Estimating graphical models for count data with applications to single-cell gene network

Anonymous Author(s)
Affiliation
Address
email

Abstract

Graphical models such as Gaussian graphical models have been widely applied for direct interaction inference in many different areas. In many modern applications, such as single-cell RNA sequencing (scRNA-seq) studies, the observed data are counts and often contain many small counts. Traditional graphical models for continuous data are inappropriate for network inference of count data. We consider the Poisson log-normal (PLN) graphical model for count data and the precision matrix of the latent normal distribution represents the network. We propose a two-step method PLNet to estimate the precision matrix. PLNet first estimates the latent covariance matrix using the maximum marginal likelihood estimator (MMLE) and then estimates the precision matrix by minimizing the lasso-penalized D-trace loss function. We establish the convergence rate of the MMLE of the covariance matrix and further establish the convergence rate and the sign consistency of the proposed PLNet estimator of the precision matrix in the high dimensional setting. Importantly, although the PLN model is not sub-Gaussian, we show that the PLNet estimator is consistent even if the model dimension goes to infinity exponentially as the sample size increases. The performance of PLNet is evaluated and compared with available methods using simulation and gene regulatory network analysis of real scRNA-seq data.

1 Introduction

Gaussian graphical model (GGM) [20, 35] is widely used for understanding the complex interactions of the observed variables in various fields [33, 19, 30]. GGM assumes that each sample is drawn from a multivariate Gaussian distribution, in which the precision matrix (i.e., the inverse of the covariance matrix) represents the network. In GGM, two nodes are connected if the corresponding element in the precision matrix is non-zero. Based on the sparsity assumption of the precision matrix, researchers have proposed many methods for estimating the precision matrix of the GGM and established consistency theories, such as methods by maximizing the penalized log-likelihood [35, 8], by solving an equivalent regression problem with the lasso penalty [20, 24] or by minimizing a smooth convex loss function (called D-trace loss) with the lasso penalty [36]. Due to its high interpretability, well-established theoretical properties, and computational advantages, GGM has been adopted in many applications.

This paper is motivated by the gene regulatory network analysis based on the single-cell RNA sequencing (scRNA-seq) data. The recent breakthrough of scRNA-seq technologies has provided tremendous opportunities for understanding transcriptional states and activities and for uncovering gene regulatory networks at single cell level. However, gene expressions from scRNA-seq are count data and the Gaussian assumption is inappropriate. In particular, for the recent unique molecular identifier (UMI) based scRNA-seq data [7, 38], the expression counts contain many zeros. Trans-

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

formations such as taking logarithms cannot make the Gaussian model a good approximation to the distribution of scRNA-seq expression data and may distort the correlation structure when there are many 0's in scRNA-seq data.

Poisson distribution is a natural choice for modeling count data. Researchers have generalized the univariate Poisson distribution to multivariate distributions for network analysis [12]. Some such models are the Poisson graphical model [2] and its variants generalized Poisson graphical models [32]. But these generalized Poisson graphical models can not handle over-dispersion often observed in real data. Negative binomial distributions are often used to account for the over-dispersion [25], but it is more difficult to generalize negative bionomial distributions to describe the network structure in the multivarate count data. Here, we use the multivariate Poisson log-normal (PLN) distribution for graphical modeling of count data. The PLN distribution is a hierarchical model of Poisson and multivariate log-normal distributions [1]. A random vector $\mathbf{Y} = (Y_1, \ldots, Y_p)^T$ is from a PLN distribution, if conditional on a latent vector $\mathbf{X} = (X_1, \ldots, X_p)^T$ with $(\log(X_1), \ldots, \log(X_p))^T \sim \mathrm{N}(\boldsymbol{\mu}, \Sigma)$, elements of Y are independent Poisson random variables with mean parameters $X_1, \ldots, X_p$.

The major advantage of the PLN model is that, similar to the GGM, the precision matrix $\Theta = \Sigma^{-1}$ can represent the network. Network recovery can be achieved by estimating Θ. In addition, PLN distributions allow over-dispersion and hence are suitable for modeling count data with over-dispersion. In gene regulatory network analysis of scRNA-seq data, the PLN graphical model has a clear biological explanation. Previous experimental researches showed that gene expressions of single cells follow log-normal distributions [4]. The latent variables $X_1, \cdots, X_p$, representing the true expressions of p genes in a single cell, can be reasonably assumed to be jointly log-normally distributed. The precision matrix of the log-normal distribution represents gene-gene interactions in single cells. The observed variables $\mathbf{Y}$ are measurements of the underlying true expressions $\mathbf{X}$. Largely speaking, the log-normal layer of the PLN model captures the biological fluctuation of gene expressions and the Poisson layer accounts for the technical and measurement noises. Only the biological fluctuation reflects gene-gene interactions and the regulatory network is the precision matrix of the latent log-normal model.

A few algorithms have been developed for estimating the precision matrix of the PLN model in high-dimensional settings. Compared with GGM, the likelihood of the PLN model is more complicated since it involves a high-dimensional integration and does not have a close form. Maximizing the penalized log-likelihood of the PLN model is more difficult. Wu et al. [31] proposed to use Monte Carlo to approximate the log-likelihood and estimate the precision matrix by maximizing the penalized approximated log-likelihood of the PLN model. Chiquet et al. [5] developed a computational more appealing method based on the variational approximation. However, these methods are all based on the approximation of the log-likelihood and the accuracies of these approximations need to be further elaborated. More importantly, no convergence theory has been developed for these precision matrix estimators.

In this article, we propose a two-step approach named PLNet that first estimates the covariance matrix of the latent log-normal variables in the PLN using the maximum marginal likelihood estimator (MMLE) and then estimates the sparse precision matrix by minimizing the lasso penalized D-trace loss [36]. One advantage of this two-step approach is that it avoids computing high-dimensional numerical integration that is involved in the log-likelihood of the multivariate PLN model. Minimizing the penalized D-trace loss is computationally cost-effective. Thus, this estimator is generally computationally more efficient. Furthermore, we show that, under mild conditions, the estimator given by PLNet is a consistent estimator of the precision matrix in high dimensional settings. Comprehensive simulation studies show that PLNet provides more accurate estimates and is computationally more efficient than available methods. We also demonstrate the application of PLNet to scRNA-seq data.

2 Methods

2.1 Notation

For a column vector $\mathbf{a}$ with the i-th entry a_i, $\|\mathbf{a}\|_2 = (\mathbf{a}^T\mathbf{a})^{1/2}$ and $\|\mathbf{a}\|_1 = \sum_i |a_i|$. For a matrix A with the (i, j)-th entry A_{ij}, $\|A\|_\infty = \max_{i,j} |A_{ij}|$ be the element-wise l_∞-norm, $\|A\|_1 = \sum_{i,j} |A_{ij}|$ be the l_1-norm, $\|A\|_{1,\infty} = \max_i(\sum_j |A_{ij}|)$ be the $l_{1,\infty}$-norm, $\|A\|_{\mathrm{F}} = \left(\sum_{i,j} |A_{ij}|^2\right)^{1/2}$ be the

Frobenius norm, $\|A\|_2 = \max_{\|v\|_2=1}\|Av\|_2$ be the operator norm, $\|A\|_{1,\text{off}} = \sum_{i\neq j}|A_{ij}|$, $\|A\|_0 = \sum_{i,j} 1_{\{A_{ij}\neq 0\}}$ be the number of nonzero entries where $1_{\{\cdot\}}$ is the indicator function, $\text{tr}\,(A) = \sum_i A_{ii}$ be the trace, and $A \succeq 0$ means A is semi-positive definite.

2.2 The PLN graphical model

Let $Y = [Y_{ij}]_{1\leq i\leq n, 1\leq j\leq p}$ be the observed count matrix, $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{ip})^T$ be the observed count data of the ith sample, and $\mathbf{X}_i = (X_{i1}, \ldots, X_{ip})^T$ be the latent random vector. In scRNA-seq data, Y_{ij} and X_{ij} are the observed expression and the underlying "true" expression of the jth gene in the ith cell, respectively. We assume that conditional on $\mathbf{X}_i$, elements of $\mathbf{Y}_i$ are independent Poisson random variables with mean $S_i X_{ij}$ ($1 \leq j \leq p$), where S_i is a scaling factor sampled from one distribution $g(S)$ and we define $\mathbf{S} = (S_1, ..., S_n)^T$ as the scaling vector. In scRNA-seq data, S_i corresponds to the library size of the ith cell. The library size is related to the total sequencing reads and can be estimated by the sum of counts within each cell or other methods [17, 18, 28]. The $\log(\mathbf{X}_i) = (\log(X_{i1}), \ldots, \log(X_{ip}))^T$ follows a multivariate normal random vector with mean $\boldsymbol{\mu}^*$ and covariance Σ^*. The precision matrix $\Theta^* = (\Sigma^*)^{-1}$ represents the network. In summary, we have the following graphical model for count data, for $1 \leq i \leq n$,

$$\begin{aligned} \mathbf{Y}_i|\mathbf{X}_i &\sim \prod_{j=1}^{p} \text{Poisson}(S_i X_{ij}), \\ \log(\mathbf{X}_i) &\sim \text{N}\left(\boldsymbol{\mu}^*, (\Theta^*)^{-1}\right). \end{aligned} \tag{1}$$

We denote this PLN ditribution by $\mathbf{Y}_i \sim \text{PLN}(S_i; \boldsymbol{\mu}^*, \Sigma^*)$. The above PLN model is a little different from the classical form [1], in which $S_i = 1$ for $1 \leq i \leq n$. We assume that the network Θ^* is sparse and therefore could use the penalized log-likelihood to estimate Θ^*. However, the likelihood function in the PLN model involves a p-dimensional integration and is difficult to compute, especially when p is large. Chiquet et al. developed a variational algorithm called VPLN to maximize the penalized log-likelihood [5]. Although the variational method is computationally more feasible than directly maximizing the penalized log-likelihood, the estimator's theoretical properties are difficult to be obtained.

We develop a two-step estimator called PLNet that is computationally efficient and has good theoretical properties. We first use the MMLE method to derive an estimator $\widehat{\Sigma}$ for Σ^*. Then, we apply the D-trace method [36] with the covariance estimator $\widehat{\Sigma}$ to estimate the sparse precision matrix Θ^*. We show that the derived estimator $\widehat{\Theta}$ is a consistent estimator of Θ^* even when the model dimension goes to infinity exponentially as the sample size increases.

2.3 The maximum marginal likelihood estimator

We estimate $\Sigma^* = [\sigma^*_{jk}]_{1\leq j,k\leq p}$ using the marginal log-likelihood to avoid high dimensional integrations. Let $\mathbf{Y}_{\cdot j}$ be the jth column of Y. For $1 \leq j \leq p$, we estimate σ^*_{jj} by maximizing the marginal log-likelihood of $Y_{\cdot j}$ and denote the estimator as $\tilde{\sigma}_{jj}$. For $1 \leq j \neq k \leq p$, we estimate σ^*_{jk} by maximizing the marginal log-likelihood of $(Y_{\cdot j}, Y_{\cdot k})$ and denote the estimator as $\tilde{\sigma}_{jk}$. The specific forms of these marginal log-likelihoods and optimization problems are shown in the Supplemental Material (S2.1). Further, we define the MMLE for Σ^* as $\widetilde{\Sigma} = [\tilde{\sigma}_{jk}]_{1\leq j,k\leq p}$. To solve the optimization problem for maximizing the marginal log-likelihood, we apply the Newton-Raphson algorithm with the moment estimator $\widetilde{\Sigma}^m$ in the Supplemental Material (S1.1) as the initial value. The explicit forms of the first and second order partial derivatives of the marginal log-likelihood function are shown in the Supplemental Material (S2.1). The integrations involved in optimizations can be well approximated by the method described in [1, 22].

The above MMLE $\widetilde{\Sigma}$ may be not semi-positive definite. However, the D-trace method requires the input covariance matrix estimator to be semi-positive definite to guarantee the convexity of the loss function. We project $\widetilde{\Sigma}$ to the space of semi-positive definite matrices and identify $\check{\Sigma}$ that is closest to $\widetilde{\Sigma}$ in the space, i.e.,

$$\check{\Sigma} = \arg\min_{A\succeq 0} \left\| A - \widetilde{\Sigma} \right\|_\infty . \tag{2}$$

Then we add a small positive definite matrix to $\check{\Sigma}$ and get a positive-definite estimator $\widehat{\Sigma}$ of Σ^*,

$$\widehat{\Sigma} = \check{\Sigma} + \left\| \check{\Sigma} - \widetilde{\Sigma} \right\|_{\infty} I, \tag{3}$$

where I is the identity matrix. The optimization problem (2) for $\check{\Sigma}$ can be solved by a splitting conic solver [9].

With the covariance matrix estimator $\widehat{\Sigma}$, we apply the D-trace method to estimate the precision matrix,

$$\widehat{\Theta} = \arg\min_{\Theta \succeq 0} \frac{1}{2} \mathrm{tr}\left(\widehat{\Sigma}\Theta^2\right) - \mathrm{tr}\left(\Theta\right) + \lambda_n \left\|\Theta\right\|_{1,\text{off}}, \tag{4}$$

The above optimization problem (4) can be solved by an alternating direction method of multipliers [36]. In this paper, we use a more efficient algorithm developed in [29] to calculate $\widehat{\Theta}$. The tuning parameter λ_n is selected by minimizing the following approximate Bayesian information criterion (BIC) [37],

$$\left\| \frac{1}{2}(\widehat{\Theta}\widehat{\Sigma} + \widehat{\Sigma}\widehat{\Theta}) - I \right\|_{\mathrm{F}} + \left(\|\widehat{\Theta}\|_0 \log n \right) / n. \tag{5}$$

3 Theoretical properties

In this section, we establish the theoretical properties in the high dimensional setting. We first prove that $\widehat{\Sigma}$ is a consistent estimator of Σ^*. The convergence rate for $\widehat{\Sigma}$ is similar to that of the sample covariance matrix for random variables with sub-Gaussian distribution. Then, under the same irrepresentability condition of the D-trace method in [36], we derive the edge recovery property and consistency for the PLNet estimator $\widehat{\Theta}$.

3.1 Notation

Let $G = \{(i,j) | \Theta^*_{ij} \neq 0\}$ be positions of non-zero elements in Θ^*, G^c be the complement set of G, d be the maximum node degree in Θ^*, $s = \|\Theta^*\|_0$, and $\theta_{\text{min}} = \min_{(i,j)\in G} \left|\Theta^*_{ij}\right|$ be the minimal absolute value of nonzero elements of Θ^*. Let $\lambda_{\text{max}}(A)$ and $\lambda_{\text{min}}(A)$ be the largest and smallest eigenvalues of a symmetric matrix A. We define $\Gamma^* = \Gamma(\Sigma^*) = (\Sigma^* \otimes I + I \otimes \Sigma^*)/2$, $\widehat{\Gamma} = \Gamma\left(\widehat{\Sigma}\right)$, where $A \otimes B$ is the Kronecker product. For a $p \times p$ matrix A, the row $(i-1)p + j$ and column $(k-1)p + l$ of $\Gamma(A)$ is $\Gamma(A)_{(i,j),(k,l)} = \left(A_{ik}1_{\{j=l\}} + A_{jl}1_{\{i=k\}}\right)/2$. For two subsets T_1 and T_2 of $\{(i,j) \,|1 \leq i,j \leq p\}$, we define $\Gamma(A)_{T_1,T_2}$ be the submatrix of $\Gamma(A)$ whose rows and columns indexed by T_1 and T_2, respectively. Other notations are as follows,

$$\gamma = 1 - \max_{(i,j)\in G^c} \left\| \Gamma^*_{(i,j),G} (\Gamma^*_{G,G})^{-1} \right\|_1,$$

$$k_\Gamma = \left\| (\Gamma^*_{G,G})^{-1} \right\|_{1,\infty}, k_\Sigma = \|\Sigma^*\|_{1,\infty}.$$

3.2 The irrepresentability condition and rate of convergence

We first present the necessary irrepresentability condition for establishing the rate of convergence for the estimator in PLNet. This irrepresentability condition is from the D-trace method in [36].

Condition 1 (Irrepresentability condition) *There exists* $0 < \alpha < 1$*, such that*

$$\max_{(i,j)\in G^c} \left\| \Gamma^*_{(i,j),G} (\Gamma^*_{G,G})^{-1} \right\|_1 \leq 1 - \alpha.$$

The irrepresentability condition 1 is equivalent to $\gamma \geq \alpha > 0$. We also need a boundedness condition in the PLN model (1).

Condition 2 (Boundedness condition) *There exist positive constants* M_1, M_2, M_3, M_4*, and* M_5*, such that the distribution of* S_i *has bounded support* $[M_1, M_2]$*,* $\max_{1\leq j,k\leq p} \left\{ \left|\mu^*_j\right|, \left|\sigma^*_{jk}\right| \right\} \leq M_3$ *and* $M_4 \leq \lambda_{min}(\Sigma^*) \leq \lambda_{max}(\Sigma^*) \leq M_5$*.*

Based on the boundedness condition 2, we can establish the convergence rate for the covariance matrix estimator $\widehat{\Sigma}$.

Theorem 1 (Rate of convergence for $\widehat{\Sigma}$) *Under Condition 2, there exist positive constants A, B, such that for any $\epsilon > 0$,*

$$pr\left(\left\|\widehat{\Sigma} - \Sigma^*\right\|_\infty \geq \epsilon\right) \leq p^2 A \exp(-Bn\epsilon^2).$$

After plugging $\widehat{\Sigma}$ into the lasso penalized D-trace loss, we get a consistent estimator $\widehat{\Theta}$ that converges to Θ^* in several matrix norms.

Theorem 2 (Rate of convergence for $\widehat{\Theta}$) *Under the irrepresentability condition 1 and the boundedness condition 2, there exist constants A, B, such that for some $\eta > 2$, if*

$$n > \quad B^{-1}(\eta \log p + \log A)\max\Bigg[12dk_\Gamma,\ 12\gamma^{-1}(k_\Sigma k_\Gamma^2 + k_\Gamma),\ \left\{12\gamma^{-1}\left(k_\Sigma k_\Gamma^3 + k_\Gamma^2\right) + 5dk_\Gamma^2\right\}\theta_{\min}^{-1},$$
$$\min\left\{s^{1/2}, d+1\right\}\left\{12\gamma^{-1}\left(k_\Sigma k_\Gamma^3 + k_\Gamma^2\right) + 5dk_\Gamma^2\right\}\lambda_{\min}^{-1}(\Theta^*)\Bigg]^2,$$

and

$$\lambda_n = 12\gamma^{-1}\left(k_\Sigma k_\Gamma^2 + k_\Gamma\right) B^{-1/2}(\eta \log p + \log A)^{1/2} n^{-1/2},$$

then with probability $1 - p^{2-\eta}$,

$$\left\|\widehat{\Theta} - \Theta^*\right\|_\infty \leq \left(12\gamma^{-1}\left(k_\Sigma k_\Gamma^3 + k_\Gamma^2\right) + 5dk_\Gamma^2\right) B^{-1/2}(\eta \log p + \log A)^{1/2} n^{-1/2},$$
$$\left\|\widehat{\Theta} - \Theta^*\right\|_F \leq s^{1/2}\left(12\gamma^{-1}\left(k_\Sigma k_\Gamma^3 + k_\Gamma^2\right) + 5dk_\Gamma^2\right) B^{-1/2}(\eta \log p + \log A)^{1/2} n^{-1/2},$$
$$\left\|\widehat{\Theta} - \Theta^*\right\|_2 \leq \min\left\{s^{1/2}, d+1\right\}\left(12\gamma^{-1}\left(k_\Sigma k_\Gamma^3 + k_\Gamma^2\right) + 5dk_\Gamma^2\right) B^{-1/2}(\eta \log p + \log A)^{1/2} n^{-1/2}.$$

Meanwhile, with a high probability, the sign of the sparse precision matrix Θ^* can be recovered by $\widehat{\Theta}$. We have the following theorem about the sign consistency of $\widehat{\Theta}$.

Theorem 3 (Sign consistency for $\widehat{\Theta}$) *Under the conditions in Theorem 2, for some $\eta > 2$, choosing the same n and λ_n in Theorem 2, then with probability $1 - p^{2-\eta}$, $\widehat{\Theta}$ recovers all zeros and nonzeros in Θ^*.*

The rate-of-convergence and sign consistency results in Theorem 2 and 3 are closely related to the tail probability of the MMLE estimator $\widehat{\Sigma}$ in Theorem 1. The boundedness condition 2 is assumed to guarantee the convergency of the MMLE. Ignoring the complicated constants in theorems, for any $\eta > 2$, if we have $n > C\eta \log p$, or in other words, if p tends to infinity not faster than exponential of the sample size n, $\widehat{\Theta}$ is a consistent estimator of Θ^*. Especially, the rate of convergence for $\widehat{\Theta}$ is $O\left([\eta(\log p)/n]^{1/2}\right)$ under l_∞-norm.

4 Simulation studies

4.1 Simulation settings

We conduct simulations to evaluate the performance of PLNet and compare with the available network inference methods including VPLN [5], glasso [8], and an estimator called PLNet-MOM by plugging the moment estimator $\widehat{\Sigma}^m$ into D-trace loss (See Supplemental Material (S1.1)). Both PLNet and VPLN are designed to estimate the precision matrix for count data in the PLN model. The glasso algorithm is a classical approach for continuous data in GGM and we apply glasso to the logarithmic transformation of the normalized data, which is defined as $\tilde{Y}_{ij} = (Y_{ij} + 1)/\hat{S}_i,\ 1 \leq i \leq n$, where $\hat{S}_i$ is the estimated library size of ith cell. In all simulations, we estimate the library size by the total

sum scaling (i.e. $\hat{S}_i = \sum_{j=1}^{p} Y_{ij}$), which is a classical normalization method for scRNA-seq and is defined as the sum of counts within each cell.

We consider 48 different simulation scenarios, which are 2 sample size setups ($n = 500, 2000$) $\times$ 3 dimension setups ($p = 100, 300, 500$) $\times$ 2 dropout levels (low and high) $\times$ 4 graph structures and simulate count data from the PLN model, where the dropout level represents the proportion of zeros in count matrix. Details of the simulation settings are shown in the Supplementary Material (S3.1). In each scenario, we independently repeat simulations 100 times. The four graph structures are as follows.

1. Banded Graph: Pairs (i, j) of nodes are connected if $|i - j| \leq 2,\ i \neq j$. All nonzero edges are set as 0.3.
2. Random Graph: Pairs of nodes are connected with probability 0.1. The nonzero edges are set as 0.3 with probability 0.8 and as -0.3 with probability 0.2.
3. Scale-free Graph: The Barabasi-Albert model [3] is used to generate a scale-free graph with power 1. The nonzero edges are set as 0.3.
4. Blocked Graph: The nodes are divided into 5 blocks of equal sizes. Pairs of nodes in the same block are connected with probability 0.1 and the nonzero edges are set as 0.3. Different blocks have no edge connection.

The diagonal elements of precision matrices are set as 1 plus a positive number to guarantee positive definiteness.

4.2 Performance comparison

Table 1 and Table 2 show the area under precision and recall curve (AUPR) of each estimator. AUPRs are calculated by varying the tuning parameters (i.e. the penalty parameters of the four estimators). As expected, the AUPR decreases as the number of genes increases. AUPRs in the high-dropout cases are generally smaller than that in the low-dropout cases. PLNet is the most robust estimation among these four estimators and outperforms PLNet-MOM, VPLN, and glasso in most simulation settings in AUPR, especially for the settings with high dropouts. For example, for $n = 2000, p = 100$, PLNet achieves an AUPR of 0.88 for the random graph under the scenario of the high dropout rate, while PLNet-MOM, VPLN and glasso only have AUPRs of 0.82, 0.69, and 0.18, respectively. Furthermore, PLNet-MOM outperforms glasso in most settings and outperforms VPLN under the scenario of large sample size. The results of the area under the receiver operating characteristic curve (AUC) are similar and shown in Supplementary Table S1-S2.

To further demonstrate the performance of PLNet, we visualize the mean networks predicted by the four methods tunned by BIC criterion for the banded graph with $n = 2000, p = 100$ over the 100 simulations (Figure 1). More specifically, we calculate the relative frequency F_{ij} that an algorithm reports edges over the 100 simulations. For positions (i, j) with $\Theta_{ij} \neq 0$, F_{ij} is the proportion that an algorithm correctly recovers the edge in the 100 simulations, while for positions (i, j) with $\Theta_{ij} = 0$, F_{ij} is the the proportion that an algorithm falsely predicts edges between nodes i and j in the 100 simulations. We plot the relative frequency matrices of the four methods in Figure 1. We clearly see that PLNet is able to detect more true positives while having fewer false positives than other methods, especially for the high dropout case.

Supplementary Table S3-S4 show the computational time of the four methods. PLNet is computationally more efficient than VPLN in most simulation settings. Also, the computational efficiency of PLNet is roughly the same for different sample size, while computational complexity of VPLN maintains a linear relationship with sample size. The detailed comparsions between computational efficiency of the four methods are shown in the Supplementary Material (S3.2.2).

5 Application to a scRNA-seq dataset

We apply PLNet and VPLN to infer the gene regulatory network of CD14+ Monocytes profiled in Kang et al. [14]. The single cells are profiled in two different conditions, IFN-β-treated and control. IFN-β is a cytokine in the interferon family that influences the transcriptional profiles for many genes, especially that in the JAK/STAT pathway [21]. We focus on the IFN-β-treated cells (2147 cells) and use the top 200 highly variable genes that are used in [27] for network analysis.

Table 1: Comparisons of PLNet, VPLN, glasso, and PLNet-MOM in terms of the area under precision and recall curve (AUPR) on simulation results for $n = 500$. The results are averages over 100 replicates with standard deviations in brackets.

Sample size	$n = 500$		$n = 500$		$n = 500$	
Dimension	$p = 100$		$p = 300$		$p = 500$	
Dropout	Low	High	Low	High	Low	High
			Banded graph			
PLNet	**0.79 (0.03)**	**0.51 (0.04)**	**0.6 (0.06)**	**0.31 (0.04)**	0.44 (0.08)	**0.22 (0.03)**
PLNet-MOM	0.62 (0.03)	0.43 (0.03)	0.33 (0.02)	0.21 (0.01)	0.21 (0.02)	0.13 (0.01)
VPLN	0.68 (0.06)	0.44 (0.04)	0.6 (0.02)	0.27 (0.02)	**0.54 (0.01)**	0.21 (0.02)
glasso	0.34 (0.04)	0.05 (0.01)	0.44 (0.03)	0.04 (0.01)	0.44 (0.02)	0.04 (0.01)
			Random graph			
PLNet	**0.73 (0.07)**	**0.49 (0.07)**	**0.6 (0.07)**	0.25 (0.05)	**0.53 (0.09)**	0.16 (0.04)
PLNet-MOM	0.6 (0.07)	0.44 (0.07)	0.44 (0.04)	0.22 (0.04)	0.36 (0.04)	0.14 (0.03)
VPLN	0.59 (0.07)	0.44 (0.08)	0.53 (0.09)	**0.26 (0.05)**	0.51 (0.06)	**0.18 (0.05)**
glasso	0.41 (0.05)	0.14 (0.02)	0.48 (0.05)	0.11 (0.02)	0.48 (0.05)	0.09 (0.02)
			Scale-free Graph			
PLNet	**0.73 (0.12)**	**0.55 (0.08)**	**0.67 (0.04)**	**0.44 (0.05)**	**0.59 (0.04)**	0.35 (0.06)
PLNet-MOM	0.63 (0.08)	0.51 (0.06)	0.48 (0.03)	0.34 (0.03)	0.39 (0.01)	0.27 (0.03)
VPLN	0.67 (0.05)	0.53 (0.09)	0.61 (0.03)	0.43 (0.04)	0.57 (0.07)	**0.38 (0.05)**
glasso	0.56 (0.06)	0.34 (0.05)	0.59 (0.02)	0.32 (0.03)	0.58 (0.02)	0.3 (0.04)
			Blocked graph			
PLNet	**0.68 (0.05)**	**0.47 (0.08)**	**0.59 (0.08)**	**0.24 (0.07)**	**0.45 (0.1)**	**0.17 (0.04)**
PLNet-MOM	0.58 (0.04)	0.43 (0.08)	0.43 (0.06)	0.2 (0.05)	0.32 (0.05)	0.15 (0.03)
VPLN	0.59 (0.04)	0.45 (0.07)	0.52 (0.07)	0.24 (0.06)	0.44 (0.08)	0.17 (0.04)
glasso	0.37 (0.03)	0.16 (0.03)	0.44 (0.06)	0.11 (0.03)	0.4 (0.06)	0.09 (0.02)

Table 2: Comparisons of PLNet, VPLN, glasso, and PLNet-MOM in terms of the area under precision and recall curve (AUPR) on simulation results for $n = 2000$. The results are averages over 100 replicates with standard deviations in brackets.

Sample size	$n = 2000$		$n = 2000$		$n = 2000$	
Dimension	$p = 100$		$p = 300$		$p = 500$	
Dropout	Low	High	Low	High	Low	High
			Banded graph			
PLNet	**0.99 (0.01)**	**0.96 (0.01)**	**0.99 (0.01)**	**0.94 (0.02)**	**0.98 (0.01)**	**0.89 (0.08)**
PLNet-MOM	0.97 (0.01)	0.92 (0.01)	0.91 (0.01)	0.83 (0.02)	0.83 (0.02)	0.75 (0.01)
VPLN	0.95 (0.01)	0.89 (0.03)	0.94 (0.01)	0.79 (0.15)	0.94 (0.01)	0.81 (0.01)
glasso	0.62 (0.03)	0.04 (0.01)	0.82 (0.01)	0.07 (0.01)	0.85 (0.01)	0.15 (0.02)
			Random graph			
PLNet	**0.98 (0.01)**	**0.88 (0.04)**	**0.98 (0.03)**	**0.85 (0.05)**	**0.99 (0.01)**	**0.83 (0.04)**
PLNet-MOM	0.94 (0.02)	0.82 (0.06)	0.94 (0.01)	0.77 (0.05)	0.93 (0.01)	0.74 (0.05)
VPLN	0.78 (0.08)	0.69 (0.07)	0.88 (0.03)	0.67 (0.1)	0.86 (0.11)	0.67 (0.11)
glasso	0.55 (0.06)	0.18 (0.03)	0.8 (0.03)	0.24 (0.04)	0.84 (0.02)	0.26 (0.04)
			Scale-free Graph			
PLNet	**0.89 (0.17)**	**0.85 (0.11)**	**0.97 (0.02)**	**0.85 (0.03)**	**0.96 (0.03)**	**0.83 (0.02)**
PLNet-MOM	0.85 (0.11)	0.81 (0.08)	0.86 (0.01)	0.75 (0.02)	0.83 (0.01)	0.71 (0.01)
VPLN	0.74 (0.16)	0.67 (0.15)	0.79 (0.04)	0.68 (0.11)	0.8 (0.05)	0.66 (0.13)
glasso	0.59 (0.14)	0.45 (0.06)	0.78 (0.02)	0.5 (0.03)	0.81 (0.02)	0.53 (0.02)
			Blocked graph			
PLNet	**0.94 (0.02)**	**0.83 (0.07)**	**0.97 (0.01)**	**0.81 (0.08)**	**0.97 (0.01)**	**0.77 (0.05)**
PLNet-MOM	0.88 (0.04)	0.75 (0.08)	0.91 (0.02)	0.72 (0.08)	0.89 (0.02)	0.68 (0.05)
VPLN	0.73 (0.03)	0.66 (0.07)	0.78 (0.04)	0.62 (0.07)	0.8 (0.06)	0.59 (0.11)
glasso	0.47 (0.05)	0.2 (0.03)	0.7 (0.04)	0.21 (0.04)	0.75 (0.03)	0.21 (0.04)

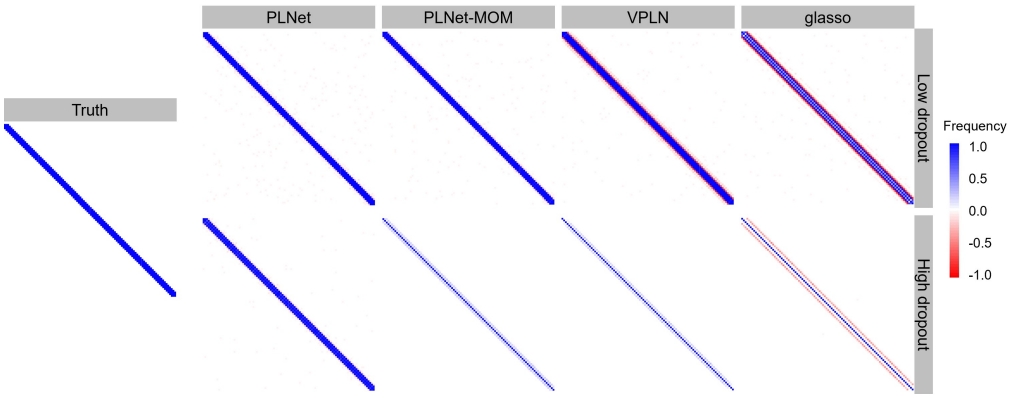
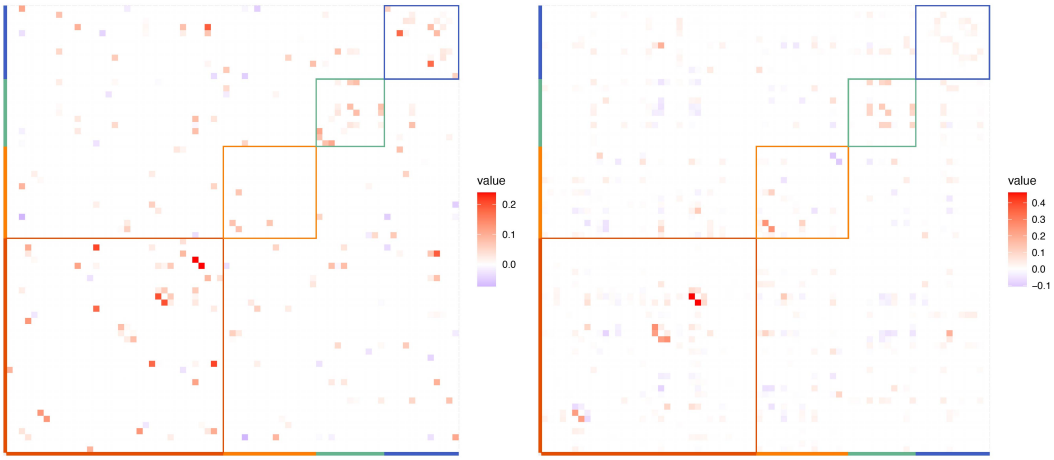


Figure 1: The mean networks predicted by PLNet, VPLN, glasso and PLNet-MOM for the banded graph with 100 nodes and n = 2000. False edges are colored in red and true edges are in blue. The left panel is the true network matrix for reference.



(a) PLNet (b) VPLN

Figure 2: Heat maps of partial correlations between genes in the 4 GO modules given by PLNet (a) and VPLN (b). Red: Cytokine-mediated signaling pathway (Module M_1); Orange: Neutrophil mediated immunity (Module M_2); Green: Cellular protein metabolic process (Module M_3); Blue: Proteolysis (Module M_4)

We compare networks of PLNet and VPLN with parameters tuned such that the network densities are around 5%. Gene Ontology (GO) analysis [15] shows that the 200 genes mainly involve in 4 biological processes, including "Cytokine-mediated signaling pathway" (Module M_1), "Neutrophil mediated immunity" (Module M_2), "Cellular protein metabolic process" (Module M_3), and "Proteolysis" (Module M_4). Figure 2 shows the predicted networks of the genes in the 4 modules by PLNet and VPLN, where the colors represent the partial correlations between genes. The partial correlation given by PLNet between genes i and j is defined as $-\widehat{\Theta}_{ij}/(\widehat{\Theta}_{ii}\widehat{\Theta}_{ij})^{1/2}$. The partial correlation given by VPLN is defined similarly. We clearly see that the network given by PLNet tends to have more connections within the modules than VPLN. To see this more clearly, for each module M_k, we calculate the ratio between within-module and between-module connections $R(M_k) = \Sigma_{i,j\in M_k} W_{ij}/\Sigma_{i\in M_k,\, j\notin M_k} W_{ij}$, where the weights W_{ij} are set as the absolute partial correlation between genes i and j (weighted within-between connection ratio) or are set as 1 and 0 depending on whether genes i and j are connected (unweighted within-between connection ratio). The within-between connection ratios of PLNet are much larger than that of VPLN in most cases (Table 3). Similar results also hold for networks of other different densities that are shown in the Supplementary Material (S4.1).

Table 3: The within-between connection ratios of the 4 modules in the networks estimated by PLNet and VPLN tuned such that the network densities are around 5%.

Type	Method	Module 1	Module 2	Module 3	Module 4
Weighted	PLNet	**0.681**	0.231	**0.582**	**0.387**
	VPLN	0.621	**0.381**	0.459	0.259
Unweighted	PLNet	**0.533**	0.129	**0.419**	**0.383**
	VPLN	0.515	**0.200**	0.170	0.213

6 Discussion

In this paper, we consider the PLN graphical model for count data. This model has an intuitive explanation for single-cell gene regulatory network analysis. The Poisson layer is to capture the technical noises and the log-normal layer is to model the biological fluctuations in single cells. Gene regulatory network is represented by the precision matrix of the latent log-normal model. To estimate the precision matrix, we propose a two-step estimator PLNet, using the MMLE to estimate the covariance matrix and then minimizing the penalized D-trace loss to estimate the precision matrix. The simplicity of this estimation procedure allows us to establish the consistency theory for the proposed PLNet estimator for the high dimensional setting. The numerical analysis also shows that the PLNet method outperforms available methods.

The proposed method can be generalized in several ways. Firstly, a straightforward generalization is the differential network analysis. We could use the same method to estimate covariance matrices in two different conditions and use the D-trace loss developed in earlier work [34] for differential network analysis in single cells. Secondly, the mean parameters of the latent log-normal random variable can depend on covariates. In such model, we could first use regression to estimate the mean parameters and derive the corresponding moment estimator of the covariance matrix. Another generalization is gene regulatory network analysis of mixtures of cell populations. Different cell populations may have different gene regulatory networks and we could jointly model the mixture and infer the gene regulatory networks for all cell populations.

References

[1] J. Aitchison and C. H. Ho. The multivariate Poisson-log normal distribution. *Biometrika*, 76(4): 643–653, 1989. ISSN 0006-3444. doi: 10.1093/biomet/76.4.643. URL `https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/76.4.643`.

[2] Genevera I. Allen and Zhandong Liu. A local Poisson graphical model for inferring networks from sequencing data. *IEEE Transactions on NanoBioscience*, 12(3):189–198, sep 2013. ISSN 1536-1241. doi: 10.1109/TNB.2013.2263838. URL `http://ieeexplore.ieee.org/document/6578145/`.

[3] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, oct 1999. ISSN 0036-8075. doi: 10.1126/science.286.5439.509. URL `https://www.sciencemag.org/lookup/doi/10.1126/science.286.5439.509`.

[4] Martin Bengtsson, Anders Ståhlberg, Patrik Rorsman, and Mikael Kubista. Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. *Genome Research*, 15(10):1388–1392, 2005.

[5] Julien Chiquet, Stephane Robin, and Mahendra Mariadassou. Variational inference for sparse network reconstruction from count data. In *International Conference on Machine Learning*, pages 1162–1171. PMLR, 2019.

[6] ENCODE Project Consortium et al. The ENCODE (ENCyclopedia of DNA elements) project. *Science*, 306(5696):636–640, 2004.

[7] Jeffrey A. Farrell, Yiqun Wang, Samantha J. Riesenfeld, Karthik Shekhar, Aviv Regev, and Alexander F. Schier. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science*, 360(6392):967–968, 2018. ISSN 0036-8075. doi: 10.1126/science.aar3131. URL `https://science.sciencemag.org/content/360/6392/eaar3131`.

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[9] Anqi Fu, Balasubramanian Narasimhan, and Stephen Boyd. CVXR: An R package for disciplined convex optimization. *Journal of Statistical Software*, 94(14):1–34, 2020. ISSN 1548-7660. doi: 10.18637/jss.v094.i14. URL `https://www.jstatsoft.org/v094/i14`.

[10] Jan Hilpert, Johanna M. Beekman, Susanne Schwenke, Kristin Kowal, David Bauer, Johannes Lampe, Rupert Sandbrink, Jürgen F. Heubach, Steffen Stürzebecher, and Joachim Reischl. Biological response genes after single dose administration of interferon β-1b to healthy male volunteers. *Journal of Neuroimmunology*, 199(1-2):115–125, aug 2008. ISSN 01655728. doi: 10.1016/j.jneuroim.2008.04.036. URL `https://linkinghub.elsevier.com/retrieve/pii/S0165572808001252`.

[11] Felicita Hornung, Giuseppe Scala, and Michael J Lenardo. TNF-α-induced secretion of CC chemokines modulates CC chemokine receptor 5 expression on peripheral blood lymphocytes. *The Journal of Immunology*, 164(12):6180–6187, 2000.

[12] David I. Inouye, Eunho Yang, Genevera I. Allen, and Pradeep Ravikumar. A review of multivariate distributions for count data derived from the Poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3):e1398, may 2017. ISSN 19395108. doi: 10.1002/wics.1398. URL `http://arxiv.org/abs/1609.00066http://doi.wiley.com/10.1002/wics.1398`.

[13] J. Jablonska, C.-F. Wu, L. Andzinski, S. Leschner, and S. Weiss. CXCR2-mediated tumor-associated neutrophil recruitment is regulated by IFN-β. *International Journal of Cancer*, 134 (6):1346–1358, mar 2014. ISSN 00207136. doi: 10.1002/ijc.28551. URL `http://doi.wiley.com/10.1002/ijc.28551`.

[14] Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M. Lanata, Rachel E. Gate, Sara Mostafavi, Alexander Marson, Noah Zaitlen, Lindsey A. Criswell, and Chun Jimmie Ye. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology*, 36(1):89–94, jan 2018. ISSN 1087-0156. doi: 10.1038/nbt.4042. URL `http://www.nature.com/articles/nbt.4042`.

[15] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Research*, 44(W1):W90–W97, 2016.

[16] Alexander Lachmann, Huilei Xu, Jayanth Krishnan, Seth I Berger, Amin R Mazloom, and Avi Ma'ayan. ChEA: transcription factor regulation inferred from integrating genome-wide ChIP-X experiments. *Bioinformatics*, 26(19):2438–2444, 2010.

[17] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):1–21, 2014.

[18] Aaron TL Lun, Karsten Bach, and John C Marioni. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, 17(1):1–14, 2016.

[19] Shisong Ma, Qingqiu Gong, and Hans J Bohnert. An Arabidopsis gene network based on the graphical Gaussian model. *Genome Research*, 17(11):1614–1625, 2007.

[20] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, aug 2006. ISSN 0090-5364. doi: 10.1214/009053606000000281. URL `https://projecteuclid.org/journals/annals-of-statistics/volume-34/issue-3/High-dimensional-graphs-and-variable-selection-with-the-Lasso/10.1214/009053606000000281.fullhttp://arxiv.org/abs/math/0608017http://dx.doi.org/10.1214/009053606000000281`.

[21] Sara Mostafavi, Hideyuki Yoshida, Devapregasan Moodley, Hugo LeBoité, Katherine Rothamel, Towfique Raj, Chun Jimmie Ye, Nicolas Chevrier, Shen-Ying Zhang, Ting Feng, Mark Lee, Jean-Laurent Casanova, James D. Clark, Martin Hegen, Jean-Baptiste Telliez, Nir Hacohen, Philip L. De Jager, Aviv Regev, Diane Mathis, and Christophe Benoist. Parsing the interferon transcriptional network and its disease associations. *Cell*, 164(3):564–578, jan 2016. ISSN 00928674. doi: 10.1016/j.cell.2015.12.032. URL `https://linkinghub.elsevier.com/retrieve/pii/S0092867415016967`.

[22] J. C. Naylor and A. F. M. Smith. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31(3):214–225, 1982. ISSN 00359254, 14679876. URL `http://www.jstor.org/stable/2347995`.

[23] V Özenci, M Kouwenhoven, Y-M Huang, P Kivisäkk, and H Link. Multiple sclerosis is associated with an imbalance between tumour necrosis factor-alpha (TNF-α)-and IL-10-secreting blood cells that is corrected by interferon-beta (IFN-β) treatment. *Clinical & Experimental Immunology*, 120(1):147–153, 2000.

[24] 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.

[25] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, jan 2009. ISSN 14602059. doi: 10.1093/bioinformatics/btp616. URL `https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp616`.

[26] Andrew D Rouillard, Gregory W Gundersen, Nicolas F Fernandez, Zichen Wang, Caroline D Monteiro, Michael G McDermott, and Avi Ma'ayan. The harmonizome: a collection of processed datasets gathered to serve and mine knowledge about genes and proteins. *Database*, 2016, 2016.

[27] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, jun 2019. ISSN 00928674. doi: 10.1016/j.cell.2019.05.031. URL `https://linkinghub.elsevier.com/retrieve/pii/S0092867419305598`.

[28] Catalina A Vallejos, John C Marioni, and Sylvia Richardson. BASiCS: Bayesian analysis of single-cell sequencing data. *PLoS Computational Biology*, 11(6):e1004333, 2015.

[29] Cheng Wang and Binyan Jiang. An efficient ADMM algorithm for high dimensional precision matrix estimation via penalized quadratic loss. *Computational Statistics & Data Analysis*, 142:106812, feb 2020. ISSN 01679473. doi: 10.1016/j.csda.2019.106812. URL `https://linkinghub.elsevier.com/retrieve/pii/S0167947319301574`.

[30] Anja Wille, Philip Zimmermann, Eva Vranová, Andreas Fürholz, Oliver Laule, Stefan Bleuler, Lars Hennig, Amela Prelić, Peter von Rohr, Lothar Thiele, et al. Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome Biology*, 5(11):1–13, 2004.

[31] Hao Wu, Xinwei Deng, and Naren Ramakrishnan. Sparse estimation of multivariate Poisson log-normal models from count data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 11(2):66–77, apr 2018. ISSN 19321864. doi: 10.1002/sam.11370. URL `http://arxiv.org/abs/1602.07337http://doi.wiley.com/10.1002/sam.11370`.

[32] Eunho Yang, Pradeep Ravikumar, Genevera I Allen, and Zhandong Liu. On poisson graphical models. In *NIPS*, pages 1718–1726, 2013.

[33] Jianxin Yin and Hongzhe Li. A sparse conditional Gaussian graphical model for analysis of genetical genomics data. *The Annals of Applied Statistics*, 5(4):2630, 2011.

[34] Huili Yuan, Ruibin Xi, Chong Chen, and Minghua Deng. Differential network analysis via lasso penalized D-trace loss. *Biometrika*, 104(4):755–770, 2017.

[35] Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, feb 2007. ISSN 0006-3444. doi: 10.1093/biomet/asm018. URL https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/asm018.

[36] Teng Zhang and Hui Zou. Sparse precision matrix estimation via lasso penalized D-trace loss. *Biometrika*, 101(1):103–120, mar 2014. ISSN 0006-3444. doi: 10.1093/biomet/ast059. URL https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/ast059.

[37] Sihai Dave Zhao, T. Tony Cai, and Hongzhe Li. Direct estimation of differential networks. *Biometrika*, 101(2):253–268, jun 2014. ISSN 0006-3444. doi: 10.1093/biomet/asu009. URL https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/asu009.

[38] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):1–12, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work do not have any negative societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Conditions 1 and 2 and Theorems.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Supplemental Material for proofs.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Supplemental Material for codes.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1 for Simulation settings and equation 5 for hyperparameter chosen.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See the standard deviations in Table 1 and 2 for error bars.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplemental Material (S3.2.2).
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See lines 105, 114, 128, 135, 138, 177, 195, 230 for citations.
   (b) Did you mention the license of the assets? [Yes] All the assets are open available.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]