BMC
Systems Biology

RESEARCH   **Open Access**

A group LASSO-based method for robustly inferring gene regulatory networks from multiple time-course datasets

Li-Zhi Liu[1], Fang-Xiang Wu[1,2]*, Wen-Jun Zhang[1,2]

Abstract

Background: As an abstract mapping of the gene regulations in the cell, gene regulatory network is important to both biological research study and practical applications. The reverse engineering of gene regulatory networks from microarray gene expression data is a challenging research problem in systems biology. With the development of biological technologies, multiple time-course gene expression datasets might be collected for a specific gene network under different circumstances. The inference of a gene regulatory network can be improved by integrating these multiple datasets. It is also known that gene expression data may be contaminated with large errors or outliers, which may affect the inference results.

Results: A novel method, Huber group LASSO, is proposed to infer the same underlying network topology from multiple time-course gene expression datasets as well as to take the robustness to large error or outliers into account. To solve the optimization problem involved in the proposed method, an efficient algorithm which combines the ideas of auxiliary function minimization and block descent is developed. A stability selection method is adapted to our method to find a network topology consisting of edges with scores. The proposed method is applied to both simulation datasets and real experimental datasets. It shows that Huber group LASSO outperforms the group LASSO in terms of both areas under receiver operating characteristic curves and areas under the precision-recall curves.

Conclusions: The convergence analysis of the algorithm theoretically shows that the sequence generated from the algorithm converges to the optimal solution of the problem. The simulation and real data examples demonstrate the effectiveness of the Huber group LASSO in integrating multiple time-course gene expression datasets and improving the resistance to large errors or outliers.

Background

A Gene regulatory network (GRN) consists of a set of genes and regulatory relationships among them. Tremendous amount of microarray data that measure expression levels of genes under specific conditions are obtained from experiments. It is a challenging problem in systems biology to reconstruct or "reverse engineer" GRNs by aiming at retrieving the underlying interaction relationships between genes from microarray data. Various approaches have been developed to infer GRNs from microarray data. Most of them can be classified into two categories: parametric or model-based methods and nonparametric or dependency-measure-based methods. Commonly used models include ordinary differential equations [1], Gaussian graphical models [2] and Bayesian networks [3]. Dependency measures include partial correlation coefficient [4], mutual information [5], and z-score [6].

* Correspondence: faw341@mail.usask.ca
[1]Department of Mechanical Engineering, University of Saskatchewan, 57 Campus Drive, S7N 5A9 Saskatoon, Canada
Full list of author information is available at the end of the article

The reconstruction of GRN is a non-trivial problem. On the one hand, the number of possible network topologies grows exponentially as the number of genes increases. On the other hand, the information in the microarray data is quite limited. The data contain a lot of inherent noises generated from the devices or the experiment processes. For large-scale networks, the number of observations is usually much less than that of genes, also known as "dimensionality problem" [2,7]. The lack of observations and the high dimensionality of the data prohibit the direct application of traditional methods and make the inference task extremely challenging.

As more and more microarray datasets on the same species are produced from different laboratories, their integration leads to more robust and more reliable results. The methods that integrate multiple datasets could synergize the strength of each dataset and either infer a more accurate network if all the integrated datasets are in high qualities or infer a robust network which is better than the worse one that is from a single dataset. However, multiple time-course datasets can not be simply combined as one dataset as there is no temporal dependencies between the datasets. Wang et al. [8] proposes a linear programming framework to integrate multiple time-course gene expression data to infer a network topology that is most consistent to all datasets. In their method, the regulatory strengths between genes is assumed to be the same across all datasets. However, different datasets may be produced under different circumstances, which may result in different regulatory strength between genes. Another problem is that the value of the tuning parameter in their method, which controls the degree of sparsity of the inferred network, is only determined intuitively. Chen et al. [9] infer one GRN from each time-course data separately, and combine edges of inferred GRNs using a strategy similar to majority vote. For this method, using each dataset separately in the inference process may miss the opportunity of taking advantage of information in other datasets and the tuning parameter is also determined intuitively.

This study focuses on inferring the topologies of GRNs from multiple time-course gene expression datasets based on an autoregressive model. We assume that one GRN corresponds to one dataset and these GRNs share the same topology across all datasets. By assigning the parameters representing the regulatory strengths of the same edge into the one group, the group LASSO [10] can be applied to find the sparse network topology. Microarray data typically contain noises and outliers, which could severely affect the quality of inferred results. Rosset and Zhu [11] proposes a robust version of LASSO by replacing the squared error loss of LASSO

with Huber loss. We propose to use the Huber loss to extend the group LASSO such that the new method, Huber group LASSO, is more resistant to the large noises and outliers.

To solve the Huber group LASSO, a new algorithm is developed in our previous work [12], which combines the idea of auxiliary function minimization [13] and the block coordinates descent method [14]. The proposed algorithm is efficient and can also be adapted for solving the group LASSO problem without the orthogonality restriction. In this study, we analyze the convergence of our proposed algorithm and show that the sequence the algorithm generated indeed converges to the optimal solution of the Huber group LASSO problem. Instead of picking a specific value for the tuning parameter which corresponds to a determinant network topology as in our previous work [12], in this study, we adapt the "stability selection" [15] strategy to our method to find a network consisting of edges with probabilities or scores. The Huber group LASSO is applied to both simulation data and real experimental data and its performances are compared with those of the group LASSO in terms of areas under the receiver operating characteristic (AUROC) and areas under the precision-recall (AUPR). Results show that the Huber group LASSO outperforms the group LASSO and therefore demonstrate the effectiveness of our proposed method.

Briefly, the remainder of the paper is organized as follows. In Model Section, we introduced the model for the GRN, based on which the network topology is inferred. In Result Section, our proposed method is applied to the both simulation data and real experimental data. The results demonstrate the effectiveness of our method. Then, we conclude this study and point out the future work along this research in Conclusion Section. Details of the method and its theoretical analysis can be found in Method Section.

Model
A model for GRN consisting of p genes is used in this study [16]:

$$\dot{\mathbf{x}} = \mathbf{C}\mathbf{x} + \mathbf{S}\mathbf{r}, \tag{1}$$
$$\mathbf{r} = \mathbf{f}(\mathbf{x}),$$

where $\mathbf{x} = [x_1, \ldots, x_p]^T \in \mathcal{R}^p$ is the vector of mRNA concentrations; $\mathbf{C} = \text{diag}[-c_1, \ldots, -c_p] \in \mathcal{R}^{p \times p}$ is a diagonal matrix with $c_i > 0$ the degradation rate of gene i; the vector $\mathbf{r} = [r_1, \ldots, r_m]^T \in \mathcal{R}^m$ represents the reaction rates, which is a function of mRNA concentrations and $\mathbf{S} \in \mathcal{R}^{p \times m}$ is the stoichiometric matrix of the network. We assume that reaction rate $\mathbf{r}$ is a linear combination of mRNA concentrations,

$$\mathbf{r} = \mathbf{F}\mathbf{x}, \tag{2}$$

where $\mathbf{F} \in \mathcal{R}^{m \times p}$. Then, (1) becomes

$$\dot{\mathbf{x}} = \mathbf{C}\mathbf{x} + \mathbf{M}\mathbf{x}, \qquad (3)$$

where $\mathbf{M} = \mathbf{SF} \in \mathcal{R}^{p \times p}$. The elements of $\mathbf{M} = (m_{ij})_{1 \leq i,\ j \leq p}$ indicate the network topology or regulatory relationships between genes. $m_{ij} \neq 0$ if gene j regulates the expression of gene i. Otherwise, $m_{ij} = 0$, gene j does not regulate gene i.

Since the gene expression levels are sampled at several time points, by using zero order hold discretization method, system (3) is discretized as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} \qquad (4)$$

where $\mathbf{A} = e^{\mathbf{C}\Delta t} + \mathbf{C}^{-1}(e^{\mathbf{C}\Delta t} - \mathbf{I})\mathbf{M}$. Note that $e^{\mathbf{C}\Delta t}$ and $\mathbf{C}^{-1}(e^{\mathbf{C}\Delta t} - \mathbf{I})$ are both diagonal matrices and their diagonal elements are all positive. Thus, the off-diagonal elements of $\mathbf{A} = (a_{ij})_{1 \leq i,j \leq p}$ have the same zero and nonzero pattern as those of $\mathbf{M}$. In this study, we focus on inferring relationships between genes and do not consider self-regulations. As mentioned above, this can be achieved by identifying the nonzero off-diagonal elements in matrix $\mathbf{A}$, which can be interpreted as regulatory strengths. Multiple time-course gene expression datasets for a GRN may be collected under different circumstances. One dataset is assumed to correspond to one inferred GRN topology, and all inferred GRNs should share the same network topology as their corresponding datasets are generated from the same underlying gene network. Our purpose is to reverse engineer the underlying network topology from these multiple datasets. More specifically, suppose we have m time-course gene expression datasets for a gene network: $\tilde{\mathbf{X}}(1), \ldots, \tilde{\mathbf{X}}(m)$, each of which is measured at $n_k + 1$ time points, i.e., $\tilde{\mathbf{X}}(k) \in \mathcal{R}^{p \times (n_k+1)}$. According to the model (4), these datasets should satisfy

$$\mathbf{Y}(k) = \mathbf{A}(k)\mathbf{X}(k) + \mathbf{E}(k), \quad k = 1, \ldots, m, \qquad (5)$$

where $\mathbf{Y}(k) = \left[\tilde{\mathbf{x}}_2(k), \ldots, \tilde{\mathbf{x}}_{n_k+1}(k)\right]$, the last n_k observations; $\mathbf{X}(k) = \left[\tilde{\mathbf{x}}_1(k), \ldots, \tilde{\mathbf{x}}_{n_k}(k)\right]$, the first n_k observations, $\mathbf{A}(k) \in \mathcal{R}^{p \times p}$, the regulatory matrix for the kth dataset and $\mathbf{E}(k)$, the errors or noises. All $\mathbf{A}(k)$'s are required to have the same structure. i.e., zero and nonzero pattern, but do not need to have the same value for every nonzero position because gene network is dynamic and regulatory strength may be different under different circumstances. In this study, we propose to use group LASSO penalty to implement this requirement and to use Huber loss function to take into account the robustness. Details of the proposed method are shown in the Method Section.

Results
To study the effectiveness of the proposed method, the Huber group LASSO is applied to inferring GRNs from both simulation datasets and real experimental datasets

and the results of Huber group LASSO are compared with those from group LASSO in both area under receiver operating characteristic (AUROC) curve and area under the precision and recall (AUPR) curve.
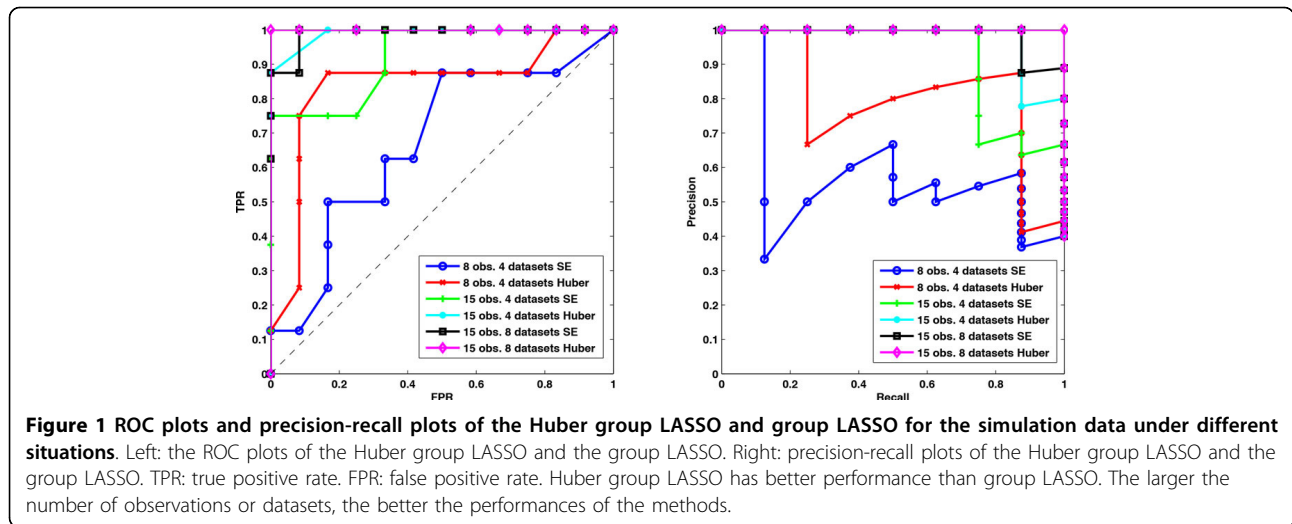
Simulation example
A small-GRN consisting of 5 genes is considered in this example. The corresponding true network topology matrix is

$$\mathbf{A}_0 = \begin{bmatrix} + & - & + & 0 & 0 \\ - & + & 0 & 0 & + \\ 0 & + & + & 0 & 0 \\ + & - & 0 & + & 0 \\ 0 & 0 & 0 & + & + \end{bmatrix},$$

where + and − indicate the existence and regulation types of the edge. We randomly generate m stable regulatory matrices $\mathbf{A}(k)$, $k = 1, \ldots, m$, according to the template $\mathbf{A}_0$, such that $\mathrm{sign}(\mathbf{A}(k)) = \mathrm{sign}(\mathbf{A}_0)$. Then, m simulated time-course gene expression datasets, each with the number of time points, n_k, are generated from (5) with randomly chosen expression values at the first time point. The simulated error follows a mixed Gaussian distribution: with probability of 0.8, it has the distribution $N(0, 1)$ and with probability of 0.2, it has the distribution $N(0, 10^2)$. In this way, the simulated data contain large errors and outliers. To investigate the performances of our methods in different situations, we vary the values of m and n_k and apply the group LASSO and Huber group LASSO respectively to these generated datasets and compare the results from these two methods.

Data are generated under three situations ($m = 8$, $n_k = 15$), ($m = 4$, $n_k = 15$) and ($m = 4$, $n_k = 8$). Using the stability selection procedure that is introduced in the Method Section, network typologies consisting of edges with scores or probabilities are inferred by Huber group LASSO and group LASSO. For the first two situations, we set the number of bootstrap samples as 30 and the moving block length as 10. For the third situation, we set the number of bootstrap samples as 30 and the moving block length as 5. Varying the threshold, the ROC plots and precision-recall plots of each method for different situations are obtained and illustrated in Figure 1. The areas under the ROCs (AUROCs) and precision-recall curves (AUPRs) are calculated and reported in Table 1. From Figure 1 and Table 1 we can see that for each situation, the Huber group LASSO outperforms the group LASSO, i.e. the AUROC and AUPR of Huber group LASSO are larger than those of group LASSO. ROC plots in Figure 1 also show that both methods have better performances than the random guess. For the case of $m = 8$ and $n_k = 15$, the Huber group LASSO even achieves the maximum value

Figure 1 ROC plots and precision-recall plots of the Huber group LASSO and group LASSO for the simulation data under different situations. Left: the ROC plots of the Huber group LASSO and the group LASSO. Right: precision-recall plots of the Huber group LASSO and the group LASSO. TPR: true positive rate. FPR: false positive rate. Huber group LASSO has better performance than group LASSO. The larger the number of observations or datasets, the better the performances of the methods.

of AUROC and AUPR. It can also be seen that for each method, the more the observations or the more the datasets, the larger AUROC and AUPR can be obtained. This is in accord with the intuition because, in this example, more observations or datasets indicate more information as these simulated data are generated under quite similar circumstances. All the simulation results have demonstrated the effectiveness of our proposed method.

In vivo reverse engineering and modeling assessment (IRMA) data

The data used in this example come from the *In vivo* Reverse Engineering and Modeling Assessment (IRMA) experiment [17], where a network composed of five genes (GAL80, GAL4, CBF1, ASH1 and SWI5) was synthesized in yeast *Saccharomyces cerevisiae*, in which genes regulate each other through a variety of regulatory interactions. The network is negligibly affected by endogenous genes and it is responsive to small molecules. Galactose and glucose are respectively used to switch on
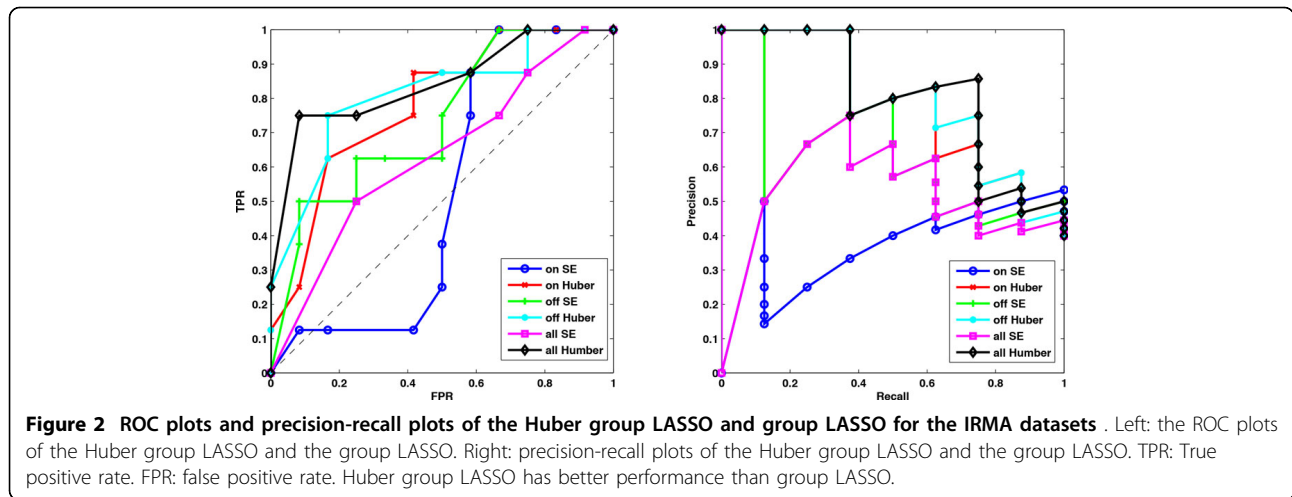
and off the network. In this study, we use the IRMA time-course data consisting of four switch off datasets (with the number of time points varying from 19 to 21) and five switch on datasets (with the number of time points varying from 11 to 16).

The Huber group LASSO and the group LASSO are applied to these data in three cases: (1) switch on datasets, (2) switch off datasets and (3) all datasets, i.e., combing switch on and switch off datasets. In the stability selection procedure, the number of bootstrap samples is 30 for all cases and the moving block length is 14 for the second case and 8 for the other cases. The ROC plots and precision-recall plots for the Huber group LASSO and the group LASSO for each case are illustrated in Figure 2 and the corresponding AUROCs and AUPRs are summarized in Table 2. It can be seen that except the group LASSO for the switch on datasets, the performances of the methods are better than random guesses. The Huber group LASSO outperforms the group LASSO in both AUROCs and AUPRs. All methods for the switch off datasets perform better than for the switch on datasets. The group LASSO for all datasets has better performance than for the switch on datasets but is not as good as for the switch off datasets. The Huber group LASSO for all datasets has the best performance among all cases. This indicates that combing multiple datasets may lead to either the best result or a robust result which is better than the worst case. The network topology with false positive rate (FPR) 0.08 of the Huber group LASSO for all datasets is shown in Figure 3 and the corresponding true positive rate (TPR) is 0.75 with precision 0.86, in which the red edges represent true positives while black edges are false positives. The results show the effectiveness of our method for the IRMA data.

Table 1 The areas under ROC (AUROC) and precision-recall (AUPR) of the Huber group LASSO and the group LASSO for the simulation datasets under different situations.

Situation	Method	AUROC	AUPR
15 observations 8 datasets	SE	0.9896	0.9852
	Huber	1.0000	1.0000
15 observations 4 datasets	SE	0.9219	0.9169
	Huber	0.9896	0.9736
8 observations 4 datasets	SE	0.6719	0.5749
	Huber	0.8385	0.8049

SE: group LASSO. Huber: Huber group LASSO.

Figure 2 ROC plots and precision-recall plots of the Huber group LASSO and group LASSO for the IRMA datasets . Left: the ROC plots of the Huber group LASSO and the group LASSO. Right: precision-recall plots of the Huber group LASSO and the group LASSO. TPR: True positive rate. FPR: false positive rate. Huber group LASSO has better performance than group LASSO.

E. coli SOS network

In this example, we apply the proposed method to identify the real GRN, *E. coli* SOS DNA repair system as shown in Figure 4. This network is responsible for repairing the DNA after some damage happens. LexA acts as the master repressor of many genes in the normal states. When a damage occurs, RecA acts as a sensor and binds to single-stranded DNA to sense the damage and mediates the autocleavage of LexA. The repressions of the SOS genes are halted by the drop in LexA levels. The SOS genes are activated and start to repair the damages. When the repair is done, RecA level drops and stops mediating the autocleavage of LexA. Then, LexA accumulates and represses the SOS genes to make the cell go back to the normal state.

Four time-course gene expression datasets of SOS DNA network are downloaded from the Uri Alon lab (http://www.weizmann.ac.li/mcb/UriAlon.), which are produced from four experiments for various UV light intensities (Experiment 1 and 2: 5 $J\ m^-2$, Experiment 3 and 4: 20 $J\ m^-2$). Each dataset contain 8 genes and their measurements at 50 time points. As other literature did, e.g. [18-21], only 6 genes, i.e., uvrD, lexA, umuD, recA,
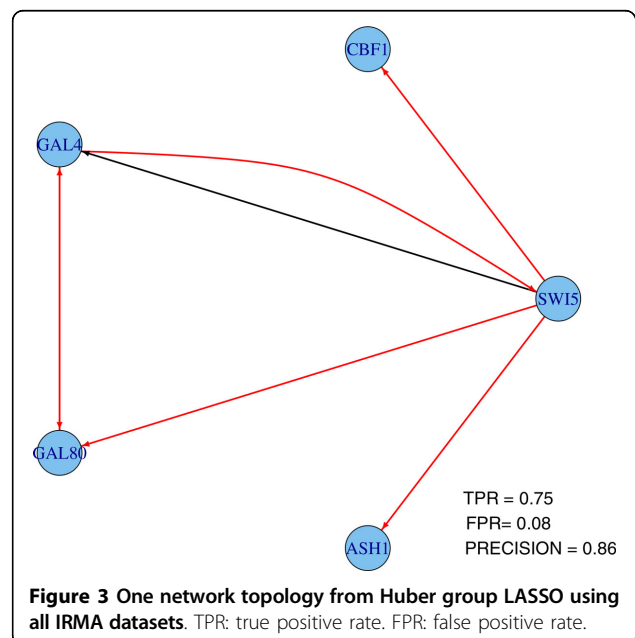
uvrA and polB are considered because they are well studied and the gold standard network of these genes are illustrated in Table 3. Details of the gold standard can be found in [18]. In this study, we do not consider the signs and the self-regulations.
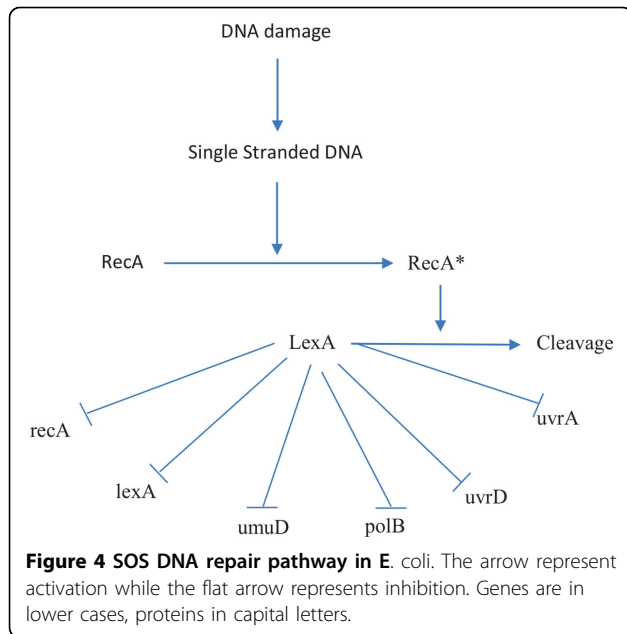
As the conditions for the first two experiments are different for the last two experiments, we consider applying the method to three cases: (1) datasets of experiment 1 and 2, (2) datasets of experiment 3 and 4 and (3) all experiment datasets. In the stability selection procedure, the number of bootstrap samples is 30 and the moving block length is 25 for all cases. The ROC plots and precision-recall plots for the Huber group LASSO and the group LASSO for each case are

Table 2 The areas under ROC (AUROC) and precision-recall (AUPR) of the Huber group LASSO and the group LASSO for the IRMA datasets.

Case	Method	AUROC	AUPR
Switch on datasets	SE	0.5208	0.3711
	Huber	0.7812	0.6971
Switch off datasets	SE	0.7344	0.6341
	Huber	0.8125	0.7928
All datasets	SE	0.6302	0.5122
	Huber	0.8438	0.8049

SE: group LASSO. Huber: Huber group LASSO.

Figure 3 One network topology from Huber group LASSO using all IRMA datasets. TPR: true positive rate. FPR: false positive rate.

TPR = 0.75
FPR= 0.08
PRECISION = 0.86

Figure 4 SOS DNA repair pathway in E. coli. The arrow represent activation while the flat arrow represents inhibition. Genes are in lower cases, proteins in capital letters.

illustrated in Figure 5 and the corresponding AUROCs and AUPRs are illustrated in Table 4. From the ROC plots and AUROCs, it can be seen that the Huber group LASSO performs significantly better than random guess while the group LASSO method is only a little bit better than random guess. Obviously, the Huber group LASSO outperforms the group LASSO both in AUROCs and AUPR for all cases. The Huber group LASSO using experiment 3 and 4 datasets has the best performance. Performance of the Huber group LASSO using all data-sets is between that in the first case and that in the second case. It can be considered as a robust result because of the using of multiple datasets. The network topology with FPR 0 and TPR 0.59 of the Huber group LASSO for all datasets is shown in Figure 6, in which all inferred edges are correct. These results demonstrate the effectiveness of our method for the *E. coli* SOS data.

S. cerevisae cell cycle subnetwork

A cell cycle regulatory subnetwork in S. cerevisae is inferred by the proposed method from 5 experimental microarray datasets. As in [22], the subnetwork consists

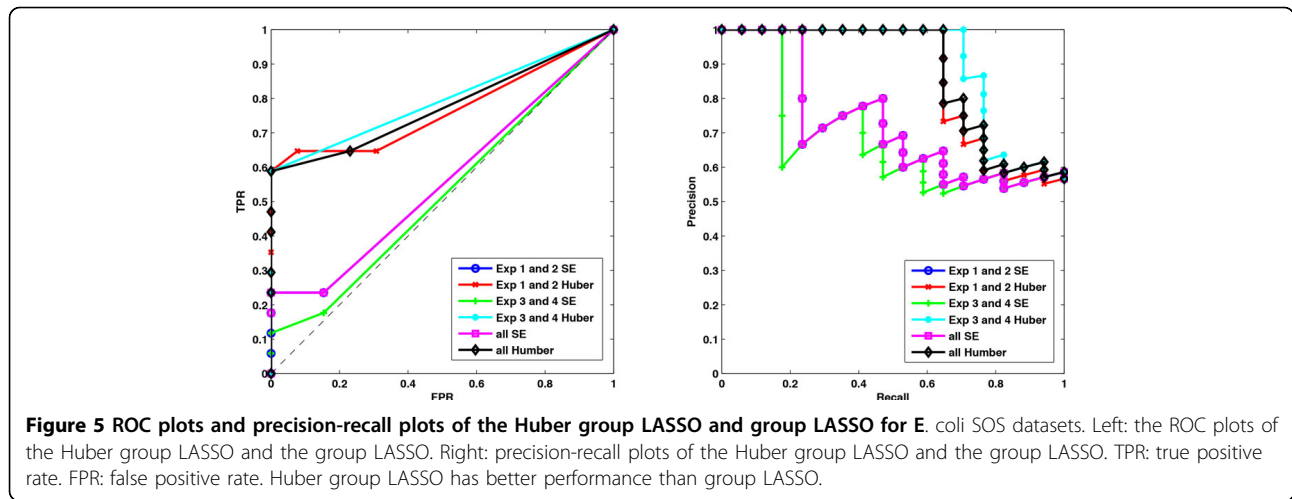**Table 3 Gold standard of SOS network, collected from literature [18].**

	uvrD	lexA	umuD	recA	uvrA	polB
uvrD	0	-1	-1	1	1	0
lexA	0	-1	-1	1	0	0
umuD	0	-1	-1	1	0	0
recA	0	-1	-1	1	0	0
uvrA	1	-1	-1	1	0	0
polB	0	-1	-1	1	0	0

of 27 genes including 10 genes for producing transcription factors (ace2, fkh1, swi4, swi5, mbp1, swi6, mcm1, fkh2, ndd1, yox1) and 17 genes for producing cyclin and cyclin/CDK regulatory proteins (cln1, cln2, cln3, cdc20, clb1, clb2, clb4, clb5, clb6, sic1, far1, spo12, apc1, tem1, gin4, swe1 and whi5). The time-course datasets we use include cell-cycle alpha factor release, cdc15, alpha factor fkh1 fkh2, fkh1,2 alpha factor and Elutriation, which are all downloaded from Stanford Microarray Database (SMD). We apply the Huber group LASSO and the group LASSO respectively to infer the network from the datasets.

In order to demonstrate the effectiveness of the proposed method, the inferred results are compared with the interaction network of the chosen 27 genes, drawn from BioGRID database [23]. The network in the database has 112 interactions, not including the self-regulations, and we take it as the gold standard regulatory network. In the stability selection procedure, the number of bootstrap samples is 30 and the moving block length is 9. The ROC plots and precision-recall plots are illustrated in Figure 7 and the AUROCs and AUPRs are shown in Table 5. We can see that both methods have better performances than random guess and the Huber group LASSO outperforms the group LASSO. One network from Huber group LASSO with FPR 0.43 and TPR 0.59 is shown in Figure 8, in which red edges are those inferred edges having been identified in the database and grey edges might be either false positives or novel discovered regulatory relations. Although the gold standard network extracted from the database may contain false edges or not be complete, this shows the effectiveness of our method to some extent.

Conclusions

A novel method, Huber group LASSO, has been proposed to integrate multiple time-course gene expression datasets to infer the underlying GRN topology. As an extension to the group LASSO, it is robust to large noises and outliers. An efficient algorithm which combines the ideas of auxiliary function minimization and block descent is developed to slove the involved optimization problem. The convergence analysis of the algorithm shows that the sequence generated from the algorithm indeed converges to the optimal solution of the problem. Instead of selecting a specific tuning parameter corresponding to a determinant network topology, an adapted stability selection procedure is used to lead to a network consisting of edges with scores. The applications of the proposed method to the simulation datasets and real experimental datasets show that Huber group LASSO outperforms the group LASSO in both AUROC and AUPR. It also shows that the integration of multiple time-course gene expression datasets by the proposed method lead to reliable inferred network typologies.

Figure 5 ROC plots and precision-recall plots of the Huber group LASSO and group LASSO for E. coli SOS datasets. Left: the ROC plots of the Huber group LASSO and the group LASSO. Right: precision-recall plots of the Huber group LASSO and the group LASSO. TPR: true positive rate. FPR: false positive rate. Huber group LASSO has better performance than group LASSO.

The information in the gene expression data is quite limited. One direction of the future work along this study is to extend the method to be able to integrate other types of data with the gene expression data.

Method

Huber group LASSO

Given m datasets, $\tilde{\mathbf{X}}(1), \ldots, \tilde{\mathbf{X}}(m)$, satisfying model (5), to ensure that all $\mathbf{A}(k)$'s have the same structure, elements of $\mathbf{A}(k)$'s on the same position are grouped together and can be inferred by the group LASSO,

$$\min_{\mathbf{A}(k)} \sum_{i=1}^{p} \sum_{k=1}^{m} w_k \sum_{j=1}^{n_k} (y_{ij}(k) - \mathbf{A}_i(k)^T \mathbf{x_j}(k))^2 + \lambda \sum_{i=1}^{p} \sum_{\ell=1}^{p} \sqrt{a_{i\ell}(1)^2 + \ldots + a_{i\ell}(m)^2},$$

where $\mathbf{A}_i(k)^T$ is the ith row of the matrix $\mathbf{A}(k)$ and $\mathbf{x}_j$ (k) is the jth column of the matrix $\mathbf{X}(k)$. w_k is the weight for the kth dataset, which can be assigned by experience. In this study, we choose $w_k = n_k / \Sigma n_k$ i.e., the more observations the dataset has, the higher weight it is assigned with. The penalty term in (6) takes advantage of the sparse nature of GRNs and has the effect making the grouped parameters to be estimated either all zeros or all non-zeros [10], i.e., $a_{i\ell}(k)$'s, $k = 1, \ldots, m$, become

either all zeros or all non-zeros. Therefore, a consistent network topology can be obtained from the group LASSO method. λ is a tuning parameter which controls the degree of sparseness of the inferred network. The larger the value of λ, the more grouped parameters become zeros.
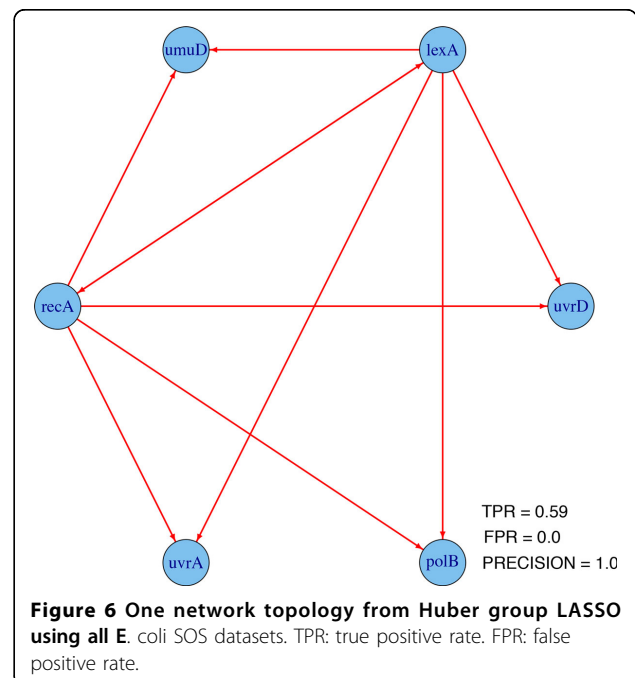
To introduce robustness, we consider using the Huber loss function instead of the squared error loss function and propose the following Huber group LASSO method
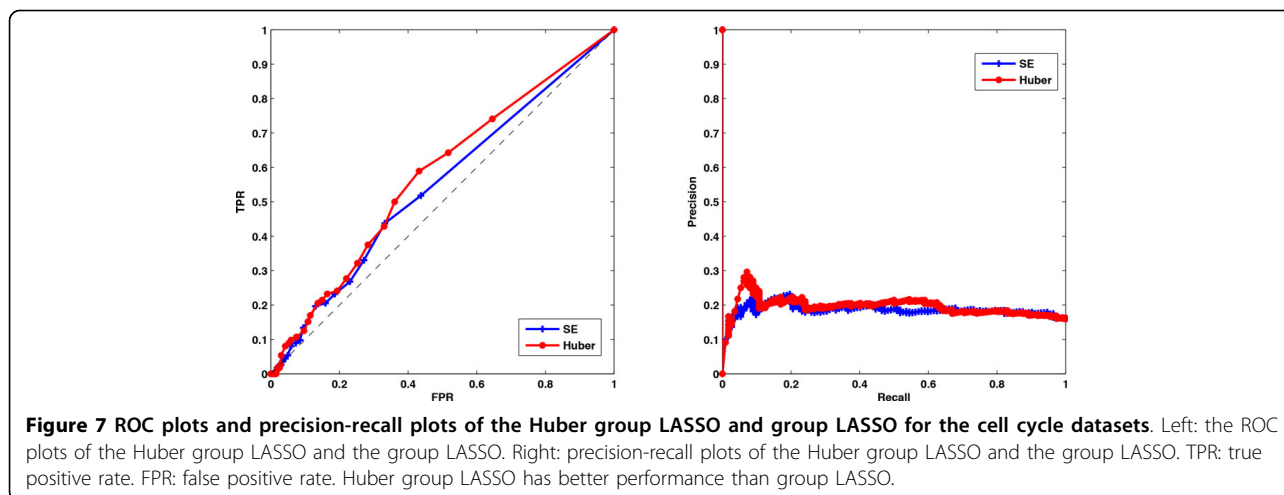
$$\min_{\mathbf{A}(k)} \sum_{i=1}^{p} \sum_{k=1}^{m} w_k \sum_{j=1}^{n_k} H_\delta(y_{ij}(k) - \mathbf{A}_i(k)^T \mathbf{x_j}(k)) + \lambda \sum_{i=1}^{p} \sum_{\ell=1}^{p} \sqrt{a_{i\ell}(1)^2 + \ldots + a_{i\ell}(m)^2}, \quad (7)$$

Table 4 The areas under ROC (AUROC) and precision-recall (AUPR) of the Huber group LASSO and the group LASSO for the E.

Case	Method	AUROC	AUPR
Experiment 1 and 2	SE	0.5588	0.7225
	Huber	0.7670	0.8649
Experiment 3 and 4	SE	0.5204	0.6801
	Huber	0.7941	0.8981
All experiment data	SE	0.5588	0.7225
	Huber	0.7760	0.8756

coli SOS datasets. SE: group LASSO. Huber: (Huber group LASSO.

Figure 6 One network topology from Huber group LASSO using all E. coli SOS datasets. TPR: true positive rate. FPR: false positive rate.

Figure 7 ROC plots and precision-recall plots of the Huber group LASSO and group LASSO for the cell cycle datasets. Left: the ROC plots of the Huber group LASSO and the group LASSO. Right: precision-recall plots of the Huber group LASSO and the group LASSO. TPR: true positive rate. FPR: false positive rate. Huber group LASSO has better performance than group LASSO.

where the Huber loss function is defined as

$$H_\delta(\theta) = \begin{cases} \theta^2 & \text{if } |\theta| \le \delta \\ 2\delta|\theta| - \delta^2 & \text{otherwise.} \end{cases} \quad (8)$$

The squared error and Huber loss function are illustrated in Figure 9. It can be seen that for small errors, these two loss functions are exactly the same while for large errors, Huber loss which increases linearly is less than the squared error loss which increases quadratically. Because Huber loss penalizes much less than the squared error loss for large errors, the Huber group LASSO is more robust than group LASSO when there exists large noise or outliers in the data. It is also known that the Huber loss is nearly as efficient as squared error loss for Gaussian errors [24].

For convenience, we define some notations and rewrite the problems (6) and (7) in more compact forms. Let $\mathbf{Y}_i = [\mathbf{Y}_i(1)^T, \ldots, \mathbf{Y}_i(m)^T]^T$, the vector stacking observations of the ith target gene across all datasets, where $\mathbf{Y}_i(k)^T$ is the ith row of $\mathbf{Y}(k)$. Let $\mathbf{b}_{i\ell} = [a_{i\ell}(1), \ldots, a_{i\ell}(m)]^T$, the vector containing the grouped parameters. Denote by $\mathbf{b}_i = [\mathbf{b}_{i1}^T, \ldots, \mathbf{b}_{ip}^T]^T$ the vector containing all parameters related to the regulation of the ith target gene. According to the orders of the parameters in $\mathbf{b}_i$, re-arrange the rows of $\mathbf{X}(k)$ and piece them together to have $\mathbf{X} = [\mathbf{X}_1^T, \ldots, \mathbf{X}_p^T]^T$ where $\mathbf{X}_i =$

Table 5 The areas under ROC (AUROC) and precision-recall (AUPR) of the Huber group LASSO and the group LASSO for the cell cycle datasets.

Method	AUROC	AUPR
SE	0.5466	0.1844
Huber	0.5753	0.1941

SE: group LASSO. Huber: Huber group LASSO.

$\text{diag}(\mathbf{X}_i(1)^T, \ldots, \mathbf{X}_i(m)^T)$ with $\mathbf{X}_i(k)^T$ being the ith row of $\mathbf{X}(k)$. Then (7) can be rewritten as

$$\sum_{i=1}^p \sum_{j=1}^n \omega_j H_\delta(y_{ij} - \mathbf{x}_j^T \mathbf{b}_i) + \lambda \sum_{i=1}^p \sum_{\ell=1}^n \|\mathbf{b}_{i\ell}\|_2, \quad (9)$$

where $\mathbf{x}_j$ is the jth column of $\mathbf{X}$, y_{ij} is the jth element of $\mathbf{Y}_i$, $n = \sum_{k=1}^m n_k$ and $\omega_i = w_1 I(i \le n_1) + \sum_{k=2}^m w_k I\left(\sum_{l=1}^{k-1} n_l < i \le \sum_{l=1}^k n_l\right)$. (6) can be rewritten similarly.

Optimization algorithm

The minimization of problem (9) is not easy as the penalty term is not differentiable at zero and the Huber loss does not have the second order derivatives at the transition points, $\pm\delta$. Observed that fixing i, the problem (9) can be decomposed into p sub-optimization problems. For each, we get $\mathbf{b}_i$ by minimizing

$$J(\mathbf{b}) = \sum_{j=1}^n \omega_j H_\delta(y_j - \mathbf{x}_j^T \mathbf{b}_i) + \lambda \sum_{\ell=1}^n \|\mathbf{b}_\ell\|_2, \quad (10)$$

where for notational convenience, we omit the subscript i here and $\mathbf{b}_\ell$ is a block of parameters of $\mathbf{b}$, i.e. $\mathbf{b} = [\mathbf{b}_1^T, \ldots, \mathbf{b}_p^T]^T$.

To optimize (10), an iterative method is developed by constructing an auxiliary function, the optimization of which keeps $J(\mathbf{b})$ decreasing. As in [13], given any current estimate $\mathbf{b}^{(k)}$, a function $Q(\mathbf{b}|\mathbf{b}^{(k)})$ is an auxiliary function for $J(\mathbf{b})$ if conditions

$$J(\mathbf{b}^{(k)}) = Q(\mathbf{b}^{(k)}|\mathbf{b}^{(k)}) \ \text{and} \ J(\mathbf{b}) \le Q(\mathbf{b}|\mathbf{b}^{(k)}) \ \text{for all } \mathbf{b}, \quad (11)$$

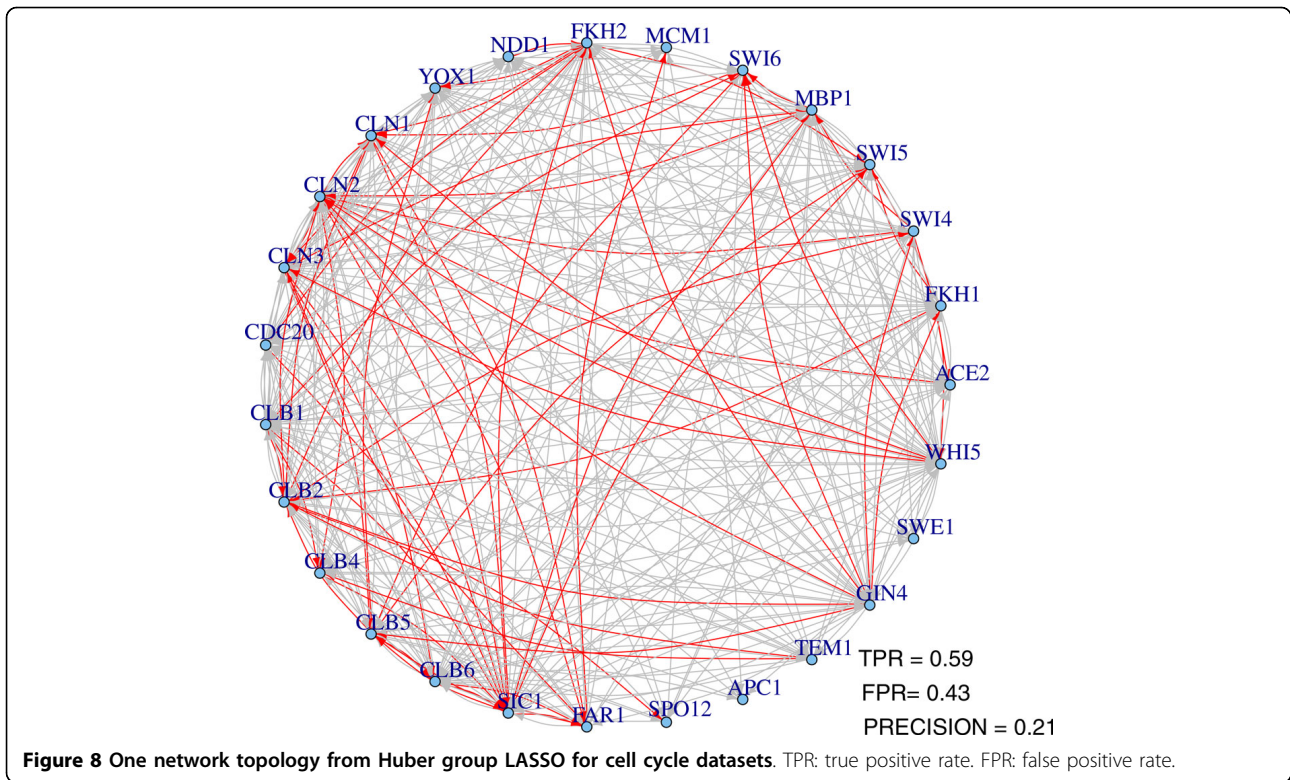are satisfied. In this study, we construct the auxiliary function as

$$Q(\mathbf{b}|\mathbf{b}^{(k)}) = \sum_{j=1}^n \omega_j H_\delta(y_j - \mathbf{x}_j^T \mathbf{b}^{(k)}) - \sum_{j=1}^n \omega_j H'_\delta(y_j - \mathbf{x}_j^T \mathbf{b}^{(k)})\mathbf{x}_j^T(\mathbf{b} - \mathbf{b}^{(k)}) + 2\gamma \left\|\mathbf{b} - \mathbf{b}^{(k)}\right\|_2^2 + \lambda \sum_{\ell=1}^p \|\mathbf{b}_\ell\|_2, \quad (12)$$

Figure 8 One network topology from Huber group LASSO for cell cycle datasets. TPR: true positive rate. FPR: false positive rate.

where γ is the largest eigenvalue of $\sum_{j=1}^{n} \omega_j \mathbf{x}_j \mathbf{x}_j^T$. It can be easily shown that this auxiliary function satisfies (11).

Considering the block structure of $\mathbf{b}$, we apply a block-wise descent strategy [14], i.e., cyclically optimize one block of parameters, $\mathbf{b}_j$, at a time. Denote by
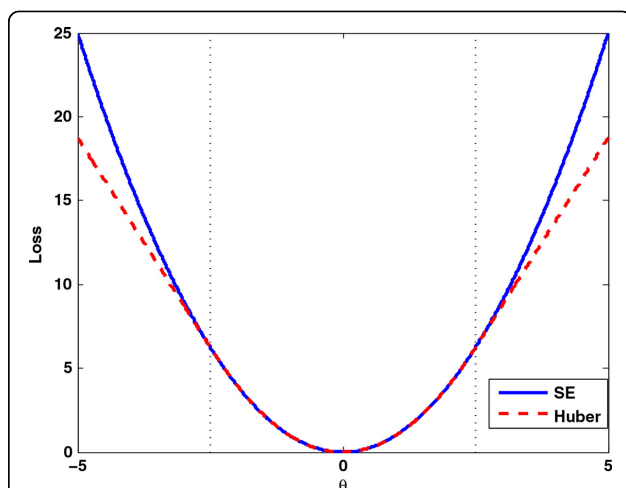

Figure 9 Squared error and Huber loss functions. For small error, θ, squared error loss and Huber loss are the same. For large error, squared error penalizes quadratically while Huber loss penalizes linearly.

$\mathbf{b}^{(k)}(\ell) = [\mathbf{b}_1^{(k+1)^T}, \ldots, \mathbf{b}_\ell^{(k+1)^T}, \mathbf{b}_{\ell+1}^{(k)^T}, \ldots, \mathbf{b}_p^{(k)^T}]^T$ the vector after updating the Rth block. Given $\mathbf{b}^{(k)}(\ell-1)$, update it to $\mathbf{b}^{(k)}(\ell)$ by computing

$$\mathbf{b}_\ell^{(k+1)} = \arg\min_{\mathbf{b}_\ell} Q\left([\mathbf{b}_1^{(k+1)^T}, \ldots, \mathbf{b}_{\ell-1}^{(k+1)^T}, \mathbf{b}_\ell^T, \mathbf{b}_{\ell+1}^{(k)^T}, \ldots, \mathbf{b}_p^{(k)^T}]^T | \mathbf{b}^{(k)}(\ell-1)\right)$$

$$= \left(\frac{1}{4\gamma} - \frac{\lambda}{4\gamma \left\| \sum_{j=1}^{n} \omega_j H'_\delta \left(y_j - \mathbf{x}_j^T \mathbf{b}^{(k)}(\ell-1)\right)\mathbf{x}_{j,\ell} + 4\gamma \mathbf{b}_\ell^{(k)} \right\|_2}\right)$$

$$\times \left(\sum_{j=1}^{n} \omega_j H'_\delta \left(y_j - \mathbf{x}_j^T \mathbf{b}^{(k)}(\ell-1)\right)\mathbf{x}_{j,\ell} + 4\gamma \mathbf{b}_\ell^{(k)}\right). \tag{13}$$

where $\mathbf{x}_{j,\ell}$ is the block of elements in $\mathbf{x}_j$ corresponding to $\mathbf{b}_\ell$ and $(\cdot)_+ = \max(\cdot, 0)$. We repeat to update every block using (13) until it converges. For a specific value of λ, the whole procedure is described as follows:

    1 Initialize $\mathbf{b}(0)$. Set iteration number $k = 0$.
    2 Cycle through (13) one at a time to update the ℓth block, $\ell = 1, \ldots, p$
    3 If $\{\mathbf{b}^{(k)}\}$ converges to $\mathbf{b}^*$, go to the next step. Otherwise, set $k := k + 1$ and go to Step 2.
    4 Return the solution $\mathbf{b}^*$.

Note that the algorithm can be adapted to solve (6) with quite similar derivations. In the following section, we show that the sequence $\{\mathbf{b}^{(k)}\}$ generated from the algorithm

guarantees the objective function $J(\mathbf{b})$ keep decreasing. We also show that the limit point of the sequence generated is indeed the minimum point of $J(\mathbf{b})$.

Convergence analysis

The convergence of the optimization algorithm for the minimization of (10) is analyzed in the way similar to [25]. We first show the descent property of the algorithm.

Lemma 1 *The sequence* $\{\mathbf{b}(k)\}$ *generated from the optimization algorithm keeps the objective function* $J(\mathbf{b})$ *decreasing, i.e.,* $J(\mathbf{b}^{(k)}) \geq J(\mathbf{b}^{(k+1)})$.

Proof By (11) and (13), we have
$$J(\mathbf{b}^{(k)}) = Q(\mathbf{b}^{(k)}|\mathbf{b}^{(k)}) \geq Q(\mathbf{b}^{(k)}(1)|\mathbf{b}^{(k)})$$
$$\geq Q(\mathbf{b}^{(k)}(2)|\mathbf{b}^{(k)}(1)) \geq \cdots \geq Q(\mathbf{b}^{(k)}(p)|\mathbf{b}^{(k)}(p-1)) \geq J(\mathbf{b}^{(k)}(p))$$
$$= J(\mathbf{b}^{(k+1)}).$$

Next, we show that if the generated sequence satisfies some conditions, it converges to the optimal solution.

Lemma 2 *Assume the data* $(\mathbf{y}, \mathbf{X})$ *lies on a compact set and the following conditions are satisfied:*

1 The sequence $\{\mathbf{b}^{(k)}\}$ *is bounded.*
2 For every convergent subsequence $\{\mathbf{b}^{(n_k)}\} \subset \{\mathbf{b}^{(k)}\}$, *the successive differences converge to zeros,* $\mathbf{b}^{(n_k)} - \mathbf{b}^{(n_k-1)} \to \mathbf{0}$.

Then, every limit point $\mathbf{b}^{\infty}$ *of the sequence* $\{\mathbf{b}^{(k)}\}$ *is a minimum for the function* $J(\mathbf{b})$, *i.e., for any* $\delta = (\delta_1^T, \ldots, \delta_p^T)^T \in \mathbb{R}^{mp}$,

Proof For any $\mathbf{b} = (\mathbf{b}_1^T, \ldots, \mathbf{b}_p^T)^T \in \mathbb{R}^{mp}$ and $\delta(j) = (0^T, \ldots, \delta_j^T, \ldots, 0^T)^T \in \mathbb{R}^{mp}$

$$\liminf_{\alpha \downarrow 0+} \left\{ \frac{J(\mathbf{b} + \alpha \delta(j)) - J(\mathbf{b})}{\alpha} \right\} = \nabla_j f(\mathbf{b})^T \delta_j + \liminf_{\alpha \downarrow 0+} \left\{ \frac{\lambda \|\mathbf{b}_j + \alpha \delta_j\|_2 - \|\mathbf{b}_j\|_2}{\alpha} \right\},$$

where $f(\mathbf{b}) = \sum_{i=1}^{n} \omega_i H_\delta(y_i - \mathbf{x}_i^T \mathbf{b})$ and ∇_j represents the partial derivatives with respect to the jth block of parameters. Denote the second term by $\partial P(\mathbf{b}_j; \delta_j)$ and it has

$$\partial P(\mathbf{b}_j; \delta_j) = \begin{cases} \lambda \dfrac{\mathbf{b}_j^T \delta_j}{\|\mathbf{b}_j\|_2} & \text{if } \mathbf{b}_j \neq 0, \\ \lambda \|\delta_j\|_2 & \text{otherwise.} \end{cases} \quad (14)$$

We assume the subsequence $\{\mathbf{b}^{(n_k)}\}$ converges to $\mathbf{b}^{\infty} = (\mathbf{b}_1^{\infty T}, \ldots, \mathbf{b}_p^{\infty T})^T \in \mathbb{R}^{mp}$. From condition 2. and (14), we have

$$\mathbf{b}^{(n_k-1)}(j) = (\mathbf{b}_1^{(n_k)^T}, \ldots, \mathbf{b}_j^{(n_k)^T}, \mathbf{b}_{j+1}^{(n_k-1)^T}, \ldots, \mathbf{b}_p^{(n_k-1)^T})^T \to \mathbf{b}^{\infty}, \quad \text{as } k \to \infty,$$

and

if $\mathbf{b}_j^{\infty} \neq 0$, $\partial P(\mathbf{b}_j^{(n_k)}; \delta_j) \to \partial P(\mathbf{b}_j^{\infty}; \delta_j)$; if $\mathbf{b}_j^{\infty} = 0$, $\partial P(\mathbf{b}_j^{\infty}; \delta_j) \geq \liminf_{k \to \infty} \partial P(\mathbf{b}_j^{(n_k)}; \delta_j)$, $\quad (15)$

since $\mathbf{b}_j^T \delta_j \leq \|\mathbf{b}_j\|_2 \|\delta_j\|_2$.

As $\mathbf{b}_j^{(n_k)}$ minimizes $Q((\mathbf{b}_1^{(n_k)^T}, \ldots, \mathbf{b}_{j-1}^{(n_k)^T}, \mathbf{b}_j^T, \mathbf{b}_{j+1}^{(n_k-1)^T}, \ldots, \mathbf{b}_p^{(n_k-1)^T})^T | \mathbf{b}^{(n_k)}(j-1))$ with respect to the jth block of parameters, using (14), we have

$$\nabla_j q(\mathbf{b}^{(n_k)}(j)|\mathbf{b}^{(n_k)}(j-1))^T \delta_j + \partial P(\mathbf{b}_j^{(n_k)}; \delta_j) \geq 0, \quad \text{for all } k, \quad (16)$$

with

$$q(\mathbf{b}^{(n_k)}(j)|\mathbf{b}^{(n_k)}(j-1)) = \sum_{i=1}^{n} \omega_i H_\delta(y_i - \mathbf{x}_i^T \mathbf{b}^{(n_k)}(j-1))$$
$$- \sum_{i=1}^{n} \omega_i H'_\delta(y_i - \mathbf{x}_i^T \mathbf{b}^{(n_k)}(j-1)) \mathbf{x}_i^T (\mathbf{b}^{(n_k)}(j) - \mathbf{b}^{(n_k)}(j-1))$$
$$+ 2\gamma \left\| \mathbf{b}^{(n_k)}(j) - \mathbf{b}^{(n_k)}(j-1) \right\|_2^2$$

Due to condition 2.,

$$\nabla_j q(\mathbf{b}^{(n_k)}(j)|\mathbf{b}^{(n_k)}(j-1)) \to \nabla_j f(\mathbf{b}^{\infty}) \quad \text{as } k \to \infty. \quad (17)$$

Therefore, (15), (16) and (17) yield

$$\nabla_j f(\mathbf{b}^{\infty})^T \delta_j + \partial P(\mathbf{b}^{\infty}; \delta_j) \geq \liminf_{k \to \infty} \left\{ \nabla_j q(\mathbf{b}^{(n_k)}(j)|(\mathbf{b}^{(n_k)}(j-1))^T \delta_j + \partial P(\mathbf{b}_j^{(n_k)}; \delta_j) \right\} \geq 0, \quad (18)$$

for any $1 \leq j \leq p$.
For $\delta = (\delta_1^T, \ldots, \delta_p^T)^T \in \mathbb{R}^{mp}$, due to the differentiability of $f(\mathbf{b})$,

$$\liminf_{\alpha \downarrow 0+} \left\{ \frac{J(\mathbf{b}^{\infty} + \alpha \delta) - J(\mathbf{b}^{\infty})}{\alpha} \right\} = \sum_{j=1}^{p} \nabla_j f(\mathbf{b}^{\infty})^T \delta_j + \sum_{j=1}^{p} \liminf_{\alpha \downarrow 0+} \left\{ \frac{\lambda \|\mathbf{b}_j^{\infty} + \alpha \delta_j\|_2 - \|\mathbf{b}_j^{\infty}\|_2}{\alpha} \right\}$$
$$= \sum_{j=1}^{p} \{ \nabla_j f(\mathbf{b}^{\infty})^T \delta_j + \partial P(\mathbf{b}^{\infty}; \delta_j) \} \geq 0.$$

Finally, we show that the sequence generated from the proposed algorithm satisfies these two conditions.

Theorem 3 Assuming the data $(\mathbf{y}, \mathbf{X})$ lies on a compact set and no column of $\mathbf{X}$ is identically $\mathbf{0}$, the sequence $\{\mathbf{b}^{(k)}\}$ generated from the algorithm converges to the minimum point of the objective function $J(\mathbf{b})$.

Proof We only need to show that the generated sequence meets the conditions in Lemma 2.

For the sake of notational convenience, for fixed j and $(\mathbf{b}_1^T, \ldots, \mathbf{b}_{j-1}^T, \mathbf{b}_{j+1}^T, \ldots, \mathbf{b}_p^T)^T$ define

$$\chi(\cdot) : \mathbf{u} \mapsto J((\mathbf{b}_1^T, \ldots, \mathbf{b}_{j-1}^T, \mathbf{u}^T, \mathbf{b}_{j+1}^T, \ldots, \mathbf{b}_p^T)^T).$$

Let $\mathbf{b}(\mathbf{u})$ be the vector containing $\mathbf{u}$ as its jth block of parameters with other blocks being the fixed values.

Assume $\mathbf{u} + \delta$ and $\mathbf{u}$ represent the values of the jth block of parameters before and after the block update, respectively. Hence, as defined in (12), $\mathbf{u}$ is obtained by minimizing the following function with respect to the jth block in the algorithm:

$$Q(\mathbf{b}(\mathbf{u})|\mathbf{b}(\mathbf{u} + \delta))$$
$$= f(\mathbf{b}(\mathbf{u} + \delta)) + \nabla_j f(\mathbf{b}(\mathbf{u} + \delta))^T (\mathbf{u} - (\mathbf{u} + \delta))$$
$$+ 2\gamma \|\mathbf{u} - (\mathbf{u} + \delta)\|_2^2 + \lambda \|\mathbf{u}\|_2 + \lambda \sum_{\ell \neq j} \|\mathbf{b}_\ell\|_2. \quad (19)$$

where $f(\mathbf{b}) = \sum_{i=1}^{n} \omega_i H_\delta(y_i - \mathbf{x}_j^T \mathbf{b})$ and $\nabla_j f(\mathbf{b}) = \sum_{i=1}^{n} \omega_i H'_\delta(y_i - \mathbf{x}_j^T \mathbf{b}) \mathbf{x}_{i,j}$. Thus, $\mathbf{u}$ should satisfy

$$\nabla_j f(\mathbf{b}(\mathbf{u} + \delta)) - 4\gamma \delta + \lambda \mathbf{s} = 0, \quad (20)$$

where $\mathbf{s} = \mathbf{u}/||\mathbf{u}||_2$ if $\mathbf{u} \neq 0$; $||\mathbf{s}||_2 \leq 1$ if $\mathbf{u} = 0$. Then, we have

$$
\begin{aligned}
&\chi(\mathbf{u} + \delta) - \chi(\mathbf{u}) \\
&= f(\mathbf{b}(\mathbf{u} + \delta)) - f(\mathbf{b}(\mathbf{u})) + \lambda(||(\mathbf{u} + \delta)||_2 - ||\mathbf{u}||_2) \\
&= \nabla_j f(\mathbf{b}(\mathbf{u} + \delta))^T \delta - \nabla_j f(\mathbf{b}(\mathbf{u} + \delta))^T \delta[\nabla_j f(\mathbf{b}(\mathbf{u} + \delta))^T \delta - 4\gamma \delta^T \delta + \lambda \mathbf{s}^T \delta] \\
&\quad + 4\gamma \delta^T \delta + \lambda(||(\mathbf{u} + \delta)||_2 - ||\mathbf{u}||_2 - s^T \delta) \\
&= -\left[\sum_{i=1}^n \omega_i H'_\delta(\gamma_i - \mathbf{x}_j^T \mathbf{b}(\mathbf{u} + \tau \delta)) \mathbf{x}_{i,j}^T \delta - \sum_{i=1}^n \omega_i H'_\delta(\gamma_i - \mathbf{x}_j^T \mathbf{b}(\mathbf{u} + \delta)) \mathbf{x}_{i,j}^T \delta \right] \\
&\quad + 4\gamma \delta^T \delta + \lambda(||(\mathbf{u} + \delta)||_2 - ||\mathbf{u}||_2 - s^T \delta) \\
&\geq -2\gamma(1 - \tau)||\delta||_2^2 + 4\gamma||\delta||_2^2 \geq 2\gamma||\delta||_2^2.
\end{aligned}
\tag{21}
$$

The second and third equalities are obtained using mean value theorem with $\tau \in (0, 1)$ and (20). For the first inequality, the following property of the Huber loss function and the property of subgradient are used.

$$
(H'_\delta(\theta_1) - H'_\delta(\theta_2))(\theta_1 - \theta_2) \leq 2(\theta_1 - \theta_2)^2.
$$

The result from (21) gives that

$$
J(\mathbf{b}^{(k)}(j-1)) - J(\mathbf{b}^{(k)}(j)) \geq 2\gamma \left\| \mathbf{b}_j^{(k)} - \mathbf{b}_j^{(k+1)} \right\|_2^2 = 2\gamma \left\| \mathbf{b}^{(k)}(j-1) - \mathbf{b}^{(k)}(j) \right\|_2^2,
\tag{22}
$$

where $\mathbf{b}^{(k)}(j) = [\mathbf{b}_1^{(k+1)^T}, \ldots, \mathbf{b}_j^{(k+1)^T}, \mathbf{b}_{j+1}^{(k)^T}, \ldots, \mathbf{b}_p^{(k)^T}]^T$.

Using (22) repeatedly across every block, for any k, we have

$$
J(\mathbf{b}^{(k)}) - J(\mathbf{b}^{(k+1)}) \geq 2\gamma \left\| \mathbf{b}^{(k)} - \mathbf{b}^{(k+1)} \right\|_2^2.
$$

Note that by Lemma 1, $\{J(\mathbf{b}^{(k)})\}$ converges as it keeps decreasing and is bounded from below. The convergence of $\{J(\mathbf{b}^{(k)})\}$ yield the convergence of $\{\mathbf{b}^{(k)}\}$. Hence, conditions of Lemma 2 hold which imply that the limit of $\{\mathbf{b}^{(k)}\}$ is the minimum point of $J(\mathbf{b})$.

Implementation

The tuning parameter λ controls the sparseness of the resulted network. A network solution path can be obtained by computing networks on a grid of λ values from $\lambda_{\max} = \max_{i,\ell} \left\| \sum_{j=1}^n \omega_j H'_\delta(\gamma_{ij}) \mathbf{x}_{j,\ell} \right\|$, which is the smallest value that gives the empty network, to a small value, e.g. $\lambda_{\min} = 0.01\lambda_{\max}$. In our previous work [12], BIC criterion is used to pick a specific λ value which corresponds to a determinant network topology. A method called "stability selection" recently proposed by Meinshausen and Buhlmann [15] finds a network with probabilities for edges. Stability selection performs the network inference method, e.g. group LASSO, many times, resampling the data in each run and computing the frequencies with which each edge is selected across these runs. It has been used with the linear regression method to infer GRNs from steady-state gene expression data in Haury et al. [26] and has shown perspective effectiveness. In this study, we

adapt the stability selection method to finding GRN topology from multiple time-course gene expression datasets. Given a family of m time-course gene expression datasets $\tilde{\mathbf{X}}(k) \in \mathcal{R}^{p \times n_k}$, $k = 1, \ldots, m$, for a specific $\lambda \in \Lambda$, the stability selection procedure is as follows

1 Use moving block bootstrap to draw N bootstrap samples for every dataset to form N bootstrap families of multiple time-course datasets, i.e. $\tilde{\mathbf{X}}(k)^{*(b)}\}_{k=1}^m$, $b = 1, \ldots, N$

2 Use the proposed Huber group LASSO to infer $\{\mathbf{A}(k)_\lambda^{*(b)}\}_{k=1}^m$ from the bth bootstrap family of datasets. Denote by $\mathbf{A}_\lambda^{*(b)}$ the network topology shared by $\{\mathbf{A}(k)_\lambda^{*(b)}\}_{k=1}^m$.

3 Compute the frequencies for each edge (i, j), i.e., from the gene j to gene i, in the network

$$
\prod_\lambda(i, j) = \frac{\#\{b : \mathbf{A}_\lambda^{*(b)}(i, j) \neq 0\}}{N},
\tag{24}
$$

where $\mathbf{A}_\lambda^{*(b)}(i, j)$ is the (i, j)'s entry of $\mathbf{A}_\lambda^{*(b)}$ and $\#\{\cdot\}$ is the number of elements in that set.

For a set of $\lambda \in \Lambda$, the probability of each edge in the inferred network is

$$
\prod(i, j) = \max_{\lambda \in \Lambda} \prod_\lambda(i, j)
\tag{25}
$$

The final network topology can be obtained by setting a threshold, edges with probabilities or scores less than which are considered nonexistent. This study only focus on giving a list of edges with scores. The selection of threshold is not discussed here. The stability selection procedure can also be applied with the group LASSO method (6).

Since the data used are time series data, the moving block bootstrap method is employed in the first step to draw bootstrap samples from each dataset. For a dataset with n observations, in the moving block bootstrap with block length l, the data is split into $n - l + 1$ blocks: block j consists of observations j to $j + l - 1$, $j = 1, \ldots,$ $n - l + 1$. $[n/b]$ blocks are randomly drawn from $n - l + 1$ blocks with replacement and are aligned in the order they are picked to form a bootstrap sample.

Another tuning parameter δ controls the degree of robustness. Generally, it picks $\delta = 1.345\hat{\sigma}$ where $\hat{\sigma}$ is the estimated standard deviation of the error and $\hat{\sigma} = MAD/0.6745$, where MAD is the median absolute deviation of the residuals. In this study, we use the least absolute deviations (LAD) regression to obtain the residuals. To avoid the overfitting of LAD which leads to a very small δ, we choose by $\delta = \max(1.345\hat{\sigma}, 1)$.

Authors' details

[1]Department of Mechanical Engineering, University of Saskatchewan, 57 Campus Drive, S7N 5A9 Saskatoon, Canada. [2]Division of Biomedical Engineering, University of Saskatchewan, 57 Campus Drive, S7N 5A9 Saskatoon, Canada.

Published: 22 October 2014

References

1. Liu LZ, Wu FX, Zhang W: **Inference of Biological S-System Using the Separable Estimation Method and the Genetic Algorithm.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2012, **9(4)**:955-965.
2. Peng J, Wang P, Zhou N, Zhu J: **Partial Correlation Estimation by Joint Sparse Regression Models.** *Journal of the American Statistical Association* 2009, **104(486)**:735-746, [PMID: 19881892].
3. Husmeier D: **Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks.** *Bioinformatics* 2003, **19(17)**:2271-2282.
4. Saito S, Hirokawa T, Horimoto K: **Discovery of Chemical Compound Groups with Common Structures by a Network Analysis Approach (Affinity Prediction Method).** *Journal of Chemical Information and Modeling* 2011, **51**:61-68.
5. Basso K, Margolin A, Stolovitzky G, Klein U, Riccardo D, Califano A: **Reverse engineering of regulatory networks in human B cells.** *Nature genetics* 2005, **37(4)**:382-390.
6. Pinna A, Soranzo N, de la Fuente A: **From knockouts to networks: establishing direct cause-effect relationships through graph analysis.** *PloS one* 2010, **5(10)**.
7. Tenenhaus A, Guillemot V, Gidrol X, Frouin V: **Gene Association Networks from Microarray Data Using a Regularized Estimation of Partial Correlation Based on PLS Regression.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2010, **7(2)**:251-262.
8. Wang Y, Joshi T, Zhang X, Xu D, Chen L: **Inferring gene regulatory networks from multiple microarray datasets.** *Bioinformatics (Oxford, England)* 2006, **22(19)**:2413-2420.
9. Chen BL, Liu LZ, Wu FX: **Inferring gene regulatory networks from multiple time course gene expression datasets.** *Systems Biology (ISB), 2011 IEEE International Conference on* 2011, 12-17.
10. Yuan M, Lin Y: **Model selection and estimation in regression with grouped variables.** *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2006, **68**:49-67.
11. Rosset S, Zhu J: **Piecewise linear regularized solution paths.** *Ann Stat* 2007, **35(3)**:1012-1030.
12. Liu LZ, Wu FX, Zhang WJ: **Robust inference of gene regulatory networks from multiple microarray datasets.** *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on* 2013, 544-547.
13. Seung D, Lee L: **Algorithms for non-negative matrix factorization.** *Advances in neural information processing systems* 2001, **13**:556-562.
14. Tseng P: **Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization.** *Journal of Optimization Theory and Applications* 2001, **109(3)**:475-494.
15. Meinshausen N, Buhlmann P: **Stability selection.** *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2010, **72(4)**:417-473.
16. Wu FX, Liu LZ, Xia ZH: **Identification of gene regulatory networks from time course gene expression data.** *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE* 2010, 795-798.
17. Cantone I, Marucci L, Iorio F, Ricci MA, Belcastro V, Bansal M, Santini S, di Bernardo M, di Bernardo D, Cosma MP: **A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches.** *Cell* 2009, **137**:172-181.
18. Yang X, Dent JE, Nardini C: **An S-System Parameter Estimation Method (SPEM) for Biological Networks.** *Journal of Computational Biology* 2012, **19(2)**:175-187.
19. Kabir M, Noman N, Iba H: **Reverse engineering gene regulatory network from microarray data using linear time-variant model.** *BMC Bioinformatics* 2010, **11(Suppl 1)**:S56.
20. Kimura S, Nakayama S, Hatakeyama M: **Genetic network inference as a series of discrimination tasks.** *Bioinformatics* 2009, **25(7)**:918-925.
21. Hsiao YT, Lee WP: **Inferring robust gene networks from expression data by a sensitivity-based incremental evolution method.** *BMC Bioinformatics* 2012, **13(Suppl 7)**:S8.
22. Montefusco F, Cosentino C, Amato F: **CORE-Net: exploiting prior knowledge and preferential attachment to infer biological interaction networks.** *Systems Biology, IET* 2010, **4(5)**:296-310.
23. Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M: **BioGRID: a general repository for interaction datasets.** *Nucleic Acids Research* 2006, **34(suppl 1)**:D535-D539.
24. Hastie T, Tibshirani R, Friedman J: **The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics).** New York: Springer; 2011.
25. Mazumder R, Friedman JH, Hastie T: **SparseNet: Coordinate Descent With Nonconvex Penalties.** *Journal of the American Statistical Association* 2011, **106(495)**:1125-1138.
26. Haury AC, Mordelet F, Vera-Licona P, Vert JP: **TIGRESS: Trustful Inference of Gene REgulation using Stability Selection.** *BMC Systems Biology* 2012, **6**:145.