

Inferring gene expression networks with hubs using a degree weighted Lasso approach

Nurgazy Sulaimanov^{1,2,*}, Sunil Kumar^{1,2}, Frédéric Burdet³,
Mark Ibberson³, Marco Pagni³, Heinz Koepl^{1,2,†}

¹Department of Electrical Engineering and Information Technology, ²Department of Biology, Technische Universität Darmstadt, Darmstadt, Germany, ³Swiss Institute of Bioinformatics, Lausanne, Switzerland

This article has been submitted to the Bioinformatics.

October 6, 2017

Abstract

Genome-scale gene networks contain regulatory genes called hubs that have many interaction partners. These genes usually play an essential role in gene regulation and cellular processes. Despite recent advancements in high-throughput technology, inferring gene networks with hub genes from high-dimensional data still remains a challenging problem. Novel statistical network inference methods are needed for efficient and accurate reconstruction of hub networks from high-dimensional data.

To address this challenge we propose DW-Lasso, a degree weighted Lasso (least absolute shrinkage and selection operator) method which infers gene networks with hubs efficiently under the low sample size setting. Our network reconstruction approach is formulated as a two stage procedure: first, the degree of networks is estimated iteratively, and second, the gene regulatory network is reconstructed using degree information. A useful property of the proposed method is that it naturally favors the accumulation of neighbors around hub genes and thereby helps in accurate modeling of the high-throughput data under the assumption that the underlying network exhibits hub structure. In a simulation study, we demonstrate good predictive

performance of the proposed method in comparison to traditional Lasso type methods in inferring hub and scale-free graphs. We show the effectiveness of our method in an application to microarray data of *E.coli* and RNA sequencing data of Kidney Clear Cell Carcinoma from The Cancer Genome Atlas datasets.

1 Introduction

With the advent of high-throughput technologies such as microarrays and RNA sequencing, inference of gene regulatory networks has attracted much scientific interest over the last decade. The technologies enable simultaneous measurement of large numbers of genes. This leads to the challenge of inferring large-scale gene regulatory networks from high-dimensional data (Marbach et al. (2012); Hill et al. (2016)). On the other hand, high-throughput experiments still remain costly and therefore experiments are usually carried out for a setting with far less samples than genes. Shrinkage methods from high-dimensional statistics such as the graphical Lasso (Glasso) (Friedman et al. (2008)) and the nodewise regression (Meinshausen and Bühlmann (2006)) are a good choice for the graph reconstruction in this scenario. Gene regulatory networks contain regulatory genes with many interaction partners called *hubs* that are essential for

*nurgazy.sulaimanov@bcs.tu-darmstadt.de

†heinz.koepl@bcs.tu-darmstadt.de

the viability of the organism because they are a central part of the interaction network (Blais and Dynlacht (2005)). This property also makes the hub genes potential drug targets. Hub genes have been shown to be conserved in several organisms and it was suggested that their normal function is to act as genetic buffers, minimizing the effects of mutations in other genes (Lehner et al. (2006)). For example, the tumor suppressor Tp53 is a central hub in a molecular network that controls cell proliferation and death in response to oncogenic conditions (Collavin et al. (2010)). The protein encoded by Tp53 is a powerful tumor suppressor, as proven by a studies of in vivo models and confirmed by frequent mutation in human cancers (Donehower and Lozano (2009)). It is very challenging to reconstruct such networks with hubs and the traditional l_1 -based methods such as Glasso and node-wise regression perform poorly for these type of networks under the setting of less samples than genes. The edges of networks estimated from the standard l_1 -norm based methods correspond to a maximum posterior mode where independent, two sided exponential prior distribution are placed on the edge coefficients (Hans (2009)). Such approaches indirectly assume that each edge is treated equally which in turn corresponds to an Erdős-Rényi network in which most of the nodes have similar degrees (Tan et al. (2014), Erdős and Rényi (1959)).

A number of methods have been developed by different authors to tackle the problem of hub network inference from high-dimensional data. For example, Peng et al. (2009) proposed a joint sparse regression model called SPACE (sparse partial correlation estimation) that allows to incorporate estimated degree information as a prior. However, their method does not perform well when the graph contains a few hubs that are highly connected to other nodes. Liu and Ihler (2011) proposed a reweighted method to infer scale-free and hub type networks that performs better than graphical Lasso and nodewise regression. However, the networks considered in their study contain hub nodes that are far less connected than the hubs which we consider in our study. Tan et al. (2014) proposed l_1 -based method involving three penalty parameters that control hub sparsity, a selection of hub nodes and overall sparsity. The method performs well in the presence of highly connected nodes in the graph, however, it includes three penalty pa-

rameters and final estimates are highly dependent on how well these penalty parameters are chosen. In the context of Ising graphical models, Tandon and Ravikumar (2014) provided theoretical guarantees for l_1 -based logistic regression to recover the networks with a few hubs that have large degrees. Moreover, the authors provided a quantitative criterion to detect hub and non-hub nodes in the network. Other authors proposed methods to screen the hubs in the network in the context of graphical models (Firouzi and Hero (2013); Hero and Rajaratnam (2012)). However, these methods do not aim at estimating the hub network.

In this manuscript, we address the problem of estimating hub graphs from data in small n large p scenarios and propose a method (DW-Lasso) that consists of an iterative degree estimation step followed by a graph reconstruction step (Figure 1). Our method in spirit is close to the method by Tandon and Ravikumar (2014), however, our method is designed for Gaussian graphical models and treats the genes as continuous variables. We demonstrate the increased performance of our approach in comparison to traditional l_1 -based methods under high-dimensional settings both on simulated and experimental data.

2 Methods

2.1 Notation

Consider a p -dimensional random vector $X = (X_1, \dots, X_p)^T$, which represents the expression levels of p genes. We assume that X follows the multivariate normal distribution with zero mean and covariance matrix Σ . We have a design matrix $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_p)$ of size $n \times p$, where n is the number of observations that are independently and identically distributed.

We consider a undirected graph $G = (V, E)$, with a set of nodes, $V = \{1, \dots, p\}$ and a set of edges, $E \subseteq V \times V$. In our case, V represents a set of genes and E represents a set of conditional dependencies between the genes.

Under the multivariate normal distribution assumption, the concentration matrix $\Theta \equiv \Sigma^{-1}$ encodes the conditional independence relationships between genes through the graph G . In particular, the genes i and j are conditionally indepen-

dent given the other genes, if (i, j) -th entry of the concentration matrix is zero.

Let ρ_{ij} denote a partial correlation between any two genes X_i and X_j . It is related to the elements of the concentration matrix through the relation

$$\rho_{ij} = -\frac{\Theta_{ij}}{\sqrt{\Theta_{ii}\Theta_{jj}}}, \quad (1)$$

for $i \neq j$ and $\rho_{ij} = 1$ for $i = j$. There is an edge between genes i and j , if $\rho_{ij} \neq 0$ and no edge if $\rho_{ij} = 0$ for $i \neq j$. Furthermore, we define the degree of gene i by d_i which represents the number of genes directly connected to gene i . It is computed using the following expression

$$d_i = \sum_{j=1, j \neq i}^p \mathbb{1}\{\Theta_{ij} \neq 0\}, \forall i \in V. \quad (2)$$

2.2 Degree estimation using a node-wise regression method

Tandon and Ravikumar (2014) proposed a theoretical criterion that allows to differentiate between hubs and non-hubs. This makes possible to infer the network using the non-hubs estimates by ignoring the estimates from hub nodes under the assumption that the underlying network contains a few hubs with large degrees. Motivating by this fact, we attempt to distinguish the hub genes based on their degrees using the nodewise regression approach (Meinshausen and Bühlmann (2006)). For example, simulation results on scale-free networks indicate that nodewise regression is able to differentiate between hub and non-hubs nodes for some values of penalty parameter (Figure 4A and for more information see Section 2.4).

For nodewise regression, we assume \mathbf{X}_i is a response vector corresponding to gene i and $\mathbf{X}^{\setminus i}$ is the submatrix after excluding the i -th column of the design matrix \mathbf{X} . In order to reconstruct the graph, we assume a linear relationship between the genes. Therefore, we linearly regress gene i given the remaining genes, using the following model

$$\mathbf{X}_i = \mathbf{X}^{\setminus i} \boldsymbol{\beta}_i + \epsilon_i, \quad (3)$$

where $\boldsymbol{\beta}_i$ is a coefficient vector of length $p - 1$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. Each element of the vector $\boldsymbol{\beta}_i$ can

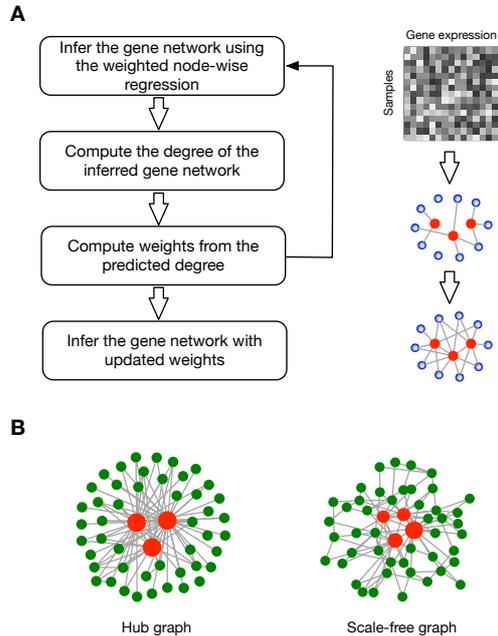


Figure 1: (A) Workflow of the network inference with the degree weighted Lasso method (DW-Lasso). Initially, the network is reconstructed using nodewise regression (MB-Lasso) which corresponds to the weighted nodewise-regression with the same weights. In the second step, the degree of the resulting network is computed. In the third step, the weights are computed from the estimated degrees. These weights are incorporated into the weighted nodewise regression and updated until the algorithm attains a steady state. The weights at the steady state are then used to reconstruct the final network. Red nodes represent the hub nodes. (B) Illustration of hub and scale-free networks considered in the study.

be expressed in terms of partial correlations by

$$\beta_{ij} = -\rho_{ij} \sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}}, \quad (4)$$

where ρ_{ij} is defined in (1). Under the high-dimensional setting ($p > n$), the coefficient vector $\boldsymbol{\beta}_i$ cannot be uniquely estimated, because $(\mathbf{X}^{\setminus i})^T \mathbf{X}^{\setminus i}$ is not invertible. Therefore, we adopt a Lasso approach (Tibshirani (1996)) to estimate the coefficient vector for gene i using the following

objective function

$$\hat{\beta}_i = \arg \min_{\beta_i} \left(\frac{1}{n} \|\mathbf{X}_i - \mathbf{X} \setminus^i \beta_i\|_2^2 + \lambda_1 \|\beta_i\|_1 \right), \quad (5)$$

where $\lambda_1 > 0$ is the penalty parameter that enforces sparsity in the estimates. By performing a nodewise regression for all genes (Meinshausen and Bühlmann (2006)), we estimate the off-diagonal elements of the concentration matrix $\hat{\Theta}_{ij} = \hat{\beta}_{ij}$ for all $i \neq j$. Since the estimates of (5) are not symmetric, we apply the following transformation to obtain the symmetric matrix

$$\tilde{\Theta} = (\hat{\Theta} + \hat{\Theta}^T)/2, \quad (6)$$

Furthermore, we compute the degree vector $\hat{\mathbf{d}} = (\hat{d}_1, \dots, \hat{d}_p)$ from the estimated concentration matrix $\tilde{\Theta}$, where each element of the vector $\hat{\mathbf{d}}$ is defined as

$$\hat{d}_i = \sum_{j=1, j \neq i}^p \mathbb{1}\{\tilde{\Theta}_{ij} \neq 0\}, \forall i \in V \quad (7)$$

In the next step, we compute a weight for gene i by

$$w_i = (1 + \hat{d}_i)^{-1}. \quad (8)$$

When the estimated degree of a gene i is zero, then the corresponding weight is equal to one. A similar formulation has been applied for the adaptive Lasso using the weight as a function of estimated coefficients (Zou (2006)). Furthermore, we normalize the weights to represent them as probabilities

$$\bar{w}_i = \frac{w_i}{\sum_j w_j}, \forall i \in V \quad (9)$$

Finally, we exploit the weight vector iteratively $\bar{\mathbf{w}} = (\bar{w}_1, \dots, \bar{w}_p)$ as prior information to reconstruct the gene-gene interaction graph using the weighted Lasso.

2.3 Iterative degree estimation and network reconstruction using a weighted nodewise regression method

We propose an iterative method to update the degree vector using the current solution. A similar iterative approach was proposed earlier by Candès et al. (2008), where the weight vector is computed

from the current value of regression coefficients. By employing a weighted Lasso setup, we formulate the iterative approach as follows:

$$\hat{\beta}_i^{(k+1)} = \arg \min_{\beta_i^{(k+1)}} \frac{1}{n} \|\mathbf{X}_i - \mathbf{X} \setminus^i \beta_i^{(k+1)}\|_2^2 + P_k(\beta_i^{(k+1)}) \quad (10)$$

where

$$P_k(\beta_i^{(k+1)}) = \lambda_1 \sum_{j \neq i} \bar{w}_j^{(k)} |\beta_{ij}^{(k+1)}| \quad (11)$$

where, $\bar{w}_j^{(k)}$ is the normalized weight for gene j at k -th iteration. For $k = 1$, we initialize the weight vector as a normalized unit vector and the problem (10) reduces to the standard nodewise regression method defined in (5). For any gene j , if the weight $\bar{w}_j^{(k)}$ is small, the coefficient $\beta_{ij}^{(k+1)}$ is less penalized and this leads to a recovery of the edge between genes i and j . Suppose, in the true graph, there is an edge between a high degree gene i and a low degree gene j . The coefficient $\beta_{ij}^{(k+1)}$ shrinks to zero with high probability, due to high weight $\bar{w}_j^{(k)}$. In contrast, the coefficient $\beta_{ji}^{(k+1)}$ is recovered with high probability due to low weight $\bar{w}_i^{(k)}$. This introduces a further asymmetry in the graph reconstruction. We obtain symmetric estimates using the transformation (6). The illustration of how each gene is weighted by the degree of other genes is depicted in a sample graph with four nodes in Figure 2. We update the normalized weight vector according to the following recursion

$$\bar{\mathbf{w}}^{(k)} = \alpha \bar{\mathbf{w}}^{(k-1)} + (1 - \alpha) \Psi(\bar{\mathbf{w}}^{(k-1)}), \quad k \geq 1, \quad (12)$$

where the i -th component of vector Ψ reads

$$\Psi_i(\bar{\mathbf{w}}^{(k-1)}) = \frac{(1 + \hat{d}_i^{(k-1)})^{-1}}{\sum_j 1/(1 + \hat{d}_j^{(k-1)})}, \quad (13)$$

with α a constant such that $0 < \alpha < 1$. We compute the degree vector $\hat{\mathbf{d}}^*$ via (7) at the $(k+1)$ -th iteration using the estimated coefficients from (10) and using the transformation (6). Using (8) and (9), we then obtain the normalized weight vector $\bar{\mathbf{w}}^* = (\bar{w}_1^*, \dots, \bar{w}_p^*)$, which is used to reconstruct the final graph employing a weighted Lasso regression

$$\hat{\beta}_i = \arg \min_{\beta_i} \left(\frac{1}{n} \|\mathbf{X}_i - \mathbf{X} \setminus^i \beta_i\|_2^2 + P(\beta_i) \right), \quad (14)$$

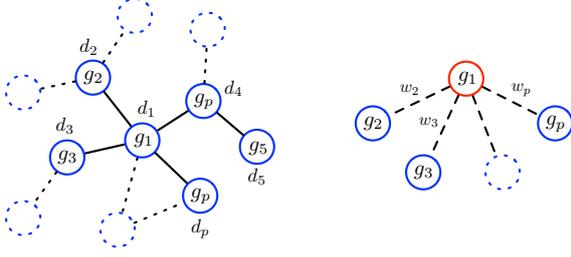


Figure 2: Integrating degree information of the graph into a Lasso setting. Illustration of the network with degree information d_i , $i = 1, \dots, p$ (left). We show how the degree information is integrated into a weighted Lasso setting for gene 1 (right). The corresponding weighted Lasso for gene 1 is then defined as $\hat{\beta}_1 = \arg \min_{\beta_1} \left(n^{-1} \|\mathbf{X}_1 - \mathbf{X} \setminus 1 \beta_1\|_2^2 + P(\beta_1) \right)$ with $P(\beta_1) = |w_2 \beta_{12}| + |w_3 \beta_{13}| + \dots + |w_p \beta_{1p}|$.

with

$$P(\beta_i) = \lambda_2 \sum_{j \neq i} \bar{w}_j^* |\beta_{ij}|, \quad (15)$$

where $\lambda_2 > 0$ is the penalty parameter that controls the sparsity in the estimates. In Section 3.3, for a carefully chosen λ_1 , we demonstrate that weight estimates converge close to weights defined by the true network. Additionally, we show that the performance of DW-Lasso increases with the number of iterations and achieves best results in terms of AUROC at the steady state (Figure 4C).

2.4 Justification of degree separation for nodewise regression

Lets assume that there are r hub and $p - r$ non-hub genes in the network. Lets denote the degree of hub and non-hub genes by \bar{d}_i , $i = 1, \dots, r$ and \tilde{d}_j , $j = r + 1, \dots, p$, respectively. Assume that $\bar{d}_i > n$, $i = 1, \dots, r$ and $\tilde{d}_j < n$, $j = r + 1, \dots, p$ such that for hub genes the Lasso estimation involves more coefficients than samples to estimate, while for non-hub genes it involves less coefficients than the samples. In case of hub genes, it leads to an underestimation problem, meaning that some coefficients cannot be uniquely estimated. In contrast, for non-hubs, the coefficients can be uniquely estimated using large penalty parameters. If we assume that each hub node is connected to most non-hub nodes, some edges connected to hubs can

be recovered by the edges estimated from non-hub nodes. Therefore, the nodewise regression is able to partially infer the edges connected to hub nodes where the number of false positive edges is controlled for large penalty parameters. Simulations with nodewise regression demonstrate that the degree of hub and non-hubs can be separated for the intermediate values of the penalty parameter (Figure 4A).

2.5 Extending the methodology to transcription factor-gene interaction networks

The proposed methodology can be extended to reconstruct transcription factor-gene interaction networks. In this case, we are only interested in the inference of edges between transcription factors and genes. We consider a set of transcription factors $Q = \{1, \dots, m\}$, $Q \subset V$ and a set of genes $G = \{m + 1, \dots, p\}$, $G \subset V$. Let \mathbf{X}^Q be a $n \times m$ -matrix that represents the expression level of m transcription factors. Similarly, let \mathbf{X}^G be a $n \times (p - m)$ -matrix that represents the expression level of $p - m$ genes. To reconstruct the network of transcription factors and genes, we linearly regress gene i against all transcription factors

$$\mathbf{X}_i^G = \mathbf{X}^Q \beta_i^Q + \epsilon_i^G \quad (16)$$

where β_i^Q is a coefficient vector of length m and $\epsilon_i^G \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. We estimate β_i^Q using the Lasso

$$\hat{\beta}_i^Q = \arg \min_{\beta_i^Q} \frac{1}{n} \|\mathbf{X}_i^G - \mathbf{X}^Q \beta_i^Q\|_2^2 + \|\beta_i^Q\|_1 \quad (17)$$

By regressing all genes against transcription factors, we estimate the $\hat{\Theta}^{QG}$ - $(p - m) \times m$ matrix that includes all pairwise estimates between genes and transcription factors. We can further compute the degree of transcription factor j using (7)

$$\hat{d}_j^Q = \sum_{i=1}^{p-m} \mathbb{1}\{\hat{\Theta}_{ij}^{QG} \neq 0\}, \forall j \quad (18)$$

By using (8), the degree \hat{d}_j^Q can be integrated into a weighted Lasso setting

$$\hat{\beta}_i^Q = \arg \min_{\beta_i^Q} \frac{1}{n} \|\mathbf{X}_i^G - \mathbf{X}^Q \beta_i^Q\|_2^2 + P(\beta_i^Q) \quad (19)$$

with

$$P(\beta_i^Q) = \lambda_1 \sum_{j=1}^m \bar{w}_j^Q |\beta_{ij}^Q| \quad (20)$$

where, \bar{w}_j^Q is the normalized weight for transcription factor j . These steps can be solved iteratively as described in Section 2.3.

2.6 Choosing penalty parameters

In this section, we discuss how to select the penalty parameter λ_1 for the degree estimation step. Therefore, we propose to employ a stability selection (Meinshausen and Bühlmann (2010)) to choose the penalty parameters from the data. According to Meinshausen and Bühlmann (2010), we denote a set of estimated graphs as $\hat{G}^\lambda = \{(i, j) \in E; \hat{\Theta}_{ij} \neq 0\}$ obtained for every value $\lambda \in \mathbb{R}_+$. Denote a set of samples as $I_n = \{1, \dots, n\}$. Let I be a random subsample of I_n of size $n/2$ drawn without replacement. If a random subsample I is drawn J times, for every set $D \subseteq E$, one can define the probability of being in the selected set \hat{G}^λ

$$\hat{\Pi}_D^\lambda = J^{-1} \sum_{j=1}^J \mathbb{1}\{D \subseteq \hat{G}^\lambda(I^j)\} \quad (21)$$

For a set of regularization parameters Λ and a cutoff π_{thr} , the set of stable edges is defined as

$$\hat{G}^{stable} = \{D : \max_{\lambda \in \Lambda} \hat{\Pi}_D^\lambda \geq \pi_{thr}\}, \quad \pi_{thr} \in (0, 1) \quad (22)$$

where the edges with high selection probabilities are selected, while the edges with low selection probabilities are excluded. As justified in Meinshausen and Bühlmann (2010), for $\pi_{thr} \in (0.5, 1]$, it is possible to control the number of falsely selected edges W given the expected number of selected edges q_Λ , the total number of edges \bar{q} and the threshold parameter π_{thr}

$$\mathbb{E}(W) \leq \frac{q_\Lambda^2}{\bar{q}(2\pi_{thr} - 1)} \quad (23)$$

under the assumption that the DW-Lasso performs better than the random guess (exchangeability condition, for more information see Meinshausen and Bühlmann (2010)). For a user defined π_{thr} and given number of maximum false positives $\mathbb{E}(W)$,

we compute the expected number of edges q_Λ in the graph. Using the computed value of q_Λ , we select the penalty parameter λ_1 .

2.7 Graph generation procedure

To generate a hub graph, we generate a sparse symmetric adjacency matrix $\mathbf{A} \in \mathbb{R}^{p \times p}$ with the edge probability p_1 . We next randomly select a set of hub nodes of size h which is a predefined parameter. Finally, we generate a set of neighboring nodes around hubs with probability p_2 . The parameters p_1 and p_2 allow us to generate various hub graphs with different sparsity levels. In our case, we set $p_1 = 0.01$ and $p_2 = 0.95$.

We also evaluate the performance of our method on scale-free graphs (Barabasi and Albert (1999), Durrett (2007)). We use a linear preferential attachment approach to generate an adjacency matrix $\mathbf{A} \in \mathbb{R}^{p \times p}$ for a scale-free graph with degree exponent γ . The approach starts with a connected graph that contains m_0 nodes. The new nodes with $m \leq m_0$ edges are added to m_0 existing nodes in the graph. A new node is added to the existing node i depending on the degree k_i with the probability $P(k_i) = k_i / \sum_j k_j$. In our case, we set $m = 3$.

Given the adjacency matrix $\mathbf{A} \in \mathbb{R}^{p \times p}$, we uniformly generate a new weighted adjacency matrix

$$B_{ij} \stackrel{\text{i.i.d.}}{\sim} \begin{cases} \text{Unif}([-0.75, -0.25] \cup [0.25, 0.75]), & \text{if } A_{ij} \neq 0 \\ 0, & \text{if } A_{ij} = 0 \end{cases}$$

Finally, we convert \mathbf{B} into a symmetric matrix $\bar{\mathbf{B}} = (\mathbf{B} + \mathbf{B}^T)/2$. We transform the matrix $\bar{\mathbf{B}}$ to the concentration matrix using the expression

$$\Theta = \bar{\mathbf{B}} + (0.1 + |\lambda_{\min}(\bar{\mathbf{B}})|)\mathbf{I}, \quad (24)$$

where $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of the matrix.

Given the concentration matrix Θ , we generate samples that follow a multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\Sigma = \Theta^{-1}$. In the final step, the data is standardized to mean 0 and standard deviation $\sigma = 1$.

2.8 Quantifying the functional content of a graph

To quantify the association between publically available gene sets and the inferred networks, we

compute two metrics: Knet score implemented in SANTA package (Cornish and Markowitz (2014)) that takes into account the global topology of the network, and compactness score that quantifies the average distance between the genes in the network (Glaab et al. (2010)). Knet function is a modified form of Ripley’s K-function (Gaetan et al. (2010)) and is defined as

$$K^{net}(s) = \frac{2}{(\bar{k}n)^2} \sum_i k_i \sum_j (k_j - \bar{k}) I(d(i, j) \leq s) \quad (25)$$

where, k_i is the phenotype observed at gene i , $\bar{k} = \frac{1}{n} \sum_{i=1}^n k_i$, $d(i, j)$ is the shortest distance between two hits in the network, and $I(d(i, j) \leq s)$ is the indicator function which equals 1, if the distance $d(i, j)$ between hits i and j is smaller or equal to s , and 0 otherwise. For a given gene set M , the compactness score is defined as

$$CS(M) = \frac{2 \sum_{i, j \in M; i < j} d(M_i, M_j)}{|M|(|M| - 1)} \quad (26)$$

where $d(M_i, M_j)$ is the shortest distance between any two pairs of genes in gene set M , and $|M|$ is the cardinality of gene set M .

3 Results on synthetic data

3.1 Performance assessment

We consider the following metrics to evaluate the performance of our method in comparison to state-of-the-art methods.

- Receiver operation characteristics (ROC):

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}$$
- Precision and Recall (PR):

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}$$
- Area under the ROC curve (AUROC)
- Area under the Precision and Recall curve (AUPR)

ROC and PR curves are computed for a fixed value of λ_1 which has been selected by stability selection (Meinshausen and Bühlmann (2010)) (Section 2.6)

and by varying the penalty parameter λ_2 and AUROC and AUPR have been computed from corresponding ROC and AUPR curves.

3.2 Performance on synthetic data

We first evaluate the performance of our method in comparison to state-of-the-art methods such as nodewise regression (MB-Lasso) (Meinshausen and Bühlmann (2006)), graphical Lasso (Glasso); (Friedman et al. (2008)) and sparse partial correlation estimation (SPACE) (Peng et al. (2009)), based on *in silico* study. The performance of selected methods is assessed on hub and scale-free graphs (Figure 1B). We generate a set of hub networks using the procedure described by Tan et al. (2014). We generate scale-free networks using the procedure given in (Barabasi and Albert (1999); Durrett (2007)). Detailed information can be found in Section 2.7.

We randomly create hub networks with node sizes $p = 100$ and 500 that contain three and ten hub nodes, respectively. Here, we consider an extreme setting where the hub nodes are connected to 95 % of non-hub nodes. Moreover, scale-free networks are also generated with node sizes of $p = 100$ and 500 . We generate scale-free networks with degree exponent $\gamma = 2.5$, which also contain hub nodes. Furthermore, we generate 20 different datasets of sample sizes $n = 50$ for $p = 100$ and $n = 100$ for $p = 500$ both for hub and scale-free networks. This allows us to evaluate the performance of the methods in presence of sample noise. The DW-Lasso achieves a good performance in the large p and small n setting (Figure 3), when the true degree information is supplied as a weight vector, which highlights the importance of degree information on the predictive power.

Therefore, the predictions using the true degree information give the upper bound for the prediction accuracy of the proposed method. Simulation results demonstrate that our DW-Lasso performs better than the state-of-the-art methods. In the case of hub networks, all three methods perform poorly, whereas DW-Lasso detects most of the true positives successfully, even when the number of samples is less than the number of genes (Figure 3A and Figure 3B). In the setting $p = 500$ and $n = 100$, the performances of the state-of-the-art methods are almost comparable to a random

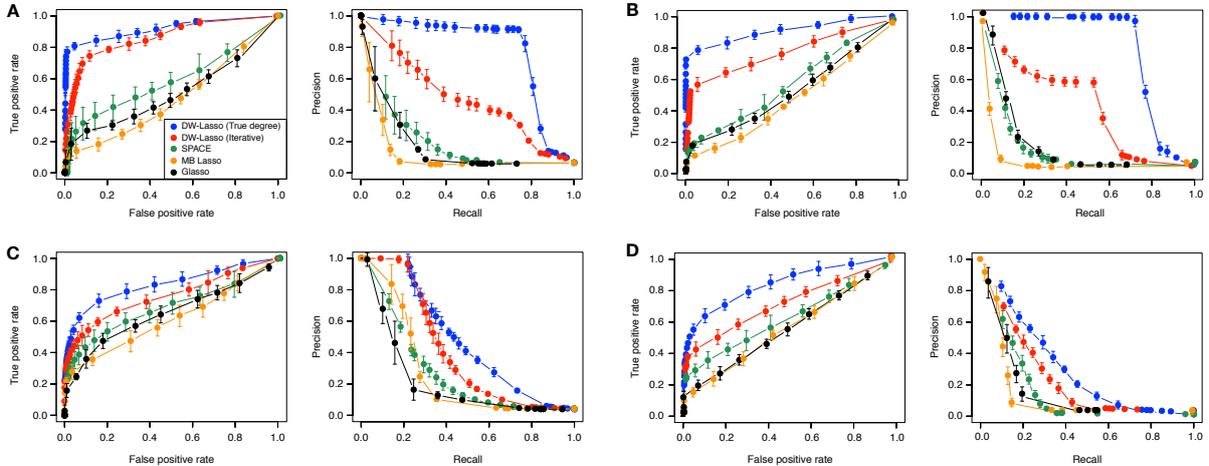


Figure 3: Performance of various methods on (A)-(B) hub and (C)-(D) scale-free networks in the simulation study ((A)-(C) $p = 100$, $n = 50$ and (B)-(D) $p = 500$, $n = 100$). The considered methods are DW-Lasso with true degree information and the proposed iterative procedure, sparse partial correlation estimation (SPACE) (Peng et al. (2009)), graphical Lasso (Glasso) (Friedman et al. (2008)) and nodewise regression (MB-Lasso) (Meinshausen and Bühlmann (2006)). ROC and PR curves for DW-Lasso are computed for fixed values of λ_1 and varying λ_2 . For hub networks, the penalty is chosen as $\lambda_1 = 0.65$ and for scale-free networks as $\lambda_1 = 0.45$ (selected by stability selection). Degree estimates at $k = 30$ iterations are used for the network inference. ROC and PR curves for Glasso and nodewise regression are computed by varying the penalty parameters. Error bars represent two standard deviation of the mean.

guess, while DW-Lasso performs far better than the random guess with AUROC greater than 0.7 (Figure 3B). In the case of scale-free networks, the DW-Lasso also performs better than the competing methods (Figure 3C and Figure 3D). Further simulation results show that DW-Lasso does not perform well for traditional scale-free networks with degree exponent $\gamma = 3$, which results in networks that are extremely sparse. In these networks, the number of edges is roughly equal to the number of genes and the graph can be efficiently reconstructed using the traditional methods.

3.3 Convergence of degree estimates of DW-Lasso

One main feature of DW-Lasso is the convergence of degree (weight) estimates. For a carefully chosen λ_1 as described in Section 2.6, most of the time, the DW-Lasso weight estimates converge close to the weights determined by the true graph (Figure 4B and Figure 5). The performance of DW-Lasso improves with the number of iterations and achieves

the best estimates in terms of AUROC at steady state (Figure 4C). For prediction accuracy, the convergence of the estimates is highly dependent on the choice of the penalty parameter λ_1 . In particular, if λ_1 is chosen outside of the high performance region (see Section 3.5), the weight estimates of DW-Lasso converge to different steady states with low prediction accuracy.

3.4 Effect of the number of hub genes in the network

We next investigate how the number of hub genes in the network affects the performance of DW-Lasso. Therefore, we consider the network with $p = 100$ genes and create 20 hub networks with different numbers of hubs (95% connectivity) that constitute 2% to 20% of the total genes. We randomly generate datasets of size $n = 50$ and evaluate the performance of the aforementioned methods using AUROC and AUPR scores (Figure 6A). Simulation results demonstrate that the DW-Lasso outperforms other methods both in AUROC and AUPR scores

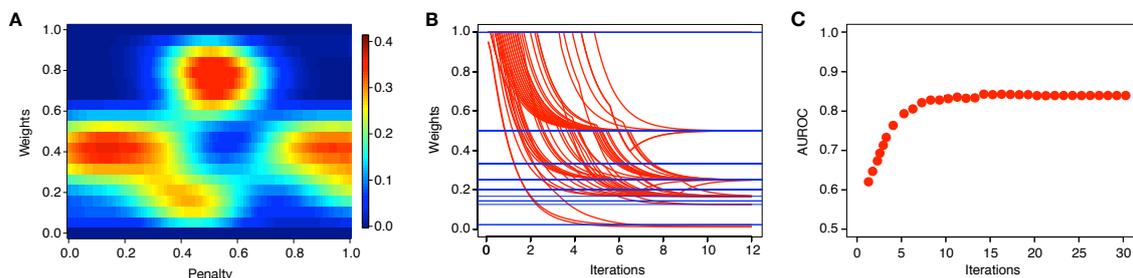


Figure 4: (A) Heatmap of weights computed for different values of penalty parameter using nodewise regression. Initially for small penalty values, all genes have similar weights. For intermediate penalty values, there is a separation of weights between hub and non-hub genes (indicates separation of degrees between hubs and non-hubs). For large penalty values, the network is very sparse and hence all genes have the similar weights. (B), (C) The convergence of the weight $w^{(k)}$ for DW-Lasso method demonstrated for $p = 50$, $n = 40$, $\lambda_1 = 0.5$. (B) The trajectories of weights (red lines) over different iterations that converge to the steady state. Blue lines indicate the true weights computed from the true network. Initially, all weights are same and equal to one. (C) The prediction accuracy of the DW-Lasso increases with the number of iterations.

throughout the considered hub densities. SPACE performs better than Glasso and nodewise regression in AUROC, but exhibits a similar performance in AUPR. As the number of hub genes increases, the performance of DW-Lasso decreases based on AUROC scores, which is also the case for SPACE. Glasso and nodewise regression perform as poorly as the random guess for large number of hub genes. The increase in AUPR score of the state-of-the-art methods, as the number of hub genes increases is trivial, since the sparsity of the graph decreases, which leads to less false positives in the estimates. While the AUPR scores of these methods increase, they perform as poorly as the random guess.

3.5 Influence of penalty parameters

Given the two penalty parameters in our method, we conduct simulations with hub graphs under the setting $p = 60$ and $n = 30$, where for each fixed value of λ_1 , we vary λ_2 and compute AUROC and AUPR scores. Simulation results demonstrate that the optimal AUROC and AUPR scores compared to those of the nodewise regression (dashed line) are obtained for intermediate values of λ_1 (Figure 6B). We call this region the high performance region. The penalty parameter λ_1 chosen via stability selection falls in this region. In this high performance region, the number of false positive edges in the inferred graph is controlled such that the degree

information estimation includes less false positive edges. When λ_1 is large enough, such that the resulting graph is very sparse, the predictions of DW-Lasso is similar to those of the nodewise regression method. Another distinct feature of our method is that it is able to infer graphs with different hub degree controlled by λ_1 . When one increases λ_1 , most of the edges tend to accumulate to a few hub genes, which are also known as super hubs (Hao et al. (2012)). In contrast, decreasing λ_1 leads to graphs containing more hubs of lesser degree.

4 Results on real data

4.1 Kidney Cancer gene expression data

We apply our method to gene expression datasets from TCGA datasets for Kidney Clear Cell Carcinoma (KIRC) measurements (McLendon et al. (2008)). We downloaded the normalized RNA sequencing dataset from UCSC Xena, a data server-based platform that stores functional genomics data (Goldman et al. (2016)). The downloaded samples were measured experimentally using the Illumina HiSeq 2000 RNA Sequencing platform by the University of North Carolina TCGA genome characterization center. First, the genes with missing values were excluded from the data. Genes with low expression levels and those with low interquar-

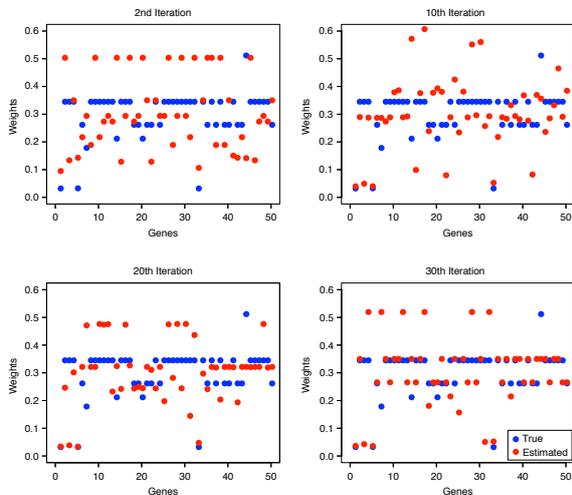


Figure 5: The convergence of the degree weighted Lasso (DW-Lasso). Blue circles indicate the true weights; red circles indicate the predicted weights. The plots indicate that weights converge closely to true weights at later iterations.

tile variability between the samples were discarded. Eventually, the final dataset has been reduced to $p = 5747$ genes and $n = 606$ samples. The gene sets were obtained from Gene Ontology Database (Ashburner et al. (2000)). To ensure that the GO terms were not too thinly or thickly spread, we extract only those gene sets that contain between 10 and 200 genes. This way the total list reduced to 3295 gene sets, which are used in the analysis.

4.2 *E. coli* gene expression data

We apply the DW-Lasso to a well-known public *E. coli* microarray dataset available in Many Microbe Microarrays database (M3D) (Faith et al. (2007)). The M3D includes microarray datasets measured using Affymetrix GeneChip genome arrays. We select the dataset that contains the expression level of 4297 genes from 466 samples. As a gold standard for performance evaluation, we choose experimentally validated interactions from a curated database, RegulonDB (Salgado et al. (2012)) that comprises a high-confidence level set of interactions supported by genome-wide transcription factor data (Harbison et al. (2004)). This database contains the interaction information between transcription factors and genes. To make

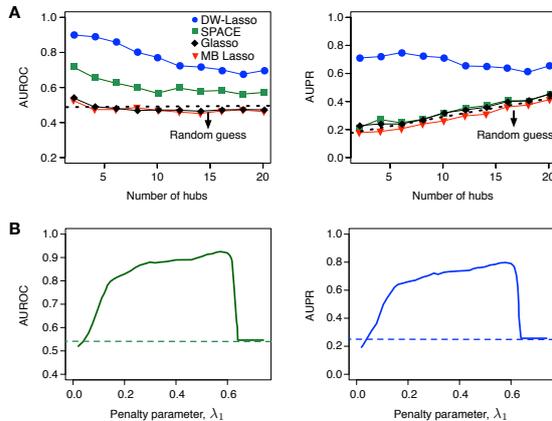


Figure 6: Effect of the different number of hubs and penalty parameters on the performance of DW-Lasso and competing methods. (A) Effect of different hubs on the performance of selected methods. The simulations are performed for hub networks with different number of hubs in the setting $p = 100$, $n = 50$. Black dashed lines indicate random guess. (B) The influence of the penalty parameter λ_1 on the inferred networks. Selection of the penalty parameter λ_1 is based on the AUROC and AUPR computed by varying λ_2 (computed for $k = 30$ iterations). The optimal AUROC and AUPR is attained for the intermediate values of the penalty parameter. Dashed green and blue lines indicate the AUROC and AUPR scores obtained with the nodewise regression.

the list of genes in both microarray and interaction data comparable, we extract 1346 genes that are unique for both datasets. As a result, our gold standard data contains the list of 2069 interactions between 147 transcription factors and 1199 genes. The final microarray dataset used in the study contains 1346 genes and 466 samples.

4.3 Quantifying the functional association between GO terms and inferred networks

Most of publicly available human interaction networks from databases are of low coverage and cannot be used for validation of inferred networks from experimental data. We therefore use an indirect approach and quantify the functional association between publicly available gene sets from Gene On-

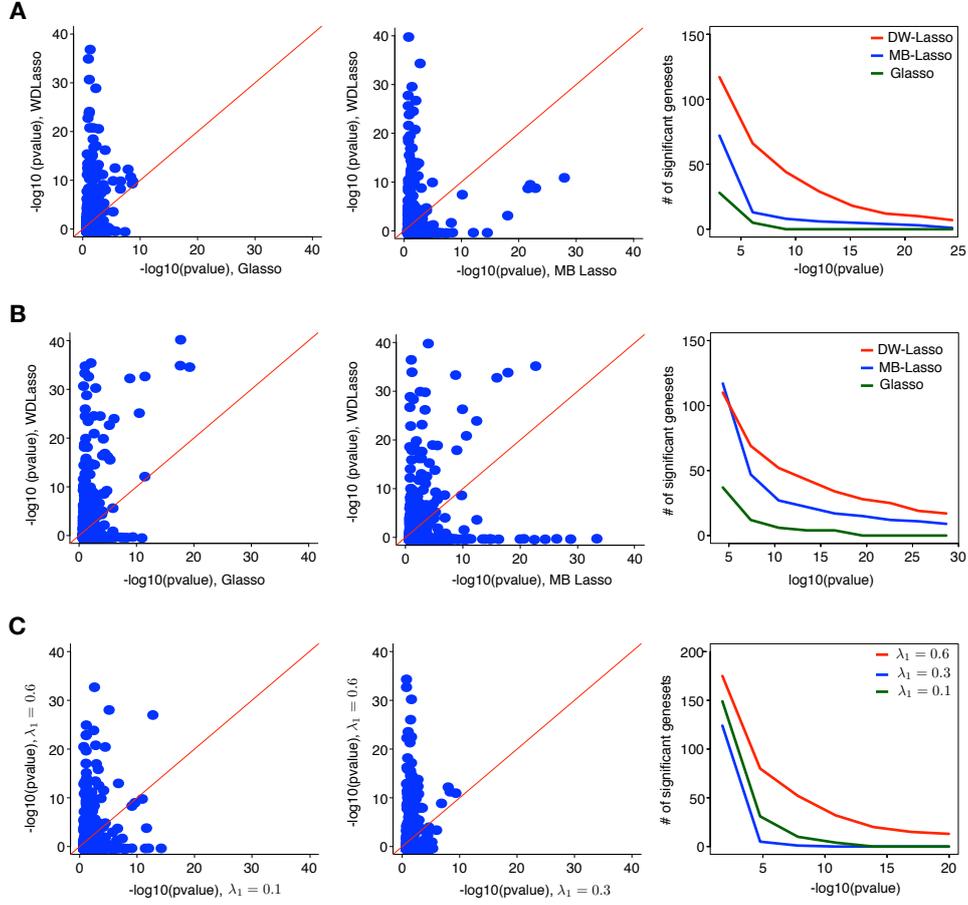


Figure 7: Strength of association between gene sets from GO ontology and networks inferred with DW-Lasso, nodewise regression (MB-Lasso) and Glasso from Kidney Clear Cell Carcinoma ((A) compactness score and (B) Knet score). The comparison of strength of association of gene sets for (left) DW-Lasso and Glasso networks and (middle) DW-Lasso and MB-Lasso networks. (right) Quantification of genesets that significantly cluster on inferred networks. (C) (left) and (middle) Strength of association between gene sets from GO ontology and networks inferred with DW-Lasso with different penalty parameters from Kidney Clear Carcinoma (compactness score). (right) Quantification of genesets that significantly cluster on inferred networks.

tology Database and inferred networks. To quantify this association, we compute two metrics: The compactness score that quantifies the average distance between the genes in the network (Glaab et al. (2010)) and the Knet score implemented in the SANTA package (Cornish and Markowetz (2014)) that takes into account the global topology of the network. More information about these metrics can be found in Section 2.8. We select λ_1 by stability selection and tune λ_2 to infer networks with different sparsities. We then tune the penalty

parameters of nodewise regression and Glasso such that we get the network with the same sparsity as the network inferred by DW-Lasso. Since the selected metrics are highly sensitive against the missed edges among the nodes in the graph, we tune λ_2 so that the resulting graphs have as few isolated nodes as possible. Thus, to allow for a fair comparison, we infer networks that contain 10^5 interactions for 5747 genes. The results show that the 3295 gene sets cluster more strongly on DW-Lasso networks than on networks inferred by nodewise re-

gression (MB-Lasso) and Glasso (Figure 7A (compactness score) and Figure 7B (Knet score)). This indicates that the DW-Lasso networks are functionally more informative than the networks inferred by nodewise regression (MB-Lasso) and Glasso. In addition, we observe that the clustering of gene sets gets better as λ_1 is chosen such that the number of false positive edges is controlled (Figure 7C (compactness score)). This indicates that the underlying network may contain modular structures that include highly connected genes. The top GO terms strongly enriched with DW-Lasso networks are related to chromatin remodeling (histone lysine methylation (p-value $< 1 \times 10^{-24}$), histone H3-K4 methylation (p-value $< 1 \times 10^{-17}$), and NF-KB signaling (p-value $< 1 \times 10^{-10}$). These processes were reported to participate in the development and progression of the kidney cancer (Cancer Genome Atlas Research Network et al. (2013); Ho et al. (2016)). We also observe significant clustering of mitochondrial processes on DW-Lasso networks (mitochondrial respiratory chain (p-value $< 1 \times 10^{-48}$), mitochondrial respiratory chain complex I (p-value $< 1 \times 10^{-11}$) and mitochondrial electron transport (p-value $< 1 \times 10^{-19}$). It was reported that mitochondrial dysfunction is common in cancer and the mitochondrial electron transport is often affected in carcinogenesis (Ellinger et al. (2016)). We next check the list of highly ranked genes by degree in the inferred graph. Our analysis reveals several interesting genes that were reported to have a transcription factor activity. For example, TCF4 was demonstrated to play an important role in the progression of kidney cancer via an inhibition of apoptosis (Shiina et al. (2003)). Another highly ranked inhibitor of apoptosis is HSP70 (Gabbert et al. (2007)). RUNX1 was reported to be upregulated in various kidney cancers and be potentially used as targets for new therapies (Xiong et al. (2014)). One study showed the overexpression of ACAT1 in kidney cancer tissues compared to normal tissues (Osunkoya et al. (2009)). The protein encoded by gene ALDH6A1 was found to be highly expressed in kidney cancer tissues and involved in metabolic processes that are associated with apoptosis and tumorigenesis and thus may play a critical role in renal carcinoma oncogenesis (Perroud et al. (2009)). Another interesting gene is UQCRC1 which was identified as a potential biomarker. Dysregulation of this gene is involved

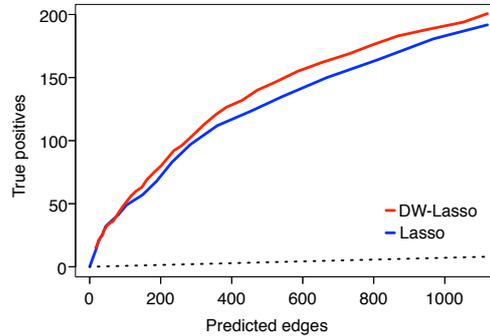


Figure 8: Performance of DW-Lasso and Lasso on *E.coli* data. Illustrated is the number of correctly predicted edges vs total predicted edges. The simulation for DW-Lasso is performed with $\lambda_1 = 0.001$ and varying λ_2 , whereas for Lasso the penalty parameter is varied.

in impaired mitochondrial electron transport chain function (Ellinger et al. (2016)).

4.4 Performance of DW-Lasso on *E.coli* data

We apply DW-Lasso designed for the inference of transcription factor-gene networks (Section 2.5) on *E.coli* data and compare it with the standard Lasso. Since we are interested in reconstructing the network of TF-genes, Glasso and nodewise regression are not relevant in this case. We evaluate the performance of methods based on estimated edges and detected true positive edges. For this purpose, the overall sparsity inducing penalty parameters for DW-Lasso and Lasso are varied until both methods detect 1000 edges. Results show that both methods are able to detect more edges compared to a random guess (Figure 8). In addition, DW-Lasso is able to predict relatively more true interactions compared than Lasso. This indicates that the accounting for hub information is advantageous in improving the accuracy of network inference.

5 Conclusion

We proposed 'DW-Lasso', a new method for the inference of hub graphical models from high-dimensional data. The proposed method consists of two steps and includes two penalty parameters. Tuning these parameters controls the number of

hubs, degree of hubs and overall sparsity of the network. We consider stability selection criteria (Meinshausen and Bühlmann (2010)) to select the penalty parameters. With simulation studies considering hub and scale-free graphs, we demonstrate the increased performance of our method in comparison to traditional methods under the small n large p scenario. Additionally, for kidney cancer data, we show good performance of our method by quantifying the functional content of inferred gene networks with GO terms. Clustering results suggest that gene networks might contain modules consisting of genes centered around hub genes. The advantage of our method over the existing methods is that it can naturally infer hub graphs with various hub sparsities, controlled only by a single penalty parameter. DW-Lasso performs well on both hub network and not-too-sparse scale-free networks. It relies on less constraining assumptions about the network topology compared to that of concurrent methods, although extreme sparsity cannot be accounted for. Although our method is designed for the analysis of gene expression data, it can also be applied to proteomics, metabolomics and in general to any high dimensional data where a underlying hub structure is expected.

Software

The R package is available under the GNU General Public Licence at <https://cran.r-project.org/package=DWLasso>.

Acknowledgements

We would like to thank Wasiur Rahman Khuda Bukhsh and Tim Prangemeier for useful comments and discussions. Part of the computations were performed at the Vital-IT Center for high-performance computing (<http://www.vital-it.ch>) of the SIB (Swiss Institute of Bioinformatics).

Funding

This work has been supported by the e:Bio project HostPathX funded by Federal Ministry of Education and Research (BMBF) and Research, Technology and Development Project grant from Sys-

temsX.ch. HK also acknowledges support from the LOEWE research priority program CompuGene and from the H2020 European project PrECISE.

References

- Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.
- Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- Alexandre Blais and Brian David Dynlacht. Constructing transcriptional regulatory networks. *Genes & development*, 19(13):1499–1511, 2005.
- Cancer Genome Atlas Research Network et al. Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature*, 499(7456):43–49, 2013.
- Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.
- L Collavin, A Lunardi, and G Del Sal. p53-family proteins and their regulators: hubs and spokes in tumor suppression. *Cell death and differentiation*, 17(6):901, 2010.
- Alex J Cornish and Florian Markowetz. Santa: quantifying the functional content of molecular networks. *PLoS Comput Biol*, 10(9):e1003808, 2014.
- Lawrence A Donehower and Guillermina Lozano. 20 years studying p53 functions in genetically engineered mice. *Nature reviews. Cancer*, 9(11):831, 2009.
- Richard Durrett. *Random graph dynamics*, volume 200. Citeseer, 2007.
- Jörg Ellinger, Arabella Gromes, Mirjam Poss, Maria Brüggemann, Doris Schmidt, Nadja

- Ellinger, Yuri Tolkach, Dimo Dietrich, Glen Kristiansen, and Stefan C Müller. Systematic expression analysis of the mitochondrial complex iii subunits identifies uqcrcl as biomarker in clear cell renal cell carcinoma. *Oncotarget*, 7(52):86490, 2016.
- Paul Erdős and Alfréd Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.
- Jeremiah J Faith, Michael E Driscoll, Vincent A Fusaro, Elissa J Cosgrove, Boris Hayete, Frank S Juhn, Stephen J Schneider, and Timothy S Gardner. Many microbe microarrays database: uniformly normalized affymetrix compendia with structured experimental metadata. *Nucleic acids research*, 36(suppl1):D866–D870, 2007.
- Hamed Firouzi and Alfred O Hero. Local hub screening in sparse correlation graphs. In *SPIE Optical Engineering+ Applications*, pages 88581H–88581H. International Society for Optics and Photonics, 2013.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- HE Gabbert, U Ramp, T Shibata, C Mahotka, S Heikaus, MO Grimm, and R Willers. Expression of heat shock protein 70 in renal cell carcinoma and its relation to tumor progression and prognosis. *Histology and histopathology*, 2007.
- Carlo Gaetan, Xavier Guyon, and Kevin Bleakley. *Spatial statistics and modeling*, volume 81. Springer, 2010.
- Enrico Glaab, Anaïs Baudot, Natalio Krasnogor, and Alfonso Valencia. Extending pathways and processes using molecular interaction networks to analyse cancer genome data. *BMC bioinformatics*, 11(1):597, 2010.
- Mary Goldman, Brian Craft, Jingchun Zhu, Teresa Swatloski, Melissa Cline, and David Haussler. The ucsc xena system for integrating and visualizing functional genomics, 2016.
- Chris Hans. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009.
- Dapeng Hao, Cong Ren, and Chuanxing Li. Revisiting the variation of clustering coefficient of biological networks suggests new modular structure. *BMC systems biology*, 6(1):34, 2012.
- Christopher T Harbison, D Benjamin Gordon, Tong Ihn Lee, Nicola J Rinaldi, Kenzie D Macisaac, Timothy W Danford, Nancy M Hannett, Jean-Bosco Tagne, David B Reynolds, Jane Yoo, et al. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99, 2004.
- Alfred Hero and Bala Rajaratnam. Hub discovery in partial correlation graphs. *IEEE Transactions on Information Theory*, 58(9):6064–6078, 2012.
- Steven M. Hill, Laura M. Heiser, Thomas Cokelaer, Michael Unger, Nicole K. Nesser, et al. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature Methods*, 13(4):310–318, 2016.
- Thai H Ho, In Young Park, Hao Zhao, Pan Tong, Mia D Champion, Huihuang Yan, Federico A Monzon, Anh Hoang, Pheroze Tamboli, Alexander S Parker, et al. High-resolution profiling of histone h3 lysine 36 trimethylation in metastatic renal cell carcinoma. *Oncogene*, 35(12):1565–1574, 2016.
- Ben Lehner, Catriona Crombie, Julia Tischler, Angelo Fortunato, and Andrew G Fraser. Systematic mapping of genetic interactions in *caenorhabditis elegans* identifies common modifiers of diverse signaling pathways. *Nature genetics*, 38(8):896, 2006.
- Qiang Liu and Alexander T Ihler. Learning scale free networks by reweighted l1 regularization. In *AISTATS*, pages 40–48, 2011.
- Daniel Marbach, James C. Costello, Robert Küffner, Nicole M. Vega, Robert Prill, et al. Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8):796–804, 2012.
- Roger McLendon, Allan Friedman, Darrell Bigner, Erwin G Van Meir, Daniel J Brat, Gena M Mastrogiannakis, Jeffrey J Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.

- Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Ann. Statist.*, 34(3):1436–1462, 06 2006.
- Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4): 417–473, 2010.
- Adeboye O Osunkoya, Qiqin Yin-Goen, John H Phan, Richard A Moffitt, Todd H Stokes, May D Wang, and Andrew N Young. Diagnostic biomarkers for renal cell carcinoma: selection using novel bioinformatics systems for microarray data analysis. *Human pathology*, 40(12):1671–1678, 2009.
- Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.
- Bertrand Perroud, Tatz Ishimaru, Alexander D Borowsky, and Robert H Weiss. Grade-dependent proteomics characterization of kidney cancer. *Molecular & Cellular Proteomics*, 8(5): 971–985, 2009.
- Heladia Salgado, Martin Peralta-Gil, Socorro Gama-Castro, Alberto Santos-Zavaleta, Luis Muñiz-Rascado, Jair S García-Sotelo, Verena Weiss, Hilda Solano-Lira, Irma Martínez-Flores, Alejandra Medina-Rivera, et al. Regulondb v8. 0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic acids research*, 41(D1):D203–D213, 2012.
- Hiroaki Shiina, Mikio Igawa, Julia Breault, Leopoldo Ribeiro-Filho, Deepa Pookot, Shinji Urakami, Masaharu Terashima, Masao Deguchi, Masaki Yamanaka, Masato Shirai, et al. The human t-cell factor-4 gene splicing isoforms, wnt signal pathway, and apoptosis in renal cell carcinoma. *Clinical cancer research*, 9(6):2121–2132, 2003.
- Kean Ming Tan, Palma London, Karthik Mohan, Su-In Lee, Maryam Fazel, and Daniela M Witten. Learning graphical models with hubs. *Journal of Machine Learning Research*, 15(1):3297–3331, 2014.
- Rashish Tandon and Pradeep Ravikumar. Learning graphs with a few hubs. In *International conference on machine learning*, pages 602–610, 2014.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- Zuquan Xiong, Hongjie Yu, Yan Ding, Chenchen Feng, Hanming Wei, Sha Tao, Dan Huang, Siqun Lilly Zheng, Jieli Sun, Jianfeng Xu, et al. Rna sequencing reveals upregulation of runx1-runx1t1 gene signatures in clear cell renal cell carcinoma. *BioMed research international*, 2014, 2014.
- Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.