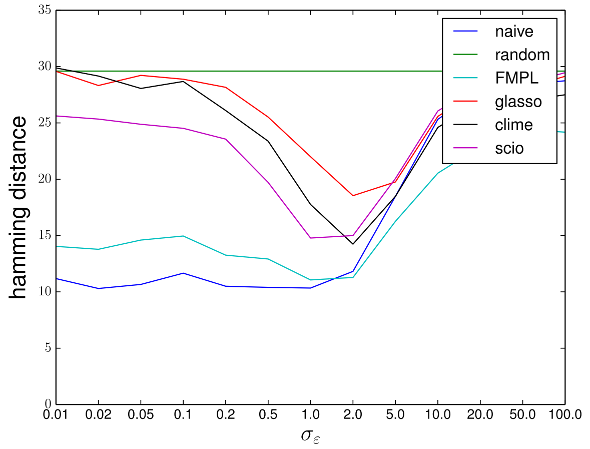

**Fig. 2** Performances of different methods on the bipartite graph model with 100 samples. (Lower values are better)

Figure 5 shows the corresponding result when changing the input dimensionality $d_1$. The results are now quite different as all methods are better than random especially for larger values. SCIO still outperforms CLIME which outperforms glasso. FMPL is really accurate for small $d_1$ but degrades for larger $d_1$ while the naive method is the most accurate in almost all cases.

To further illustrate the behaviour of glasso on these examples, Fig. 6 shows the contributions of the different parts of the glasso objective function (1) as a function of the noise level both for the true solution ("truth") as well as the glasso solution. The results show that for low noise levels the penalty incurred by the true solution becomes massive. The glasso solution has a much lower log-likelihood ("logl") than ground truth but this is amply compensated by the significantly smaller penalty. As the

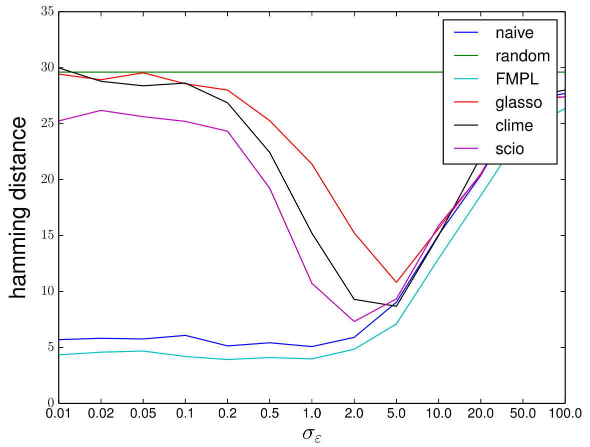

**Fig. 3** Performances of different methods on the bipartite graph model with 1000 samples. (Lower values are better)

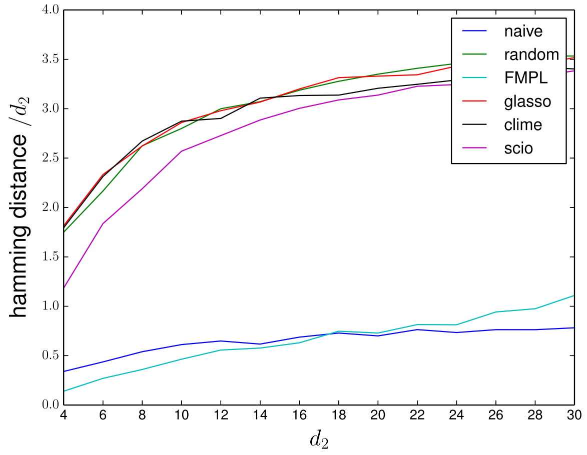

**Fig. 4** Performances of different methods on the bipartite graph model with varying output dimensionality. (Lower values are better)

noise increases, the penalty of the true solution decreases and the glasso solution converges to similar values.

### Necessity of assumption 1

It can be checked that the norm $\gamma$ in Assumption 1 and Eq. (3) for bipartite graph models presented above depends on the scale of **A**. We took advantage of this by creating examples with different values of $\gamma$ and testing the precision of glasso using the true covariance which corresponds to infinite data limit. The results of this experiment are shown in Fig. 7. The results verify that glasso consistently yields perfect results when $\gamma < 1$ which is a part of the sufficient conditions for consistency of glasso. As $\gamma$ grows and the sufficient conditions are no longer satisfied, it is clearly seen that the accuracy of glasso starts to deteriorate rapidly. This suggests that the sufficient condition of Assumption 1 is in practice also necessary to ensure consistence.

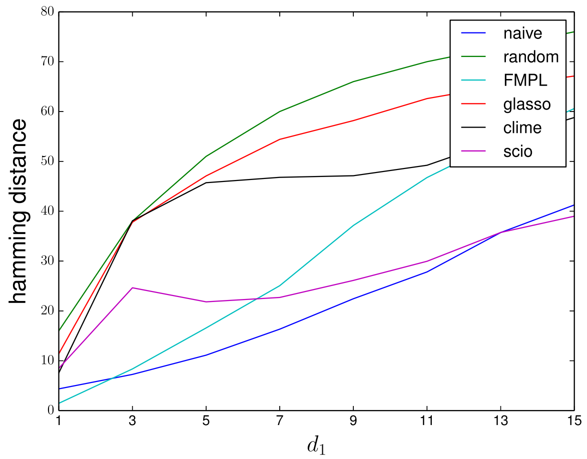

**Fig. 5** Performances of different methods on the bipartite graph model with varying input dimensionality. (Lower values are better)

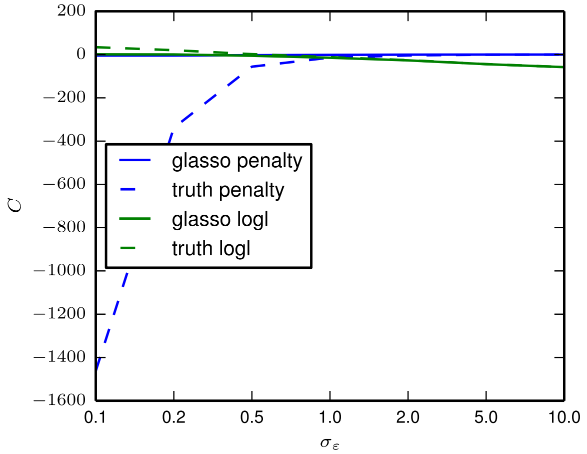

**Fig. 6** Contributions of the different terms of the glasso objective (1) for the bipartite graph model with 1000 samples. The *green curves* show the contributions of the first two terms of Eq. (1) and the *blue curves* show the contributions of the last penalty term. *Solid lines* show the result of the glasso optimal solution while *dashed lines* show the result for the true solution

### Inconsistency for models of real gene expression data

We tested how often the problems presented above appear in real data using the "TCGA breast invasive carcinoma (BRCA) gene expression by RNAseq (IlluminaHiSeq)" data set [25] downloaded from https://genome-cancer.ucsc.edu/proj/site/hgHeatmap/. The data set contains gene expression measurements for 20530 genes for $n = 1215$ samples. After removing genes with a constant expression across all samples there are $p = 20252$ genes remaining.

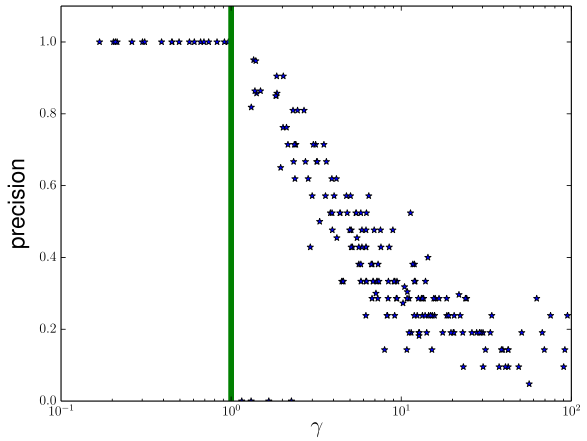

**Fig. 7** Precision of glasso on infinite data as a function of the norm $\gamma$ of Assumption 1 and Eq. (3). Values to the *left* of the *green vertical line* satisfy this condition while values to the *right* violate it. (Higher values are better)

In order to test the methods we randomly sampled subsets of $d$ genes and considered the correlation matrix $\mathbf{C}_0$ over that subset. We generated sparse models with known ground truth by computing the corresponding precision matrix $\mathbf{\Lambda}_0$ from the empirical correlation matrix, setting elements with absolute values below chosen cutoff $\delta = 0.1$ to 0 to obtain

$$\mathbf{\Lambda}_{ij} = \begin{cases} (\mathbf{\Lambda}_0)_{ij} & \text{if } |(\mathbf{\Lambda}_0)_{ij}| > \delta \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and the testing covariance matrix $\mathbf{C} = \mathbf{\Lambda}^{-1}$. The cutoff lead to networks that were sparse with on average 60 % zeros in the precision matrix.

Figure 8 shows the fraction of covariances derived from random subsets of $d$ genes that satisfy the Assumption 1 of [22] ($c = 1$) as well as the fraction of values below more relaxed bounds. The figure shows that the assumption is reliably satisfied only for very small $d$ while for $d \geq 20$, the assumption is essentially never satisfied. Based on the results of Fig. 7 it is likely that glasso results will degrade significantly for $\gamma > 10$ and beyond which are very common for large networks.

We further studied how accurately glasso can recover the graphical structures when the data were generated using the precision matrices described above. We used a similar thresholding with a cut-off value of 0.1 in order to first form sparse precision matrices for a random subset of genes with given dimension. These matrices were then inverted to obtain covariance matrices. We checked that the resulting matrices were positive definite and then used them to sample multivariate normal data with zero mean with different sample sizes.

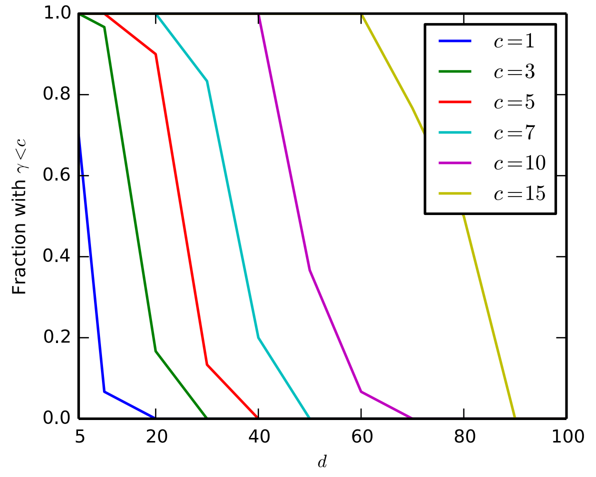

**Fig. 8** Testing the condition of Assumption 1 of [22] in Eq. (3) on real gene expression data showing the fraction of random subsets of $d$ genes that fulfil the requirement and various relaxations. The condition (3) requires $\gamma < 1$, but the figure shows results also for larger $\gamma$ cutoffs, denoted by $c$

The obtained data sets were centred and scaled before computing the sample covariance which was used as input to the glasso algorithm. The regularisation parameter was chosen with the aid of the ground truth graph, so that the the graph identified by glasso would contain as many edges as there were in the real graph. Results are shown in Fig. 9. The results show that glasso performance decreases as the network size increases and is approaching that of random guessing for the largest networks considered here.

Figure 10 shows the contributions of different parts of the glasso objective function (1) as a function of the number of genes $d$. The regularisation parameter $\lambda$ of glasso was tuned to return a solution with the same number of edges as in the true solution. We used the glasso implementation of scikit-learn [26], which ignores the diagonal terms of $\mathbf{\Omega}$ when computing the penalty. The figure shows clearly how the penalty term for the true solution increases superlinearly as a function of $d$. (A linear increase would correspond to a horizontal line.) The result is even more striking given that the optimal $\lambda$ decreases slightly as $d$ increases. The penalty contribution for glasso solution increases much more slowly. The excess loss in log-likelihood from glasso solution increases as $d$ increases, but this is compensated by a larger saving in the penalty. Together these suggest that glasso solutions are likely to remain further away from ground truth as $d$ increases.

## Discussion

The class of models with bipartite graphs presented above is an interesting example of models that have a

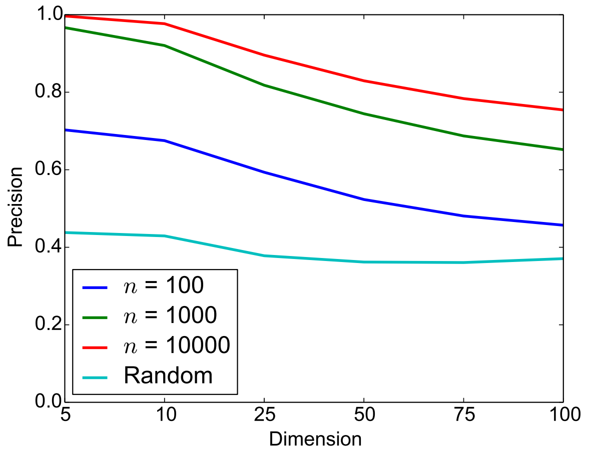

**Fig. 9** Average precisions for glasso with different dimensions and sample sizes of the real gene expression data, higher values are better. In these experiments, 50 data sets were created. We encountered convergence problems with few of the data sets and the corresponding results were omitted when computing the average values shown here. The precision obtained by random guessing is also illustrated

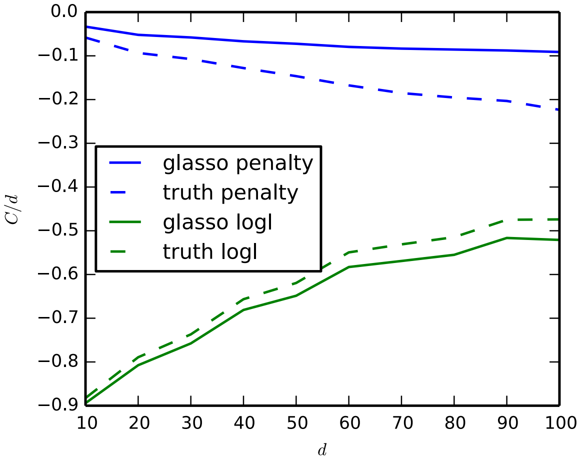

**Fig. 10** Average contributions of the different terms of the glasso objective function (1) on real gene expression data over random subsets of *d* genes. The values are shown for the $\ell_1$ penalty term as well as the unnormalised log-likelihood, divided by *d* to make them comparable. *Solid lines* show the values for glasso result while *dashed lines* show the result for ground truth

very clear sparse structure, which all $\ell_1$-penalisation-based methods seem unable to recover even in the limit of infinite data. This class complements the previously considered examples of models where glasso is inconsistent including the "two neighbouring triangles" model of [27] and the star graph of [22], the latter of which can be seen as a simple special case of our example.

An important question arising from our investigation is how significant the discovered limitation to inferring sparse covariance matrices is in practice, i.e. how common are the (nearly) bipartite structures in real data sets. Given the popularity and success of linear models in diverse applications it seems plausible such structures could often exist in real data sets, either as an intrinsic property or as a result of some human intervention, e.g. through inclusion of partly redundant variables.

The gene expression data set is a natural example of an application where graphical model structure learning has been considered. The original glasso paper [7] contained an example on learning gene networks, although from proteomics data. Other authors (e.g. [28]) have applied Gaussian graphical models and even glasso (e.g. [29]) to gene network inference from expression data. Our experiments on the TCGA gene expression data suggest that in such applications it is advisable to consider the conditions for the consistency of $\ell_1$-penalised methods very carefully when planning to apply those.

Previous publications presenting new methods for sparse precision matrix have typically tested the method on synthetic examples where the true precision matrix is specified to contain mostly small values. Specifying the precision matrix provides a convenient way to generate test cases as the sparsity pattern can be defined very naturally through it. At the same time, this excludes any models that have an ill-conditioned covariance. As shown by our example, such ill-conditioned covariances arise very naturally from model structures that are plausible from the application perspective. The example presented in this paper thus represents a very useful additional test case for method developers and benchmarkers.

## Conclusions

Our results strongly suggest that users of the numerous $\ell_1$-penalised and other $\ell_1$-based sparse precision matrix and Gaussian graphical model structure learning methods should be very careful about checking whether the conditions of consistency for precision matrix estimation are likely to be fulfilled in the application area of interest. The consistency conditions are typically presented in a form which requires knowing the ground truth which makes it difficult to test them directly. Developing alternative criteria that can be checked more easily in practice would be an important avenue of future research for these methods.

**Acknowledgments**
We would like to thank the authors of the FMPL method for sharing their code.

**Declarations**
This article has been published as part of *BMC Bioinformatics* Volume 17 Supplement 16, 2016: Proceedings of the Tenth International Workshop on Machine Learning in Systems Biology (MLSB 2016). The full contents of the supplement are available online at http://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-17-supplement-16.

**Funding**
This work was supported by the Academy of Finland [259440 to A.H., 251170 to J.C.] and the European Research Council [239784 to J.C.].

### Availability of data and materials

Python code used in the experiments is available at https://github.com/PROBIC/l1-inconsistency. Matlab code for FMPL is available from the FMPL authors upon request.

### Authors' contributions

OH implemented and performed the consistency criterion analyses and experiments on synthetic data. JL performed the FMPL analyses and took part in analysing the gene expression data. AH conceived the approach and performed the gene expression data analyses. OH, JL, JC and AH interpreted the results and wrote the manuscript. All authors read and approved the final manuscript.

### Competing interests

The authors declare that they have no competing interests.

### Consent for publication

Not applicable.

### Ethics approval and consent to participate

Not applicable.

**Author details**
[1]Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Helsinki, Finland. [2]Helsinki Institute for Information Technology HIIT, Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland. [3]Department of Biostatistics, University of Oslo, Oslo, Norway.

Published: 13 December 2016

## References

1. Holter NS, Mitra M, Maritan A, Cieplak M, Banavar JR, Fedoroff NV. Fundamental patterns underlying gene expression profiles: simplicity from complexity. Proc Natl Acad Sci U S A. 2000;97(15):8409–14. doi:10.1073/pnas.150242097.
2. Liao JC, Boscolo R, Yang YL, Tran LM, Sabatti C, Roychowdhury VP. Network component analysis: reconstruction of regulatory signals in biological systems. Proc Natl Acad Sci U S A. 2003;100(26):15522–7. doi:10.1073/pnas.2136632100.
3. Sabatti C, James GM. Bayesian sparse hidden components analysis for transcription regulation networks. Bioinformatics. 2006;22(6):739–46. doi:10.1093/bioinformatics/btk017.
4. Sanguinetti G, Lawrence ND, Rattray M. Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities. Bioinformatics. 2006;22(22):2775–81. doi:10.1093/bioinformatics/btl473.
5. Arrieta-Ortiz ML, Hafemeister C, Bate AR, Chu T, Greenfield A, Shuster B, Barry SN, Gallitto M, Liu B, Kacmarczyk T, Santoriello F, Chen J, Rodrigues CDA, Sato T, Rudner DZ, Driks A, Bonneau R, Eichenberger P. An experimentally supported model of the Bacillus subtilis global transcriptional regulatory network. Mol Syst Biol. 2015;11(11):839.
6. Meinshausen N, Bühlmann P. High-dimensional graphs and variable selection with the lasso. Ann Stat. 2006;34(3):1436–62. doi:10.1214/009053606000000281.
7. Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics. 2008;9(3):432–41. doi:10.1093/biostatistics/kxm045.
8. Cai T, Liu W, Luo X. A constrained $\ell_1$ minimization approach to sparse precision matrix estimation. J Am Stat Assoc. 2011;106(494):594–607. doi:10.1198/jasa.2011.tm10155.
9. Zhao P, Yu B. On model selection consistency of lasso. J Mach Learn Res. 2006;7:2541–63.
10. Whittaker J. Graphical Models in Applied Multivariate Statistics. Chichester: John Wiley & Sons; 1990.
11. Lauritzen SL. Graphical Models. Oxford: Oxford University Press; 1996.
12. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B. 1996;58:267–88.
13. Yuan M, Lin Y. Model selection and estimation in the Gaussian graphical model. Biometrika. 2007;94(1):19–35.
14. Banerjee O, El Ghaoui L, d'Aspremont A. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. J Mach Learn Res. 2008;9:485–516.
15. Peng J, Wang P, Zhou N, Zhu J. Partial correlation estimation by joint sparse regression models. J Am Stat Assoc. 2009;104(486):735–46. doi:10.1198/jasa.2009.0126.
16. Hsieh C, Sustik MA, Dhillon IS, Ravikumar PD. QUIC: quadratic approximation for sparse inverse covariance estimation. J Mach Learn Res. 2014;15(1):2911–47.
17. Liu W, Luo X. Fast and adaptive sparse precision matrix estimation in high dimensions. J Multivar Anal. 2015;135:153–62. doi:10.1016/j.jmva.2014.11.005.
18. Leppä-aho J, Pensar J, Roos T, Corander J. Learning Gaussian graphical models with fractional marginal pseudo-likelihood. arXiv:1602.07863 [stat.ML]. 2016.
19. Geiger D, Heckerman D. Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. Ann Stat. 2002;30(5):1412–40. doi:10.1214/aos/1035844981.
20. Consonni G, Rocca LL. Objective Bayes factors for Gaussian directed acyclic graphical models. Scand J Stat. 2012;39(4):743–56. doi:10.1111/j.1467-9469.2011.00785.x.
21. Pensar J, Nyman H, Niiranen J, Corander J. Marginal pseudo-likelihood learning of discrete Markov network structures. Bayesian Anal. doi:10.1214/16-BA1032.
22. Ravikumar P, Wainwright MJ, Raskutti G, Yu B. High-dimensional covariance estimation by minimizing $\ell_1$-penalized log-determinant divergence. Electron J Stat. 2011;5:935–80. doi:10.1214/11-ejs631.
23. Lu TT, Shiou SH. Inverses of $2 \times 2$ block matrices. Comput Math Appl. 2002;43(1-2):119–29. doi:10.1016/s0898-1221(01)00278-4.
24. Powell PD. Calculating determinants of block matrices. 2011. arXiv:1112.4379 [math.RA].
25. Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. Nature. 2012;490(7418):61–70. doi:10.1038/nature11412.
26. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
27. Meinshausen N. A note on the Lasso for Gaussian graphical model selection. Stat Probab Lett. 2008;78(7):880–4. doi:10.1016/j.spl.2007.09.014.
28. Ma S, Gong Q, Bohnert HJ. An Arabidopsis gene network based on the graphical Gaussian model. Genome Res. 2007;17(11):1614–25. doi:10.1101/gr.6911207.
29. Menéndez P, Kourmpetis YAI, ter Braak CJF, van Eeuwijk FA. Gene regulatory networks from multifactorial perturbations using Graphical Lasso: application to the DREAM4 challenge. PLoS One. 2010;5(12):14147. doi:10.1371/journal.pone.0014147.