HHS Public Access

Author manuscript

Biometrics. Author manuscript; available in PMC 2018 November 27.

Published in final edited form as:

Biometrics. 2017 December ; 73(4): 1221–1230. doi:10.1111/biom.12682.

Learning Gene Regulatory Networks from Next Generation Sequencing Data

Bochao Jia[1,*], **Suwa Xu**[1,*], **Guanghua Xiao**[2,**], **Vishal Lamba**[3], and **Faming Liang**[1,*]

[1]Department of Biostatistics, University of Florida, Gainesville, Florida, U.S.A.

[2]Department of Clinical Sciences, University of Texas, Southwestern Medical Center, Dallas, Taxes, U.S.A.

[3]Department of Pharmacotherapy and Translational Research, University of Florida, Gainesville, Florida, U.S.A.

Summary:

In recent years, next generation sequencing (NGS) has gradually replaced microarray as the major platform in measuring gene expressions. Compared to microarray, NGS has many advantages, such as less noise and higher throughput. However, the discreteness of NGS data also challenges the existing statistical methodology. In particular, there still lacks an appropriate statistical method for reconstructing gene regulatory networks using NGS data in the literature. The existing local Poisson graphical model method is not consistent and can only infer certain local structures of the network. In this paper, we propose a random effect model-based transformation to continuize NGS data, and then we transform the continuized data to Gaussian via a semiparametric transformation and apply an equivalent partial correlation selection method to reconstruct gene regulatory networks. The proposed method is consistent. The numerical results indicate that the proposed method can lead to much more accurate inference of gene regulatory networks than the local Poisson graphical model and other existing methods. The proposed data-continuized transformation fills the theoretical gap for how to transform discrete data to continuous data and facilitates NGS data analysis. The proposed data-continuized transformation also makes it feasible to integrate different types of data, such as microarray and RNA-seq data, in reconstruction of gene regulatory networks.

Keywords

Data-Continuized Transformation; Gaussian graphical model; Gene Regulatory Network; Poisson Graphical Model; RNA-seq

1. Introduction

The emergence of high-throughput technologies has made it feasible to measure the activities of thousands of genes simultaneously, which provides scientists with a major opportunity to infer global gene regulatory networks (GRNs). Accurate inference of global

* jbc409@ufl.edu. ** faliang@ufl.edu.

GRNs is pivotal to gaining a systematic understanding of the molecular mechanism, to shedding light on the mechanisms of diseases that occur when cellular processes are dysregulated, and to further identifying potential therapeutic targets for diseases. Given the high dimensionality and complexity of high-throughput data, inference of global GRNs largely relies on the advance of computational statistical methods.

Gaussian graphical models.

The traditional methods for learning GRNs include Boolean networks, Bayesian networks and differential equation models. See Karlebach and Shamir (2008) for an overview. Since these methods are not scalable, they are usually only applicable to small sets of genes. For large sets of genes, the GRN can be constructed based on Gaussian graphical models (GGMs). The idea underlying GGMs is to use the partial correlation coefficient as a measure of dependency of any two variables (referred to as genes in GRNs). A zero partial correlation coefficient indicates *conditional independence* of the two variables. A variety of methods have been proposed for constructing GGMs from observed data. A popular method is covariance selection (Dempster, 1972), which identifies the non-zero elements in the concentration matrix (i.e., inverse of the covariance matrix) because the non-zero entries in the concentration matrix correspond to conditionally dependent variables. However, this approach cannot be applied to the case of $p > n$, where the sample covariance matrix is singular and thus the concentration matrix can no longer be directly estimated. To tackle this difficulty, regularization methods such as nodewise regression (Meinshausen and Bühlmann, 2006) and graphical Lasso (Yuan and Lin, 2007; Friedman *et al.*, 2008) have been proposed. Nodewise regression uses Lasso (Tibshirani, 1996) as a variable selection method to identify the neighborhood of each variable, and thus the nonzero elements of the concentration matrix. A neighborhood is the set of predictor variables with nonzero coefficients in a regression model estimated separately for each variable. Meinshausen and Bühlmann (2006) showed that this method asymptotically recovers the true graph. To avoid estimating a large number of regressions, Yuan and Lin (2007) proposed to directly estimate the concentration matrix using the regularization method with a l_1-penalty. Soon, this method was accelerated by Friedman *et al.* (2008) using a coordinate descent algorithm that was originally designed for Lasso regression, and this led to the so-called graphical Lasso algorithm. Quite recently, Liang et al. (2015) proposed the ψ-learning method, which works based on an equivalent measure of partial correlation coefficients calculated with reduced conditional sets. The nodewise regression, graphical Lasso and ψ-learning methods can generally work well for Gaussian data, such as the gene expression data measured in DNA microarray.

RNA-seq Data and Poisson Graphical Models.

In recent years, next generation sequencing (NGS) has gradually replaced microarray as the major platform in transcriptome studies, say, through sequencing RNAs (RNA-seq). RNA-seq uses counts of reads to quantify gene expression levels. Compared to microarray data, RNA-seq data have many advantages, such as providing digital rather than analog signals of expression levels, dynamic and wider ranges of measurements, less noise, higher throughput, etc. However, their discreteness also challenges the existing statistical methods. In practice, RNA-seq data are often modeled using Poisson (Sultan et al., 2008) or negative-binomial distributions (Robinson and Oshlack, 2010; Anders and Huber, 2010), but difficulties often

arise in the computation or knowing the properties of the statistics based on these distributions.

Let $\boldsymbol{Y} = (Y_1, \ldots Y_p)$ denote a p-dimensional Poisson random vector associated with a graphical model $\boldsymbol{G}$. It is natural to assume that all the node-conditional distributions, i.e., the conditional distribution of one variable given all other variables, are Poisson with the distribution given by

$$P\left(Y_j \middle| Y_k, \forall k \neq j; \Theta_j\right) = \exp\left[\theta_j Y_j - \log\left(Y_j!\right) + \sum_{k \neq j} \theta_{jk} Y_j Y_k - A\left(\theta_j, \theta_{jk}\right)\right], \quad (1)$$

where $\Theta_j = \{\theta_j, \theta_{jk}, k \neq j\}$, and $A(\theta_j, \theta_{jk})$ is the log-partition function of the Poisson distribution. Following from the Hammersley-Clifford theorem (Besag, 1974), the node-conditional distributions combine to yield the joint Poisson distribution

$$P(\boldsymbol{Y}, \Theta) = \exp\left[\sum_{j=1}^{p}\left(\theta_j Y_j - \log\left(Y_j!\right)\right) + \sum_{j \neq k} \theta_{jk} Y_j Y_k - \phi(\Theta)\right], \quad (2)$$

where $\Theta = (\Theta_1, \ldots, \Theta_p)$ and $\phi(\Theta)$ is the normalizing term ensuring the properness of this distribution. However, the Poisson graphical model suffers from a major caveat: the interaction parameters θ_{jk} must be non-positive for all $j \neq k$ to ensure $\phi(\Theta)$ to be finite and thus the distribution $P(\boldsymbol{Y}; \Theta)$ to be proper (Besag, 1974; Yang et al., 2012). Therefore, the Poisson graphical model only permits negative conditional dependencies, which is a severe limitation in practice. As shown in Patil and Joshi (1968), the negative binomial graphical model also suffers from the same limitation.

To relax this limitation, Allen and Liu (2013) proposed a local Poisson graphical model (LPGM), which ignores the joint distribution of Y_j's, and works by finding a local model for each gene using a regularization method based on the conditional distribution (1) and then defining the network structure as the union of the local models. To account for the high dispersion of the NGS data when the inter-sample variance is greater than the sample mean, Gallopin et al. (2013) proposed a hierarchical log-normal Poisson model which assumes $Y_{ij} \sim Poisson(\lambda_{ij})$ with $\log(\lambda_{ij}) = \sum_{k \neq j} \beta_{jk} \tilde{y}_{ik} + \epsilon_{ij}$ for $j = 1, \ldots, n$, where ϵ_{ij} is a Gaussian random variable, and $\tilde{y}_{ik}$ denotes the standardized, log-transformed data. For each variable Y_j, the local model can be found via a regularization approach for the log-normal Poisson regression. Quite a few related models have been proposed along this direction, including the truncated PGM, quadratic PGM, sub-linear PGM and square-root PGM. Refer to Yang et al. (2013) and Inouye et al. (2016) for the detail. However, these LPGM-based methods are not consistent due to their ignorance of the joint distribution of Y_j's. Without the joint distribution, the conditional dependence $Y_k \not\perp Y_j \middle| Y_{V \setminus \{k, j\}}$ is not well defined and thus the

theoretical basis $Y_k \not\perp Y_j \mid Y_{V\setminus\{k,j\}} \Leftrightarrow \theta_{kj} \neq 0$ and $\theta_{jk} \neq 0$ of nodewise regression (Meinshausen and Bühlmann, 2006; Ravikumar et al., 2009) does not hold, where θ_{kj} and θ_{jk} are defined in (1). Hence, linking the Poisson graphical model to nodewise Poisson regression will not lead to a consistent estimate for the underlying network.

In this paper, we propose a random effect model-based transformation for RNA-seq data. This transformation transforms count data to continuous data, which can then be further transformed to Gaussian data via a semiparametric transformation as prescribed in Liu et al. (2009). Then, we adopt the ψ-learning method developed in Liang et al. (2015) to construct GGMs for the transformed data. Under mild regularity and sparsity conditions, we show that the proposed method is consistent. Transforming count data to continuous data greatly facilitates the analysis of NGS data.

The remainder of this paper is organized as follows. Section 2 describes the random effect model-based transformation, and gives a brief review for the semiparametric transformation of Liu et al. (2009) and the ψ-learning method of Liang et al. (2015). Section 3 illustrates the proposed method using simulated data along with comparisons with gLasso, nodewise regression, LPGM, and some other existing methods. Section 4 applies the proposed method to two real data examples. Section 5 concludes the paper with a brief discussion.

2. Method

The proposed method consists of three steps: (i) data-continuized transformation, (ii) data-Gaussianized transformation, and (iii) ψ-learning, which are described in sequel as follows.

2.1 Data-Continuized Transformation

To continuize the RNA-seq data, we propose a random effect model-based transformation. Let Y_{ij} denote the RNA-seq expression of gene i from subject j for $i = 1,\ldots,p$ and $j = 1,\ldots,n$, where p denotes the number of genes and n denotes the number of subjects. We assume that

$$Y_{ij} \sim Poisson(\theta_{ij}), \quad \theta_{ij} \sim Gamma(\alpha_i, \beta_i), \quad (3)$$

where α_i and β_i are two parameters of the Gamma distribution. It is easy to see that (3) forms a random effect model with the gene-specific random effect modeled by a Gamma distribution. If we integrate out θ_{ij} from the joint distribution $f(Y_{ij}, \theta_{ij} \mid \alpha_i, \beta_i)$, we will have Y_{ij} distributed according to a negative binomial distribution $NB(r; q)$ with $r = \beta_i$ and $q = \alpha_i/(1 + \alpha_i)$. Hence, the model (3) is quite flexible, which accommodates potential overdispersion of the data.

To avoid an explicit specification for the values of α_i and β_i, we conduct a Bayesian analysis for the model. For this purpose, we let α_i and β_i be subject to the prior distributions:

$$\alpha_i \sim Gamma(a_1, b_1), \quad \beta_i \sim Gamma(a_2, b_2),$$

where a_1, b_1, a_2 and b_2 are prior hyperparameters. By assuming that αi and β_i are a priori independent, the full conditional posterior distributions of θ_{ij}, α_i and β_i are given as follows:

$$f\left(\alpha_i \middle| \theta_{ij}, \beta_i, \mathbf{y}_i\right) \propto \frac{\alpha_i^{a_1 - 1}}{\Gamma^n\left(\alpha_i\right)} e^{\alpha_i\left(-b_1 + n\log\beta_i + \sum_{j=1}^{n} \log\theta_{ij}\right)}, \quad (4)$$

$$f\left(\beta_i \middle| \alpha_i, \theta_{ij}, \mathbf{y}_i\right) \propto \beta_i^{n\alpha_i + a_2 - 1} e^{-\beta_i\left(\sum_{j=1}^{n} \theta_{ij} + b_2\right)}$$

$$\propto Gamma(n\alpha_i + a_2, \sum_{j=1}^{n} \theta_{ij} + b_2),$$

$$f\left(\theta_{ij} \middle| \alpha_i, \beta_i, \mathbf{y}_i\right) \propto \theta_{ij}^{y_{ij} + \alpha_i - 1} e^{-\theta_{ij}\left(1 + \beta_i\right)}$$

$$\propto Gamma\left(y_{ij} + \alpha_i, \beta_i + 1\right),$$

where $\mathbf{y}_I = \{y_{ij} : j = 1; 2, \ldots, n\}$. Regarding the choice of prior hyperparameters, we establish the following lemma, whose proof is given in the supplementary material.

Lemma 1: *If a_1 and a_2 take small positive values, then for all i and j, the posterior mean of θ_{ij},* denoted by $E[\theta_{ij}|\mathbf{y}_i]$, *will converge to y_{ij} as $b_1 \to \infty$* and $b_2 \to \infty$.

Suppose that a MCMC algorithm, e.g., the Metropolis-within-Gibbs sampler (Müller, 1993), was used to simulate from the posterior distribution (4). Let $\theta_{ij}^{(t)}$ denote the posterior samples of θ_{ij} for = 1, 2, …, and let $\hat{\theta}_{ij}^{(T)} = \sum_{t=1}^{T} \theta_{ij}^{(t)}/T$ denote the Monte Carlo estimator of $E\left[\theta_{ij} \middle| \mathbf{y}_i\right]$. Then, following from the standard theory of MCMC, we have $\hat{\theta}_{ij}^{(T)} \xrightarrow{p} E\left[\theta_{ij} \middle| y_i\right]$ as $T \to \infty$, where $\xrightarrow{p}$ denotes convergence in probability. To ensure the convergence $\hat{\theta}_{ij}^{(T)} \xrightarrow{p} y_{ij}$ hold in a rigorous manner, the iteration number T and the prior hyperparameters b_1 and b_2 need to go to infinity simultaneously. To achieve this goal, we let b_1 and b_2 increase with iterations. Let $b_1^{(t)}$ and $b_2^{(t)}$ denote the respective values of b_1 and b_2 taken at iteration t, and we set

$$b_1^{(t)} = b_1^{(t-1)} + \frac{c}{t^\zeta}, \; b_2^{(t)} = b_2^{(t-1)} + \frac{c}{t^\zeta}, \; t = 1, 2, \ldots, \quad (5)$$

where $b_1^{(0)}$ and $b_2^{(0)}$ are fixed large constants, $c > 0$ is a small constant, and $0 < \zeta \leq 1$. Under this setting, the MCMC sampler for (4) forms an adaptive Markov chain for which the target distribution gradually shrinks toward a Dirac delta measure defined on $(\alpha_i, \beta_i, \theta_{ij}) = (0; 0; y_{ij})$. For simplicity in theoretical development (see supplementary material), we assume that a random walk proposal is used in simulating from the conditional posterior distribution

$f(\alpha_i|\cdot)$, i.e., the proposal distribution $q(\alpha_i'|\alpha_i^{(t)}) = q(|\alpha_i' - \alpha_i^{(t)}|)$ depends on $|\alpha_i' - \alpha_i^{(t)}|$ only. In summary, we have the following lemma, whose proof is given in the supplementary material.

Lemma 2: *If a random walk proposal is used in simulating from* $f(\alpha_i|\cdot)$ *and the prior hyperparameters are chosen in (5), then* $\hat{\theta}_{ij}^{(T)} \xrightarrow{p} y_{ij}$ *for all i and j as* $T \to \infty$, *where* $\hat{\theta}_{ij}^{(T)} = \sum_{t=1}^{T} \theta_{ij}^{(t)}/T$ *and* $\theta_{ij}^{(t)}$ *denotes the posterior sample of* θ_{ij} *generated at iteration t.*

Lemma 2 implies that the statistical inference for y_{ij}'s can be approximately made using $\hat{\theta}_{ij}^{(T)}$'s as $T \to \infty$. The validity of the approximation can be argued as follows: Let $\mathbb{F}(\hat{\theta}_1^{(T)}, \ldots, \hat{\theta}_p^{(T)})$ denote the empirical CDF of p-continuized random variables. Let $\mathbb{F}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p)$ denote the empirical CDF of $(Y_{1,\ldots,} Y_p)$. It is easy to see that the convergence $\hat{\theta}_{ij}^{(T)} \xrightarrow{p} y_{ij}$ implies $\sup_{\boldsymbol{t} \in \mathbb{R}^p} \left\| \mathbb{F}_{\hat{\theta}_1^{(T)}, \ldots, \hat{\theta}_p^{(T)}}(\boldsymbol{t}) - \mathbb{F}_{y_1, \ldots, y_p}(\boldsymbol{t}) \right\| \xrightarrow{p} 0$ as $T \to \infty$. Further, as the sample size $n \to \infty$, $\sup_{\boldsymbol{t} \in \mathbb{R}^p} \left\| \mathbb{F}_{y_1, \ldots, y_p}(\boldsymbol{t}) - F_{Y_1, \ldots, Y_p}(\boldsymbol{t}) \right\| \xrightarrow{a.s.} 0$ holds under some regularity and sparsity conditions, where $F_{Y1,\ldots,Yp}(\boldsymbol{t})$ denotes the CDF of Y_i's, and $\xrightarrow{a.s.}$ denotes almost sure convergence. For example, we can assume that for each Y_i, the number of variables that Y_i depends on is upper bounded by $n/\log(n)$. In summary, we have $\sup_{\boldsymbol{t} \in \mathbb{R}^p} \left\| \mathbb{F}_{\hat{\theta}_p^{(T)}, \ldots, \hat{\theta}_p^{(T)}}(\boldsymbol{t}) - F_{Y_1, \ldots, Y_p}(t) \right\| \xrightarrow{p} 0$ as $T \to \infty$ and $n \to \infty$, which implies that a consistent estimate can be formed based on the continuized data for each conditional probability used for inference of the network structure underlying $Y_{1,\ldots,} Y_p$. That is, the conditional independence relations among $Y_{1,\ldots,} Y_p$ can be learned from the continuized data $\hat{\theta}_1^{(T)}, \ldots, \hat{\theta}_p^{(T)}$ in a consistent manner.

2.2 Data Gaussianized transformation

Since GGMs have been extensively studied, we seek for a transformation that transforms the continuized data to be Gaussian, while maintaining the conditional independence relations among the variables. The semiparametric Gaussian copula transformation, the so-called nonparanormal transformation, proposed by Liu et al. (2009) satisfies this requirement. It can be described as follows.

Let $X = (X_{1,\ldots,} X_p)^T$ be a continuous p-dimensional random vector. It is said that X has a nonparanormal distribution if there exist functions $\{f_j\}_{j=1}^{p}$ such that $Z = f(X) \sim N(\mu, \Sigma)$, where $f(X) = (f_1(X_1), \ldots, f_p(X_p))^T$. We write $X \sim NPN(\mu, \Sigma, f)$. It is known that if f_j's are monotone and differentiable, the joint probability density function of X is given by

$$P_X(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(f(x)-\mu)^T \Sigma^{-1}(f(x)-\mu)\right\} \cdot \prod_{j=1}^{p} \left|f_j'(x_j)\right|. \quad (6)$$

Based on this formula, Liu et al. (2009) argued that if $X \sim NPN(\mu, \sum, f)$ and each f_j is monotone and differentiable, then $X_i \perp X_j \left| X_{V\setminus\{i,j\}} \Leftrightarrow Z_i \perp Z_j \right| Z_{V\setminus\{i,j\}}$. With the similar argument, we can have that for any triple of disjoint sets $A, B, C \subseteq V, X_A \perp X_B \left| X_C \Leftrightarrow Z_A \perp Z_B \right| Z_C$. In other words, the nonparanormal transformation preserves the conditional independence structure of the original graphical model formed by X. Liu et al. (2009) further showed that $f_j(x) = \mu_j + \sigma_j \Phi^{-1}(F_j(x))$ is such a monotone and differentiable transformation, where μ_j is the mean of X_j, σ_j^2 is the variance of X_j, and $F_j(x)$ is the CDF of X_j. For the high dimensional case where p is greater than and can increase with n, $F_j(x)$ can be replaced by a truncated or Winsorized estimator of the marginal empirical distribution of X_j in order to reduce the variance of the estimate.

2.3 ψ-Learning for Gaussian Graphical Models

There are several methods for learning the structure of Gaussian graphical models, such as gLasso, nodewise regression, and ψ-learning. In this paper ψ-learning is adopted, which, as shown in Liang et al. (2015), tends to have better numerical performance and less CPU cost than gLasso and nodewise regression. The ψ-learning method consists of three steps:

(a) (Correlation screening) Determine the neighborhood for each vertex (or variable) X_i.

- **(i)** Conduct a multiple hypothesis test to identify the pairs of variables for which the empirical correlation coefficient is significantly different from zero. This step results in a so-called empirical correlation network.
- **(ii)** For each vertex X_i, identify its neighborhood in the empirical correlation network, and reduce the size of the neighborhood to $O(n/\log(n))$ by removing the variables having lower correlation (in absolute value) with X_i. This step results in a so-called reduced correlation network.

(b) (ψ-calculation) For each pair of vertices i and j, identify a separator S_{ij} based on the reduced correlation network resulted in step (a) and calculate $\Psi_{ij} = \rho_{ij|S_{ij}}$, where $\rho_{ij|S_{ij}}$ denotes the partial correlation coefficient of X_i and X_j conditioned on the variables $\{X_k : k \in S_{ij}\}$. For a pair of vertices $i \neq j$, a set of vertices is called a separator of i and j if all paths from vertex i to vertex j have at least one vertex in the set.

(c) (ψ-screening) Conduct a multiple hypothesis test to identify the pairs of vertices for which ψ_{ij} is significantly different from zero, and set the corresponding element of the adjacency matrix to be 1.

Under mild conditions, Liang *et al.* (2015) showed that the ψ-partial correlation coefficient is equivalent to the true partial correlation coefficient in determining the structure of GGMs in the sense that

$$\Psi_{ij} = 0 \Leftrightarrow \rho_{ij|V\setminus\{i,j\}} = 0, \quad (7)$$

where $\rho_{ij|V\setminus\{i,j\}}$ denotes the partial correlation coefficient of X_i and X_j conditioned on all other variables in the set V. As implied by (7), the key to the success of the ψ-learning method is that it has reduced the computation of partial correlation coefficients from a high dimensional problem to a low dimensional problem. In general, the cardinality of the set $V\setminus\{i,j\}$ can be much higher than the sample size n, while the cardinality of S_{ij} is upper bounded by $O(n/log(n))$. As shown in Liang et al. (2015), the Ψ-learning method is consistent, i.e., the network produced by it will converge to the true one as the sample size $n \to \infty$.

The multiple hypothesis tests involved in the correlation screening and Ψ-screening steps can be done using an empirical Bayes method developed in Liang and Zhang (2008). The advantage of this method is that it allows for the general dependence between test statistics. Other multiple hypothesis tests which accounts for the dependence between test statistics, e.g., Benjamini et al. (2006), can also be applied here. The performance of multiple hypothesis tests depend on their significance levels. Following the suggestion of Liang et al. (2015), we set the significance level of correlation screening to be $\alpha_1 = 0{:}2$ and that of Ψ-screening to be $\alpha_2 = 0{:}05$. In general, a high significance level of correlation screening will lead to a slightly large separator set S_{ij}, which reduces the risk of missing some important variables in the conditioning set. Including a few false variables in the conditioning set will not hurt much the accuracy of Ψ-partial correlation coefficients.

2.4 Consistency

In summary, the proposed method consists of three steps: (i) data-continuized transformation, (ii) data-Gaussianized transformation, and (iii) Ψ-learning for Gaussian graphical models. From Lemma 2 and the followed arguments, we can conclude that the network structure of $Y_{1,\ldots,}Y_p$ can be consistently learned from the continuized data $\hat{\theta}_1^{(T)},\ldots,\hat{\theta}_p^{(T)}$. Liu et al. (2009) showed that the data-Gaussianized transformation preserves the network structure underlying the data, and Liang et al. (2015) showed that the Ψ-learning method is consistent in recovering the underlying network structure. Therefore, the consistency also holds for the proposed method; that is, the true gene regulatory relations can be recovered from the RNA-seq data using the proposed method when the sample size becomes large.

3. Simulation Studies

To illustrate the performance of the proposed method, we consider some simulation examples with the known conditional independence structure. Since the most NGS data tend to be zero-inflated and highly over-dispersed, the data were simulated from a multivariate zero-inflated negative binomial (ZINB) distribution. The ZINB distribution contains three parameters, λ, k and ω, which controls its mean, dispersion and degree of zero-inflation, respectively. The algorithm developed by Yahav and Shmueli (2012) was adopted to simulate the data, which works via an inverse nonparanormal transformation as follows:

(a) Simulate a random sample of n multivariate Gaussian random variables with the known concentration matrix. Denote the random sample by $(X_{1,\ldots,}X_p)$, where each variable $X_i = (X_{i1,\ldots,}X_{in})^T$ consists of n realizations.

(b) For each variable X_i, find its empirical CDF based on the n realizations and calculate the cumulative probability value for each realization X_{ij}.

(c) Generate a random sample of n zero-inflated negative binomial random variables with pre-specified parameters λ, k and ω by inverting the cumulative probability values obtained in (b).

In our simulations, we set the concentration matrix as follows:

$$C_{i,j} = \begin{cases} 0.5, & \text{if} |j-i| = 1, i = 2, \ldots, (p-1), \\ 0.25, & \text{if} |j-i| = 2, i = 3, \ldots, (p-2), \\ 1, & \text{if } i = j, i = 1, \ldots, p, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

This matrix has been used by quite a few authors to demonstrate their GGM algorithms, say, Yuan and Lin (2007), Mazumder and Hastie (2012), and Liang et al. (2015). To make the simulation similar to the real world, we set the parameters λ, k and ω of the ZINB distribution to their estimates from a real dataset, Acute myeloid leukemia(AML) mRNA sequencing data, which is available on The Cancer Genome Atlas(TCGA) data portal. We estimated these parameters using the function *"glm.nb"* in R for each gene, and then set the simulation parameters to the medians of the estimates: λ = 515; 743, k = 3:304 and ω = 0:003. For the other parameters, we set n = 100 and p = 200. We then applied the proposed method to the simulated data, which went through the steps of data-continuized transformation, nonparanormal transformation, and ψ-learning. To measure the performance of the method, we plot the precision-recall curve (defined in the supplementary material) in Figure 2, which is drawn by fixing the significance level of correlation screening to α_1 = 0.2 and varying the value of α_2, the significance level of ψ-screening.

To conduct the data-continuized transformation, the Metropolis-within-Gibbs sampler was run for 10000 iterations for this dataset, where the first 1000 iterations were discarded for the burn-in process and the remaining iterations were used for inference. The total CPU time cost by the sampler was 39.0 seconds on a personal computer with 2.8GHz Intel Core i7. On average, it cost less than 0.2 seconds per variable. For this transformation, we set α_1 = α_2 =

1, $b_1^{(0)} = b_2^{(0)} = 10000$, $c = 1$ and $\varsigma = 1$, the default setting of the prior hyperparameters used throughout the paper. The left panel of Figure 1 shows the scatter plot of the continuized data versus raw counts for one variable, and the right panel shows the Q-Q plot of the Gaussianized data for the variable. The scatter plot indicates that the continuized data and the raw counts are very close to each other. To have a thorough exploration for the data-continuized transformation, we reported in Table 2 of the supplementary material the posterior mean and standard deviation of α_i, β_i, θ_{ij} and the AUC value, i.e., the area under the precision-recall curve, for measuring the performance of the proposed method. The results indicate again that θ_{ij} can be very close to y_{ij} and our method is robust to the choice of $(a_1, a_2, b_1^{(0)}, b_2^{(0)})$. The data-continuized transformation does not lose much information of the raw counts.

For comparison, we have applied the existing methods, including gLasso, nodewise regression, Local Poisson Graphical Model (LPGM), Truncated Poisson Graphical Model (TPGM) and Sublinear Poisson Graphical Model (SPGM) to the simulated data. For gLasso and node-wise regression, the simulated ZINB data first went through the logarithm transformation and nonparanormal transformation, which have been widely used in RNA-seq data analysis, and then the methods were applied. The gLasso and nodewise regression methods have been implemented in the *R*-package huge (Zhao et al., 2015). In our applications, the stability approach was used to determine their regularization parameters. The stability approach selects the network with the smallest amount of regularization that simultaneously makes the network sparse and replicable under random sampling. For LPGM, we used the method proposed by Allen and Liu (2013). For SPGM and TPGM, we used the method proposed by Yang et al. (2013). The three methods have been implemented in the *R*-package *XMRF* (Wan et al., 2015). Besides these existing method, we also compared the proposed method with the one without data-continuized process, i.e., ψ-learning with logarithmic and non-paranormal transformations, which is labeled as "Log +NPN+ ψ-Learning" in Figure 2.

The comparison indicates that the proposed method significantly outperforms other methods, although the improvement mainly comes from ψ-learning. The data-continuized transformation does not loss the information of the data, and it provides a justification for the empirical use of treating log-NGS data as continuous. Multiple datasets have been tried, the results are very similar. Note that LPGM is an extension of the nodewise regression method (Meinshausen and Bühlmann, 2006) to multivariate Poisson. Both the LPGM and nodewise regression methods are based on the idea of neighborhood selection. This experiment also shows that the data-continuized transformation and nonparanormal transformation improves the performance of the neighborhood selection method. Based on this experiment, we suspect that the graph consistency established in Meinshausen and Bühlmann (2006) for nodewise normal regression might not hold for LPGM.

We have also considered several common network structures such as hub, scale-free, small-world and random. The multivariate Gaussian random variables given these structures can be generated by functions provided in *"huge"* package. Then we continue steps (b) and (c) of Yahav and Shmueli's algorithm to get ZINB samples with the same parameters as used

before, i.e., $(n, p) = (100, 200)$, $\lambda = 515; 743$, $k = 3{:}304$ and $\omega = 0{:}003$. The results are summarized in Figure 3. It shows that the proposed method significantly outperforms all other methods for the scale-free, small world and random structures, and performs similarly to gLasso and nodewise regression for the hub structure. To have a thorough comparison with the existing methods, we also considered the scenario of $n > p$ with the results reported in the supplementary material.

4. Real Data Examples

4.1 Liver cytochrome P450s subnetwork

Liver cytochrome P450s play critical roles in drug metabolism, toxicology, and metabolic processes. They form a superfamily of monooxygenases critical for anabolic and catabolic metabolism in all organisms characterized so far (Nelson et al., 1996; Aguiar et al., 2005; Plant, 2007). Specifically, P450 enzymes are involved in the metabolism of various endogenous and exogenous chemicals, including steroids, bile acids, fatty acids, eicosanoids, xenobiotics, environmental pollutants, and carcinogens (Ortiz, 2005). Through experimental work, Yang et al. (2010) determined the human liver transcriptional network structure, uncovered subnet-works representative of the P450 regulatory system, and identified novel candidate regulatory genes. Our goal is to recover the P450s gene regulatory subnetwork, as shown in the left panel of Figure 4, using the RNA-seq data generated at Dr. Lamba's lab. In the plot, the P450 genes and the known P450 regulators are highlighted as red circles and blue squares, respectively.

The original dataset consisted of 100 samples, and each sample consisted of 22337 genes. In our study, we only considered the genes shown in the left panel of Figure 4. The genes "AK097548s", "BC019583", "ENST00000301162" and "NM 173466" have been excluded from our study, as they are not protein-coding genes and their expression data are not available in the original dataset. According to the proposed method, we first applied the data-continuized transformation to the RNA-seq data. After the data-continuized transformation, we adjusted some effects that potentially affect the distribution of the data, including the age, gender and batch of data collection, through linear regression. Then, we applied the nonparanormal transformation and Y-learning method to the adjusted data. Figure 4 shows the resulting subnetwork.

The subnetwork published in Yang et al. (2010) contains 48 genes, and the subnetwork produced by the proposed method contains 26 genes which are connected to some other genes. Although the two subnetworks contain different numbers of genes, they share very similar relations for gene regularations. For example, in the subnetwork by Yang et al. (2010), the gene GLYAT connects to the genes ZGPAT, ETNK2, and AKR1D1; gene HAAO connects to gene CYP27A1; gene CYP2A7 connects to gene CYP2A13; gene CLU connects to SLC27A5; gene ACSM3 connects to EHHADH, and gene CYP4F2 connects to gene SLC16A2. All these connections have been recovered in our subnetwork. Although the rest connections in the two subnetworks do not match exactly, they show some similar dependence. For example, gene CYP2A7 connects to both CYP2A6 and CYP2A13 in the subnetwork by Yang et al. (2010), our subnetwork also shows that they are dependent. This example indicates the validity of the proposed method.

4.2 Acute myeloid leukemia mRNA sequencing network

This example illustrates the performance of the proposed method in the small-n-large-p scenario. The dataset is the mRNA sequencing data from acute myeloid leukemia (AML) patients and available at The Cancer Genome Atlas(TCGA) data portal (http://cancergenome.nih.gov/). In this study, we directly worked on the raw count data, which contains 179 patients and 19990 genes. In preprocessing the data, we filtered out some low expression genes: we first excluded the genes with at least one zero count, and then selected 500 genes with the largest inter-sample variance as suggested by Gallopin et al. (2013). The selected genes are more likely linked to the development of acute myeloid leukemia as their expression levels are highly variable.

Figure 5 shows the GRN produced by the proposed method for the AML RNA-seq data. Through this network, we can identify some hub genes that are likely related to AML. A hub gene refers to a gene which has strong connectivity to other genes. Our finding is pretty consistent with the existing knowledge. For example, the hub gene MKI67 is a well known tumor proliferation marker. The prognostic value of the MKI67 protein expression has been reported for many types of malignant tumors including brain, breast, and lung cancer, with only a few exceptions for certain types of tumors (Mizuno et al., 2009). Another example is the gene KLF6. Humbert et al. (2011) showed the expression patterns of KLFs with a putative role in myeloid differentiation in a large cohort of primary AML patient samples, CD34+ progenitor cells and granulocytes from healthy donors. They found that KLF2, KLF3, KLF5 and KLF6 are significantly lower expressed in AML blast and CD34+ progenitor cells compared to normal granulocytes, and that KLF6 is upregulated by RUNX1-ETO and participates in the RUNX1-ETO gene regulation. This finding provides new insights into the under-studied mechanism of RUNX1-ETO target gene upregulation and identifies KLF6 as a potentially important protein for further study in AML development (DeKelver et al., 2013). The biological functions of other hub genes, such as H3F3B and TMC8, will be further studied.

For comparison, gLasso, nodewise regression and LPGM have been applied to this dataset. They were run as for the simulated examples. Nodewise regression and gLasso were run using the package huge under their default setting, but the regularization parameter was determined using the stability approach. LPGM was run using the package XMRF under its default setting. All these methods produced much denser networks than the proposed method. To assess the quality of the networks produced by different methods, the power law curve (see, e.g., Kolaczyk 2009, pp.80–85) was t to them. A nonnegative random variable X is said to have a power-law distribution if

$$P(X = x) \propto \sim x^{-\upsilon}, \quad (9)$$

for some positive constant ν. The power law states that the majority of vertices are of very low degree, although some are of much higher degree. A network whose degree distribution follows the power law is called a scale-free network and it has been verified that many biological networks are scale-free, e.g., gene expression networks, protein-protein

interaction networks, and metabolic networks (Barabási and Albert 1999). Figures 6 show the log-log plots of the degree distributions of the networks generated by four methods, where the curves are fitted by the loess function in R. It shows that the network produced by the proposed method approximately follows the power law, while those by gLasso, nodewise regression and LPGM do not.

5. Discussion

We have proposed a method for learning gene regulatory networks from RNA-seq data. The proposed method is a combination of a random effect model-based data-continuized transformation, the nonparanormal transformation, and the ψ-learning algorithm. The proposed method is consistent in the sense that the true gene regulatory networks can be recovered from the RNA-seq data when the sample size becomes large. The major contribution of this paper lies on the data-continuized transformation, which fills the theoretical gap of how to transform NGS data to continuous data and facilitates learning of gene regulatory networks.

The proposed data-continuized transformation involves an adaptive Markov chain. We proved the convergence and the weak law of large numbers for the adaptive Markov chain under the framework provided by Liang et al. (2016). A strong law of large numbers (SLLN) can potentially be proved for the algorithm under the framework provided by Fort (2011). With the SLLN, some stronger theoretical properties might be obtained for the resulting networks.

In practice, some authors treated the logarithm of the RNA-seq data as continuous, though not rigorous. The proposed method provides a justification for this use, which is necessary and important given the popularity of NGS techniques. As discussed in Liang et al. (2015), the ψ-learning algorithm provides a general framework for how to integrate multiple sources of data in reconstructing Gaussian graphical networks, where it is proposed to use a meta-analysis method to combine the ψ-partial correlation coefficients calculated from different sources of data. Similarly, with the proposed method, we can integrate different types of omics data, such as the RNA-seq and microarray data, to improve inference for gene regulatory networks. We expect that this method will be widely used in the near future.

Finally, we note that alternative to the LPGM method, an existing method that can potentially be used for Poisson graphical modeling is the latent copula Gaussian graphical modeling method (Ho , 2007; Dobra and Lenkoski, 2011). The basic idea of this method is to introduce Gaussian latent variables in place of discrete random variables in the Poisson network inference. Since the method involves imputation for a large number of latent variables, it is very slow and can only be applied to the problems with a small set of genes.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

Liang's research was partially supported by grants DMS-1612924 and R01-GM117597. The authors thank Jingnan Xue for helpful discussions on the paper, and thank the editor, associate editor and three referees for their constructive comments which have led to significant improvement of this paper.

References

Allen G and Liu Z (2013). A local Poisson graphical model for inferring networks from sequencing data. IEEE Transactions on NanoBioscience, 12(3), 189–198. [PubMed: 23955777]

Aguiar M, Masse R, Gibbs BF. (2005). Regulation of cytochrome P450 by post translational modification. Drug Metab, Rev 37, 379–404. [PubMed: 15931769]

Anders S and Huber W (2010). Differential expression analysis for sequence count data. Nature Proceedings 11, R106.

Barabási A and Albert R (1999). Emergence of scaling in random networks. Science, 286, 509. [PubMed: 10521342]

Benjamini Y, Krieger AM, and Yekutieli D (2006). Adaptive linear step-up procedures that control the false discovery rate. Biometrika, 93, 491–507.

Besag J (1974). Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society, Series B, 36, 192–236.

DeKelver RC, Lewin B, Lam K, et al. (2013). Cooperation between RUNX1-ETO9a and novel transcriptional partner KLF6 in upregulation of Alox5 in acute myeloid leukemia[J]. PLoS Genet, 9(10): e1003765. [PubMed: 24130502]

Dempster AP (1972). Covariance selection. Biometrics, 28, 157–175.

Dobra A and Lenkoski A (2011). Copula Gaussian graphical models and their application to modeling functional disability data. Ann. Appl. Statist, 5, 969–993.

Fort G, Moulines E and Priouret P (2011). Convergence of adaptive and interacting Markov chain Monte Carlo algorithms. Annals of Statistics, 39, 3262–3289.

Friedman J, Hastie T and Tibshirani R (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9, 432–441. [PubMed: 18079126]

Gallopin M, Rau A, Ha rzic F (2013). A hierarchical Poisson log-normal model for network inference from RNA sequencing data. PloS One, 8:10.

Genest C, and Neslehova J (2007). A primer on copulas for count data. Austin Bulletin, 37(2), 475–515.

Hastings WK (1970). Monte Carlo sampling methods using Markov chain and their applications. Biometrika, 57, 97–109.

Hoff P (2007). Extending the rank likelihood for semiparametric copula estimation. Ann. Appl. Statist, 1, 265–283.

Humbert M, Halter V, Shan D, et al. (2011). Deregulated expression of Kruppel-like factors in acute myeloid leukemia[J]. Leukemia research, 35(7): 909–913. [PubMed: 21470678]

Inouye DI, Ravikumar P, and Dhillon IS (2016). Square root graphical models: multivariate generalizations of univariate exponential families that permit positive dependencies. In Proceedings of the 33rd International Conference on Machine Learning, W&CP Volume 48.

Karlebach G and Shamir R (2008). Modelling and analysis of gene regulatory networks. Nature Reviews, 9, 770–780.

Kolaczyk ED (2009). Statistical Analysis of Network Data: Methods and Models Springer.

Liang F, Jin IH, Song Q, and Liu JS (2016). An Adaptive Exchange Algorithm for Sampling from Distribution with Intractable Normalizing Constants. Journal of the American Statistical Association, 111, 377–393.

Liang F, Song Q and Qiu P (2015). An Equivalent Measure of Partial Correlation Coefficients for High Dimensional Gaussian Graphical Models. Journal of the American Statistical Association, 110, 1248–1265.

Liang F and Zhang J (2008). Estimating the false discovery rate using the stochastic approximation algorithm. Biometrika, 95, 961–977.

Liu H, Lafferty J and Wasserman L (2009). The Nonparanormal: Semiparametric Estimation of High Dimensional Undirected Graphs. Journal of Machine Learning Research, 10, 2295–2328.

Mazumder R and Hastie T (2012). The graphical lasso: New insights and alternatives. Electronic Journal of Statistics, 6, 2125–2149. [PubMed: 25558297]

Meinshausen N and Bühlmann P (2006). High-dimensional graphs and variable selection with the Lasso. Annals of Statistics, 34, 1436–1462.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, and Teller E (1953). Equation of state calculations by fast computing machines. Journal of Chemical Physics, 21, 1087–1091.

Mizuno H, Kitada K, Nakai K, et al. (2009). PrognoScan: a new database for meta-analysis of the prognostic value of genes. BMC medical genomics, 2(1): 18. [PubMed: 19393097]

Müller P (1993). Alternatives to the Gibbs sampling scheme. Technical Report, Institute of Statistics and Decision Sciences, Duke University.

Nelson DR, Koymans L, Kamataki T, Stegeman JJ, Feyereisen R, Waxman DJ, Waterman MR, Gotoh O, Coon MJ, Estabrook RW, et al. (1996). P450 super-family: Update on new sequences, gene mapping, accession numbers and nomenclature. Pharmacogenetics, 6, 1–42. [PubMed: 8845856]

Ortiz De Montellano PR (2005). Cytochrome P450: Structure, mechanism, and biochemistry Springer, New York.

Plant N (2007). The human cytochrome P450 sub-family: Transcriptional regulation, inter-individual variation and interaction networks. Biochim Biophys Acta, 1770, 478–488. [PubMed: 17097810]

Patil GP and Joshi SW (1968). A Dictionary and Bibliography of Discrete Distributions Hafner, New York.

Ravikumar P, Wainwright M, and Lafferty J (2009). High-dimensional Ising model selection using l1-regularized logistic regression. Annals of Statistics, 38, 1287–1319.

Robinson MD and Oshlack A (2010). A Scaling Normalization Method for Differential Expression Analysis of RNA-seq Data. Genome Biology 11, R25. [PubMed: 20196867]

Sultan M, Schulz M and Richard H (2008). A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. Science, 321, 956–960. [PubMed: 18599741]

Tibshirani R (1996). Regression analysis and selection via the Lasso. Journal of the Royal Statistical Society, Series B, 58, 267–288.

Wan Y-W, Allen GI, Baker Y, Yang E, Ravikumar P, and Liu Z (2015). Package `XMRF': Markov Random Fields for High-Throughput Genetics Data https://cran.r-project.org/web/packages/XMRF/.

Yahav I and Shmueli G (2012). On generating multivariate Poisson data in management science applications. Applied Stochastic Models in Business and Industry, 28(1), 91–102.

Yang E, Ravikumar P, Allen G, and Liu Z (2012). Graphical models via generalized linear models. In Advances in Neural Information Processing Systems 25, 1367–1375.

Yang E, Ravikumar P, Allen G, and Liu Z (2013). On Poisson graphical models. In Neural Information Processing Systems (NIPS), pp.1718–1726.

Yang X, Zhang B, Molony C, Chudin E, Hao K, Zhu J, : : : and Guengerich FP (2010). Systematic genetic and genomic analysis of cytochrome P450 enzyme activities in human liver. Genome research, 20(8), 1020–1036. [PubMed: 20538623]

Yuan M and Lin Y (2007). Model selection and estimation in the Gaussian graphical model. Biometrika, 94, 19–35.

Zhao T, Li X, Liu H, Roeder K, Lafferty J, Wasserman L (2015). Package 'huge': High-Dimensional Undirected Graph Estimation https://cran.r-project.org/web/packages/huge/.

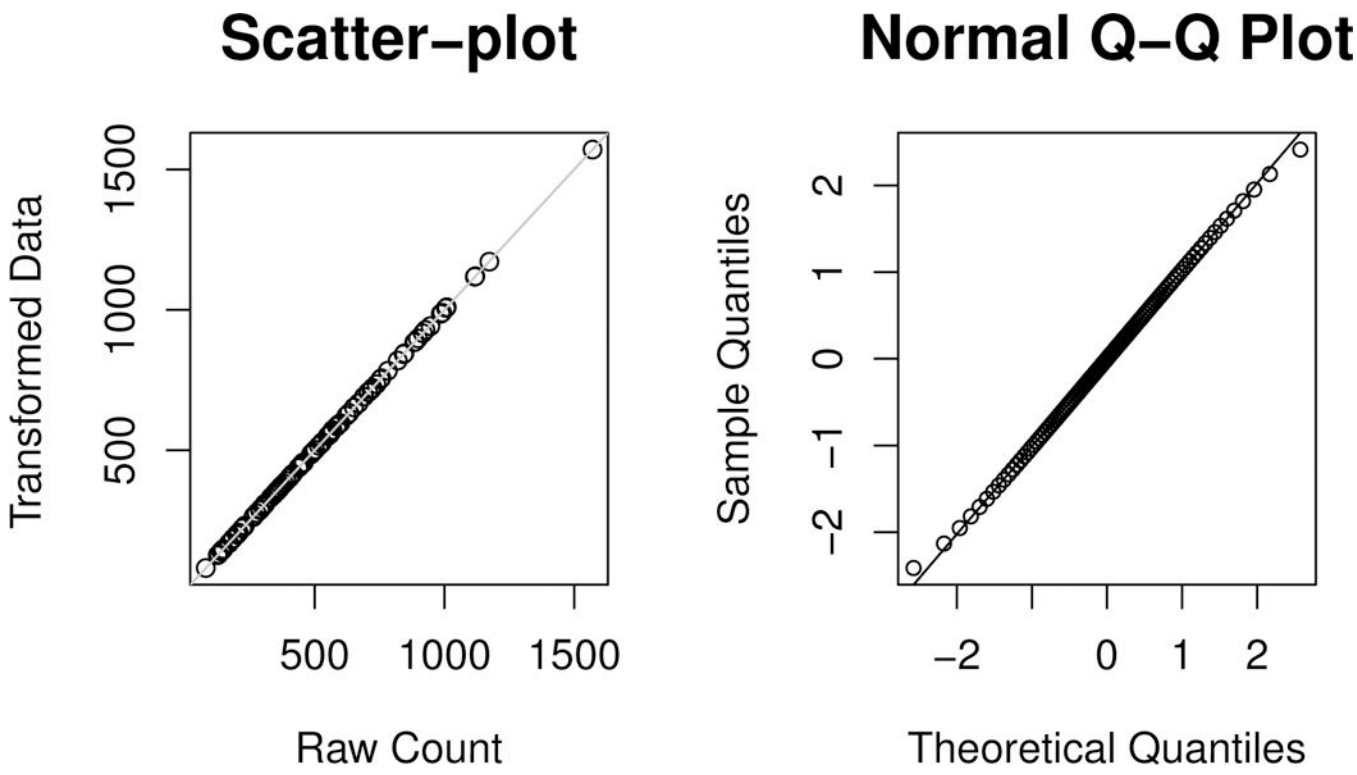

Figure 1.

Left: Scatter plot of the continuized data versus raw counts for one variable. Right: QQ-plot of the Gaussianized data for one continuized variable.

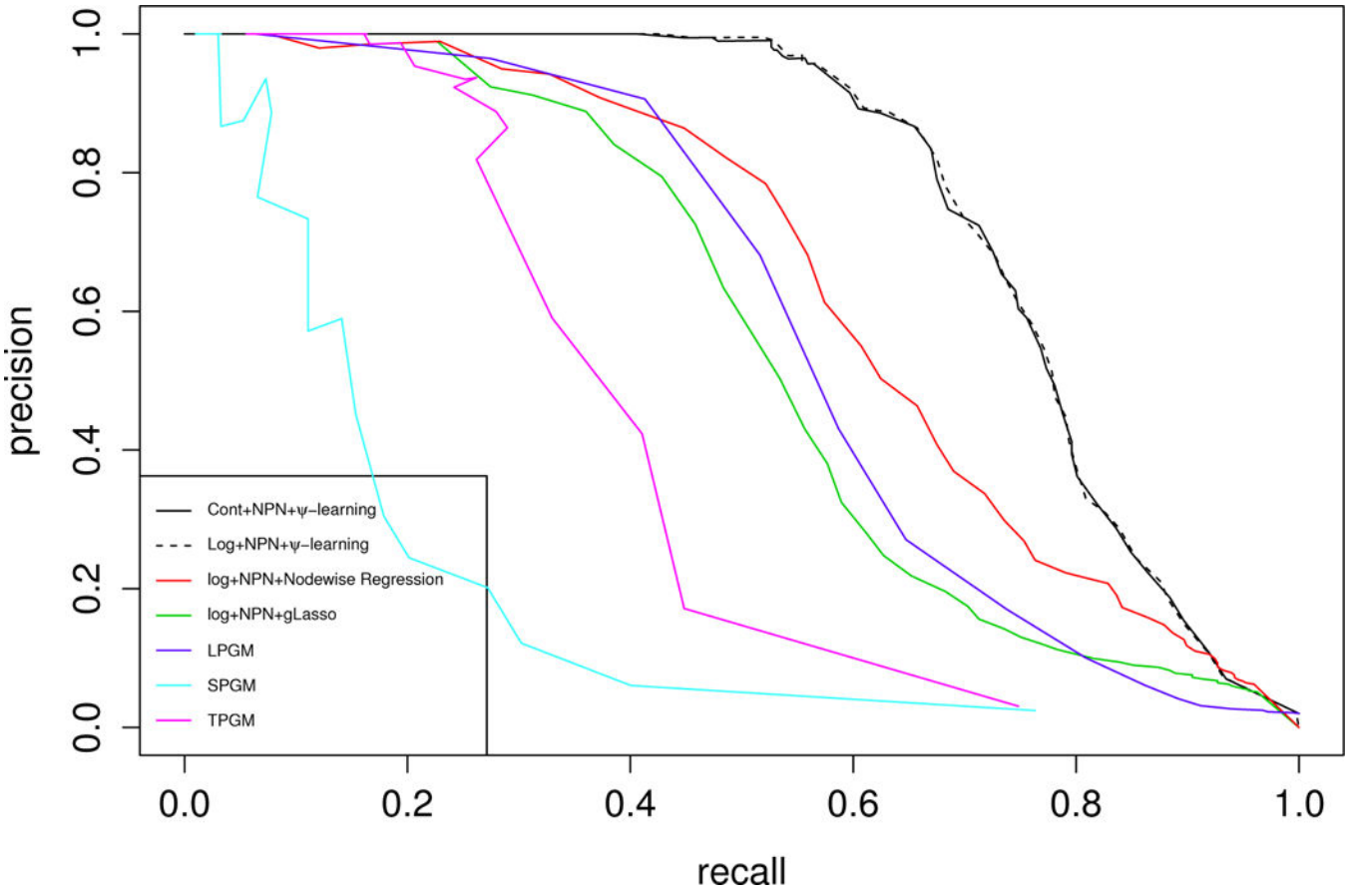

Figure 2.

Precision-recall curves produced by the proposed method (Cont+NPN+ Ψ-learning), log transformation-based Ψ-learning (Log+NPN+ Ψ-learning), log transformation-based gLasso (Log+NPN+gLasso), log transformation-based nodewise regression (Log+NPN+nodewise Regression), LPGM SPGM, TPGM for the simulated data with (n, p) = (100, 200).

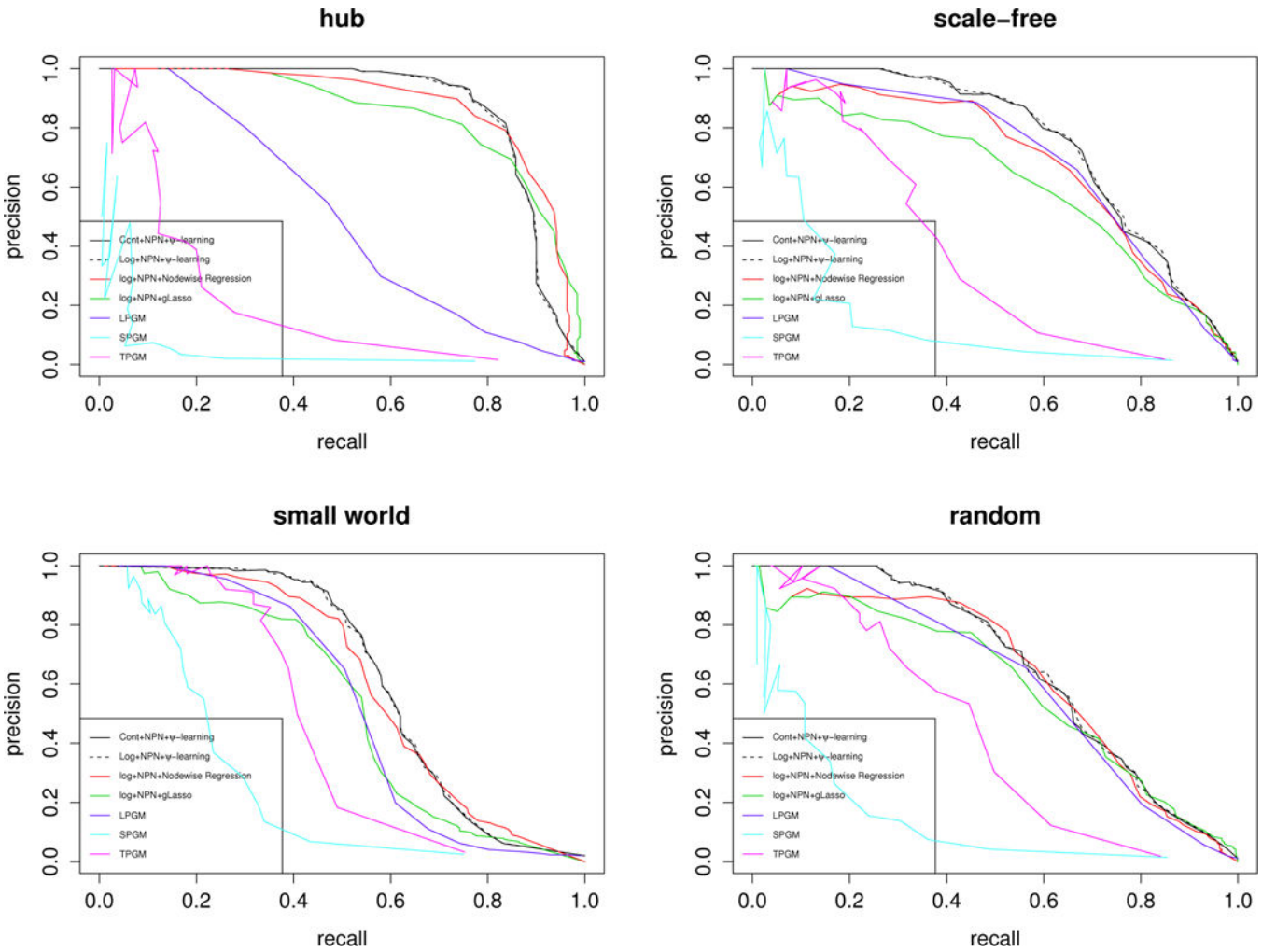

Figure 3.

Precision-recall curves of each method for different type of structures with (n, p) = (100, 200). Upper left: hub; upper right: scale-free; lower left: small-world; lower right: random. Refer to the legend of Figure 2 for the labels.

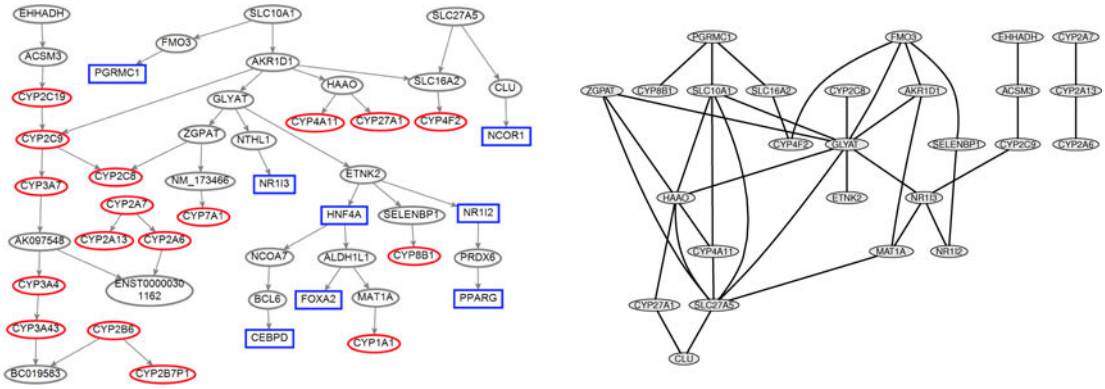

Figure 4.

Left: P450 gene regulatory subnetwork reproduced from Yang et al. (2010), where the known regulators and P450 genes are shown as blue rectangles and red ovals, respectively. Right: the subnetwork produced by the proposed method.

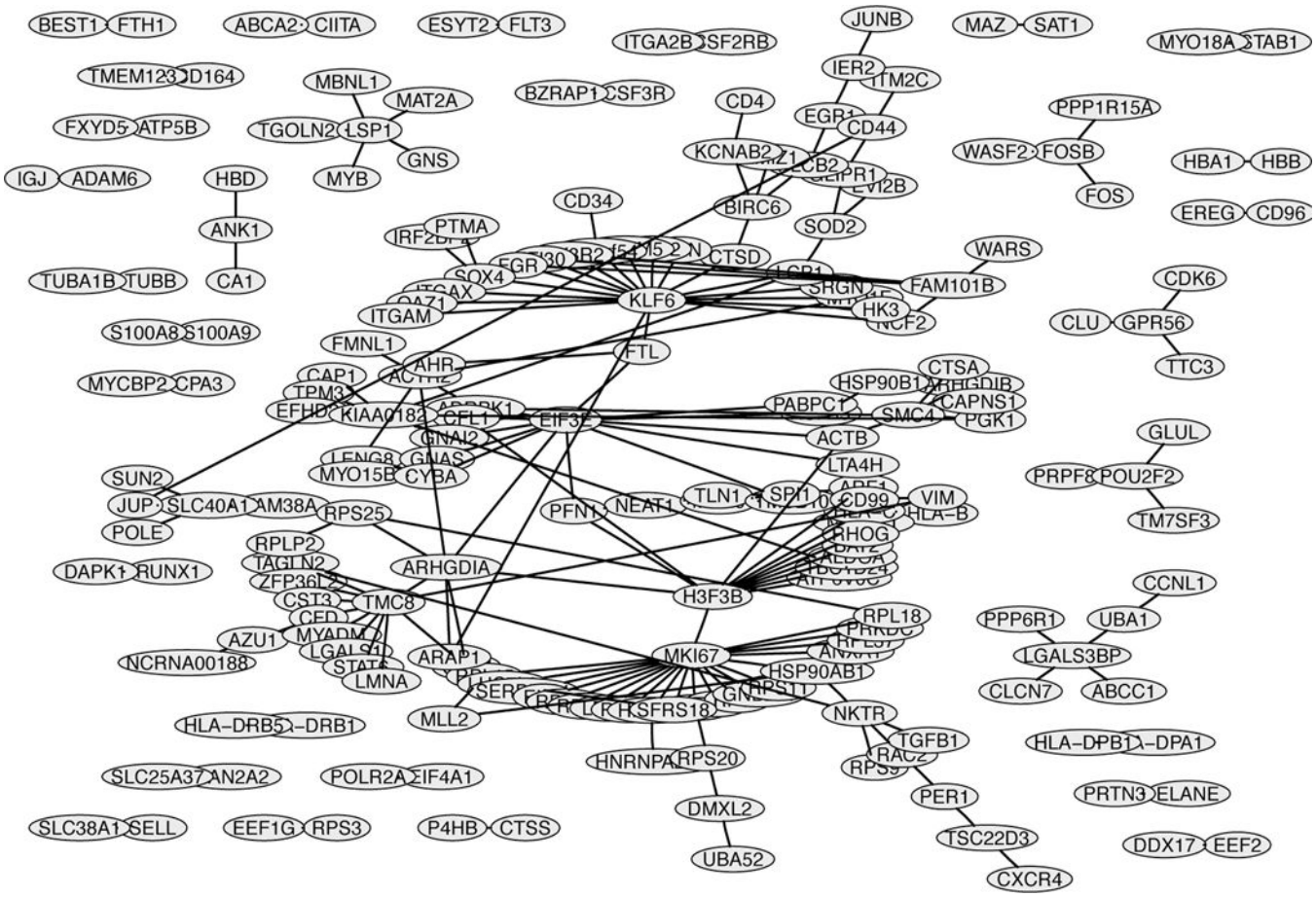

Figure 5.

Gene regulatory network produced by the proposed method for the acute myeloid leukemia RNA-seq data with (n, p) = (179, 500).

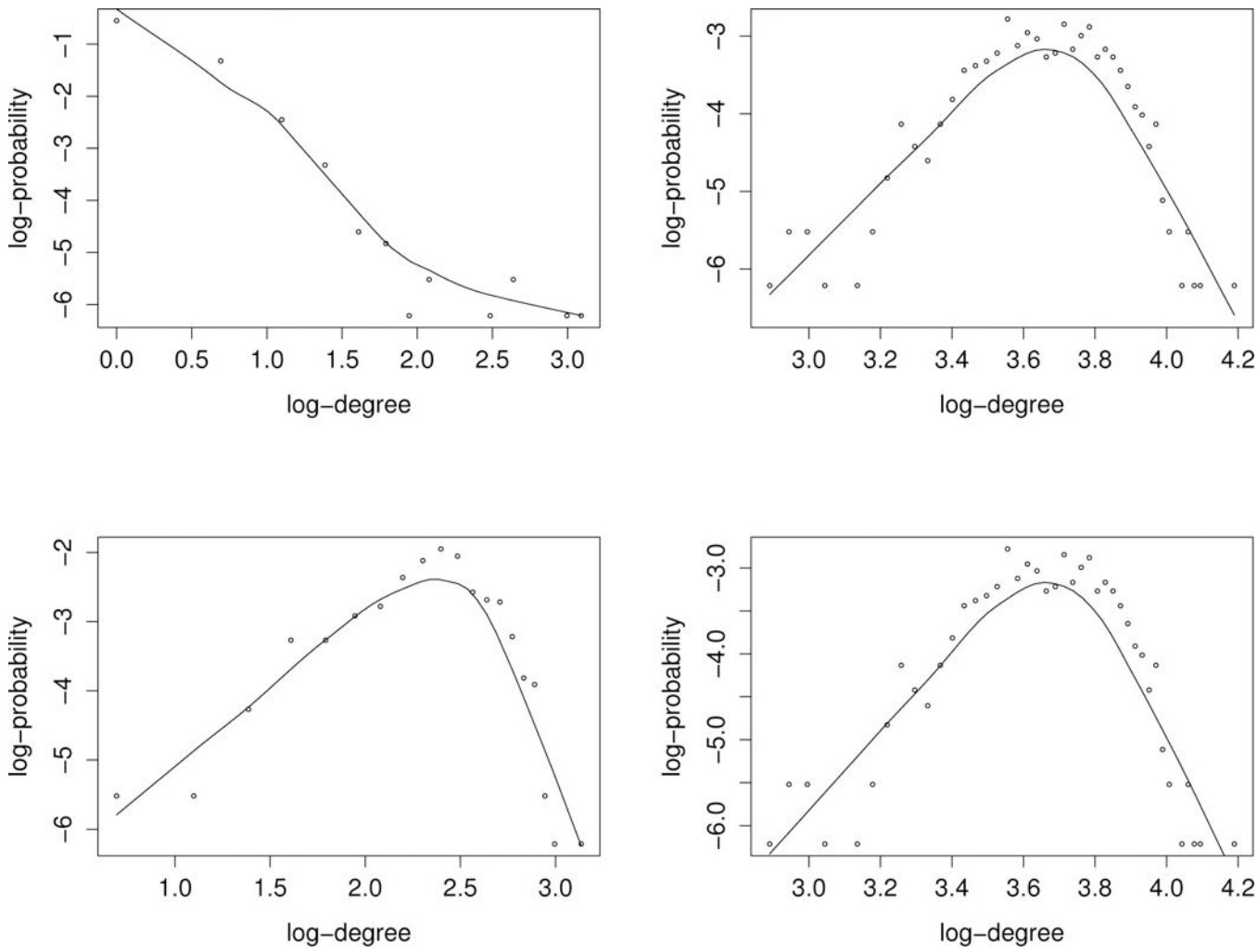

Figure 6.

Log-log plots of the degree distributions of the four networks generated by the proposed method (upper left), gLasso (upper right), nodewise regression (lower left), and LPGM (lower right).