Multi-level Gaussian Graphical Models Conditional on Covariates

Gi Bum Kim
Computational Biology Department
Carnegie Mellon University
gibumk@cs.cmu.edu

Seyoung Kim
Computational Biology Department
Carnegie Mellon University
sssykim@cs.cmu.edu

Abstract

We address the problem of learning the structure of a high-dimensional Gaussian graphical model conditional on covariates, when each sample belongs to groups at multiple levels of hierarchy. The existing statistical methods for learning covariate-conditioned Gaussian graphical models focused on learning the aggregate behavior of inputs and outputs in a single-layer network. We propose a statistical model called multi-level conditional Gaussian graphical models for modeling multi-level output networks influenced by both individual-level and group-level inputs. We describe a decomposition of our model into a product of two components, one for sum variables and the other for difference variables derived from the original variables. This decomposition leads to an efficient learning algorithm for both complete data and incomplete data with randomly missing individual observations, as the expensive repeated computation of the partition function can be avoided. We demonstrate our method on simulated data and real-world data in finance and genomics.

1 INTRODUCTION

We consider the problem of modeling multiple correlated outputs influenced by multiple inputs in high-dimensional setting, when individuals belong to groups at multiple levels of hierarchy. The individuals' outcomes can be influenced both by individual-level inputs at the lower level of hierarchy and by group-level inputs at the upper level of hierarchy. For example, exam scores of multiple subjects for each student may be influenced by the attributes of the student but also by the attributes of the school. A similar problem arises when predicting voting patterns of voters in districts based on the attributes of voters and districts or predicting energy consumption of households in regions based on the attributes of households and regions. Multi-level regression has been widely used to learn regression coefficients for both individual- and group-level inputs (Snijders and Bosker, 2012). However, these methods have considered only multi-level regression coefficients but not multi-level output networks.

Here, we address the problem of modeling multi-level output network structure in addition to multi-level input-to-output mapping for grouped data. We model the overall network over multiple correlated outputs as an aggregate of multiple networks at different granularity. The network component at the individual level governs the output behavior of individuals within each group, whereas the network component at the group level governs the output behavior across groups.

We introduce sparse multi-level conditional Gaussian graphical models for modeling multi-level sparse output network influenced by multi-level sparse inputs in a high-dimensional setting. We extend conditional Gaussian graphical models that have been previously developed as a Gaussian counterpart of conditional random fields (CRFs) for structured output prediction (Lafferty et al., 2001; Sohn and Kim, 2012; Wytock and Kolter, 2013). We modify both the output network and input-to-output mapping parameters of this single-level model to model the hierarchy in multi-level data. We adopt the CRF framework rather than extending regression models, because 1) the conditional dependencies represented by the edges in the probabilistic graphical models are more intuitively appealing and 2) the optimization problem for model estimation is convex, unlike the bi-convex optimization problem for learning regression models with correlated

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

noises to model correlated outputs.

We present an efficient learning algorithm that scales to large datasets in both complete and missing data settings. We first show that our multi-level model decomposes into a product of two components, one for sum variables and the other for difference variables derived from the original input and output variables. Then, we show that learning our model using this decomposed representation leads to an efficient optimization method, both when complete data are available and when only aggregate group-level data are available for randomly missing output variables. A naive approach of EM algorithm (Dempster et al., 1977) or a direct optimization after marginalizing out the missing variables would require the expensive computation of performing inference or evaluating the partition function repeatedly for each sample. Our learning algorithm based on the decomposition reduces or sidesteps this expensive computation, leading to orders of magnitude speed up for large datasets.

Related Works. CRFs have been extended to take into account hierarchy in data at pixel and super-pixel levels (Huang et al., 2011; Xie et al., 2019; Wang et al., 2016) or hierarchy in pixel labels (Ruiz-Sarmiento et al., 2019). These methods modeled discrete outputs at different levels with different variables or even with different CRFs, whereas we consider continuous outputs and learn a single output network that decomposes into network components at each level. Multi-layer networks have been introduced in social science and other network sciences to model different aspects of interactions among the entities or nodes as different layers of a network (Boccaletti et al., 2014; Finn et al., 2019; Basaras et al., 2017). These works have mainly focused on mining multi-layer networks for properties such as centralities, communities, and spreaders using graph-mining techniques or exponential random graph models (Wang et al., 2013), rather than estimating the networks themselves from grouped observations.

2 BACKGROUND

We consider the following problem set-up for predicting multivariate outputs given both individual-level and group-level inputs when individuals are grouped at multiple levels of hierarchy. Let $\mathbf{y} = [\mathbf{y}_1^T, \ldots, \mathbf{y}_m^T]^T \in \mathbb{R}^{mq}$, where $\mathbf{y}_i \in \mathbb{R}^q$ for $i = 1, \ldots, m$, denote q output variables for each of the m individuals in a group. Let $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_m^T, \mathbf{z}^T]^T \in \mathbb{R}^{mp+r}$ denote input variables, where $\mathbf{x}_i \in \mathbb{R}^p$ for $i = 1, \ldots, m$ represents individual-level p inputs for the ith individual and $\mathbf{z} \in \mathbb{R}^r$ represents group-level r inputs shared across all m individuals in the group.

Multi-level regression models (Snijders and Bosker, 2012) have been introduced to model the outputs for the ith individual in the group given the individual-level and group-level inputs as

$$\mathbf{y}_i = \mathbf{B}_w^T \mathbf{x}_i + \mathbf{B}_g^T \mathbf{z} + \mathbf{e}_i, \quad \mathbf{e}_i \sim \mathcal{N}\left(0, \mathbf{\Omega}^{-1}\right).$$

$\mathbf{B}_w \in \mathbb{R}^{p \times q}$ in the equation above are regression coefficients describing within-group effects of individual-level inputs on outputs, whereas $\mathbf{B}_g \in \mathbb{R}^{r \times q}$ represents across-group effects of group-level inputs on outputs. The $q \times q$ inverse covariance parameter $\mathbf{\Omega}$ models the correlated noise across q outputs. Collecting the models across all m individuals in the group, the model above can be equivalently written as

$$\mathbf{y} = \mathbf{B}^T \mathbf{x} + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}\left(0, \mathbf{I}_{m \times m} \otimes \mathbf{\Omega}^{-1}\right),$$

where $\mathbf{B} = \begin{bmatrix} \mathbf{I}_{m \times m} \otimes \mathbf{B}_w \\ \mathbf{1}_{1 \times m} \otimes \mathbf{B}_g \end{bmatrix}$ with an $m \times m$ identity matrix $\mathbf{I}_{m \times m}$ and an $1 \times m$ matrix of ones $\mathbf{1}_{1 \times m}$. The multi-level regression models above with fixed effects and with no group-specific effects have been extended to model group-specific random effects in $\mathbf{B}_w$ and $\mathbf{B}_g$.

In this paper, we consider problems in high-dimensional setting, focusing on multi-level models with fixed individual-level and group-level effects. In this case, a sparse estimate of the parameters of the multi-level regression model above could be obtained within the framework of multivariate regression with covariance estimation (MRCE) (Rothman et al., 2010). Given input and output data for n groups of m individuals, $\mathbf{Y} \in \mathbb{R}^{n \times mq}$ and $\mathbf{X} \in \mathbb{R}^{n \times (mp+r)}$, multi-level MRCE estimates the parameters by minimizing the L_1-regularized negative log-likelihood

$$\underset{\mathbf{\Omega} \succ 0, \mathbf{B}}{\operatorname{argmin}} - \log|\mathbf{\Omega}| + \frac{1}{n}\operatorname{tr}\left((\mathbf{Y} - \mathbf{XB})^T(\mathbf{Y} - \mathbf{XB})\mathbf{\Omega}\right) + \lambda_{\mathbf{B}_w}||\mathbf{B}_w||_1 + \lambda_{\mathbf{B}_g}||\mathbf{B}_g||_1 + \lambda_{\mathbf{\Omega}}||\mathbf{\Omega}||_1. \quad (1)$$

This optimization problem is bi-convex in $\mathbf{B}$ and $\mathbf{\Omega}$ and can be solved by alternately estimating $\mathbf{B}$ with Lasso while fixing $\mathbf{\Omega}$ and estimating $\mathbf{\Omega}$ with QUIC while fixing $\mathbf{B}$ (Rothman et al., 2010). While multi-level MRCE assumes a single-level output structure $\mathbf{\Omega}$, in the next section, we introduce a statistical approach for modeling output network structure at multiple levels, in addition to input effects on outputs at multiple levels.

3 SPARSE MULTI-LEVEL CONDITIONAL GAUSSIAN GRAPHICAL MODELS

We introduce multi-level conditional Gaussian graphical models, adopting a conditional random field framework that has been widely used for structured output

prediction. Our model builds on conditional Gaussian graphical models to model both multi-level input-to-output mapping and multi-level output network. We derive an alternative decomposed representation of our model, which provides additional insights into our model and also allows us to efficiently estimate the model for large problems both with complete data and in the presence of missing individual-level data.

3.1 The model

Given the problem setting in the previous section with q outputs for m individuals $\mathbf{y} \in \mathbb{R}^{mq}$ and p individual-level and r group-level inputs $\mathbf{x} \in \mathbb{R}^{mp+r}$ in each group, in order to model multi-level output structure and multi-level input-to-output mapping, we extend the previous single-level conditional Gaussian graphical model

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) = \exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{\Lambda}\mathbf{y} - \mathbf{x}^T\mathbf{\Theta}\mathbf{y}\right)/Z(\mathbf{x}), \quad (2)$$

with a single-level input-to-output mapping parameter $\mathbf{\Theta} \in \mathbb{R}^{p\times q}$ and $q \times q$ output network $\mathbf{\Lambda}$ for p inputs and q outputs with $m = 1$ and $r = 0$, where $Z(\mathbf{x}) = (2\pi)^{q/2}|\mathbf{\Lambda}|^{-1/2}\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{\Theta}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x}\right)$ is the normalization factor that ensures the probability distribution integrates to 1. Our multi-level extension of this model has multi-level input-to-output mapping parameter $\mathbf{\Theta} \in \mathbb{R}^{(mp+r)\times q}$

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{I}_{m\times m} \otimes \mathbf{\Pi} \\ \mathbf{1}_{1\times m} \otimes \mathbf{\Xi} \end{bmatrix},$$

where $\mathbf{\Pi} \in \mathbb{R}^{p\times q}$ and $\mathbf{\Xi} \in \mathbb{R}^{r\times q}$ model the influence of the within-group individual-level inputs $\mathbf{x}_i$'s for $i = 1, \ldots, m$ and group-level inputs $\mathbf{z}$ on outputs, respectively. Our multi-level model represents multi-level output network as an $mq \times mq$ positive definite matrix

$$\mathbf{\Lambda} = \mathbf{I}_{m\times m} \otimes \mathbf{\Delta} + \mathbf{1}_{m\times m} \otimes \mathbf{\Omega},$$

where each of the two $q \times q$ matrices, $\mathbf{\Delta}$ and $\mathbf{\Omega}$, models the individual-level and group-level output structure, respectively. Our multi-level network models the decomposition of the overall output correlation in $\mathbf{\Lambda}$ into two components, $\mathbf{\Delta}$ describing the correlated output behavior across individuals within each group at the lower-level of the data hierarchy and $\mathbf{\Omega}$ representing the correlated output behavior across groups at the higher-level of the hierarchy. The identity matrix $\mathbf{I}_{m\times m}$ in the Kronecker-product term for $\mathbf{\Delta}$ above allows the outputs of each individual in a group to have independent behavior, whereas the matrix $\mathbf{1}_{m\times m}$ in the Kronecker-product term for $\mathbf{\Omega}$ forces the outputs of all individuals in each group to be tied up to model group-level correlated output behavior.

3.2 Challenges in learning multi-level conditional Gaussian graphical models

A naive approach for estimating our multi-level model is to minimize the L_1-regularized negative data log-likelihood by directly adopting the alternate Newton coordinate descent method, which has been previously developed for an efficient optimization of the single-level model (McCarter and Kim, 2016). Using this strategy, given mean-centered output data $\mathbf{Y} \in \mathbb{R}^{n\times mq}$ and input data $\mathbf{X} \in \mathbb{R}^{n\times(mp+r)}$ for n groups, a sparse estimate of the parameters of our model can be obtained by solving the following optimization problem:

$$\underset{\mathbf{\Lambda}\succ 0, \mathbf{\Pi}, \mathbf{\Xi}}{\operatorname{argmin}} -\log|\mathbf{\Lambda}| + \operatorname{tr}(\mathbf{S_y}\mathbf{\Lambda} + 2\mathbf{S_{yx}}\mathbf{\Theta} + \mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{S_x}\mathbf{\Theta}) + \lambda_{\mathbf{\Lambda}}(\|\mathbf{\Delta}\|_1 + \|\mathbf{\Omega}\|_1) + \lambda_{\mathbf{\Theta}}(\|\mathbf{\Pi}\|_1 + \|\mathbf{\Xi}\|_1),$$

where $\mathbf{S_y} = \frac{1}{n}\mathbf{Y}^T\mathbf{Y}$, $\mathbf{S_{yx}} = \frac{1}{n}\mathbf{Y}^T\mathbf{X}$, and $\mathbf{S_x} = \frac{1}{n}\mathbf{X}^T\mathbf{X}$ are the sample covariance matrices. The $||\cdot||_1$ is an L_1 penalty and $\lambda_{\mathbf{\Lambda}}$ and $\lambda_{\mathbf{\Theta}}$ are regularization parameters controlling the sparsity level in the estimated parameters. We do not penalize the diagonal elements of $\mathbf{\Delta}$ and $\mathbf{\Omega}$, as is typically done in the estimation of Gaussian graphical models.

However, with this optimization strategy, the computational efficiency of the single-level model estimation does not directly translate to the multi-level model for the following two reasons. First, the line search and evaluation of the objective involve the costly computation of $\log|\mathbf{\Lambda}|$ and $\mathbf{\Lambda}^{-1}$ for the large $mq \times mq$ matrix $\mathbf{\Lambda}$. Second, it is necessary to maintain large sample covariance matrices throughout the optimization, including the $mq \times mq$ matrix $\mathbf{S_y}$, $mq \times (mp+r)$ matrix $\mathbf{S_{yx}}$, and $(mp+r)\times(mp+r)$ matrix $\mathbf{S_x}$. In the next section, we introduce a decomposition of our multi-level model as an alternative representation and show that this decomposition allows us to sidestep these two key challenges for an efficient estimation of the multi-level model.

3.3 Decomposed representation

We show in Theorem 1 below that the multi-level model based on Eq. (2) can be written as an equivalent form of a product of two components, one for sum variables and the other for $m(m-1)/2$ pairwise difference variables derived from inputs and outputs for m individuals in each group.

Theorem 1. *Let* $\mathbf{y}_s = \sum_{i=1}^m \mathbf{y}_i$ *and* $\mathbf{w} = \begin{bmatrix}\mathbf{x}_s^T & \mathbf{z}^T\end{bmatrix}^T$ *with* $\mathbf{x}_s = \sum_{i=1}^m \mathbf{x}_i$ *be sum variable. Let* $\mathbf{y}_{ij} = \mathbf{y}_i - \mathbf{y}_j$ *and* $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ *be difference variables for all pairs* (i,j)*'s for* $i, j \in \{1, \ldots, m\}$*. Assuming* $\mathbf{\Delta\Omega} = \mathbf{\Omega\Delta}$*, the following decomposition of the multi-level conditional Gaussian graphical model holds:*

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) = f_s(\mathbf{y}_s \mid \mathbf{w}; \mathbf{V}, \mathbf{F}) f_d(\mathbf{y}_{ij}\text{'s} \mid \mathbf{x}_{ij}\text{'s}; \mathbf{\Gamma}, \mathbf{\Psi}).$$

The sum component in this decomposition is given as

$$f_s(\mathbf{y}_s \mid \mathbf{w}; \mathbf{V}, \mathbf{F}) = \exp\left(-\frac{1}{2}\mathbf{y}_s^T\mathbf{V}\mathbf{y}_s - \mathbf{w}^T\mathbf{F}\mathbf{y}_s\right)/Z_s(\mathbf{x})$$

with $\mathbf{V} = \boldsymbol{\Omega} + \frac{1}{m}\boldsymbol{\Delta}$ *and* $\mathbf{F} = \left[\frac{1}{m}\boldsymbol{\Pi}^T\ \boldsymbol{\Xi}^T\right]^T$. *The difference component is given as*

$$f_d(\mathbf{y}_{ij}\text{'s} \mid \mathbf{x}_{ij}\text{'s}; \boldsymbol{\Gamma}, \boldsymbol{\Psi}) = \left[\prod_{i<j}\exp\left(-\frac{1}{2}\mathbf{y}_{ij}^T\boldsymbol{\Gamma}\mathbf{y}_{ij} - \mathbf{x}_{ij}^T\boldsymbol{\Psi}\mathbf{y}_{ij}\right)/Z_{ij}(\mathbf{x})\right]/Z_d(\mathbf{x})$$

with $\boldsymbol{\Gamma} = \frac{1}{m}\boldsymbol{\Delta}$ *and* $\boldsymbol{\Psi} = \frac{1}{m}\boldsymbol{\Pi}$. *The normalization factors in the sum and difference components are given as*

$$Z_s(\mathbf{x}) = (2\pi)^{q/2}|\mathbf{V}|^{-1/2}m^{-q/2}\exp\left(-\frac{1}{2}\mathbf{w}^T\mathbf{F}\mathbf{V}^{-1}\mathbf{F}^T\mathbf{w}\right)$$

$$Z_{ij}(\mathbf{x}) = \exp\left(-\frac{1}{2}\mathbf{x}_{ij}^T\boldsymbol{\Psi}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Psi}\mathbf{x}_{ij}\right)$$

$$Z_d(\mathbf{x}) = (2\pi)^{(m-1)q/2}|\boldsymbol{\Gamma}|^{-(m-1)/2}m^{-q(m-1)/2}.$$

The proof is in Supplementary Information A.

The decomposition in Theorem 1 turns the original multi-level model with $mq \times mq$ output network parameter $\boldsymbol{\Lambda}$ and $(mp+r) \times q$ input-to-output mapping parameter $\boldsymbol{\Theta}$ into another form in which each component is described by significantly smaller $q \times q$ output networks, $\mathbf{V} = \boldsymbol{\Omega} + \frac{1}{m}\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma} = \frac{1}{m}\boldsymbol{\Delta}$, and $(p+r) \times q$ and $p \times q$ input-to-output mapping parameters, $\mathbf{F} = \left[\frac{1}{m}\boldsymbol{\Pi}^T\ \boldsymbol{\Xi}^T\right]^T$ and $\boldsymbol{\Psi} = \frac{1}{m}\boldsymbol{\Pi}$. Each component in this decomposition takes the form that resembles conditional Gaussian graphical models with p inputs and q outputs. In particular, when $m = 2$, the multi-level model decomposes exactly into two conditional Gaussian graphical models, each modeling sums and differences of inputs and outputs for $m = 2$ individuals per group, as we state in the following corollary.

Corollary 1. *Let* $\mathbf{y}_s = \mathbf{y}_1 + \mathbf{y}_2$ *and* $\mathbf{x}_s = \mathbf{x}_1 + \mathbf{x}_2$ *be the sum of the input and output variables of* $m = 2$ *individuals in a group. Let* $\mathbf{y}_d = \mathbf{y}_1 - \mathbf{y}_2$ *and* $\mathbf{x}_d = \mathbf{x}_1 - \mathbf{x}_2$ *represent the difference of the input and output variables. Then, our multi-level model in Eq. (2) factorizes into two conditional Gaussian graphical models, one based on the sum variables* $\mathbf{y}_s$ *and* $\mathbf{w} = [\mathbf{x}_s, \mathbf{z}]$ *and the other based on the difference variables* $\mathbf{y}_d$ *and* $\mathbf{x}_d$*:*

$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\Lambda}, \boldsymbol{\Theta}) = p(\mathbf{y}_s \mid \mathbf{w}; \mathbf{V}, \mathbf{F})p(\mathbf{y}_d \mid \mathbf{x}_d; \boldsymbol{\Gamma}, \boldsymbol{\Psi}),$$

where

$$p(\mathbf{y}_s \mid \mathbf{w}; \mathbf{V}, \mathbf{F}) = (2\pi)^{-q/2}|\mathbf{V}|^{1/2} \cdot \exp\big(-1/2\big(\mathbf{y}_s^T\mathbf{V}\mathbf{y}_s + 2\mathbf{w}^T\mathbf{F}\mathbf{y}_s + \mathbf{w}^T\mathbf{F}\mathbf{V}^{-1}\mathbf{F}^T\mathbf{w}\big)\big)$$

$$p(\mathbf{y}_d \mid \mathbf{x}_d; \boldsymbol{\Psi}, \boldsymbol{\Gamma}) = (2\pi)^{-q/2}|\boldsymbol{\Gamma}|^{1/2} \cdot \exp\big(-1/2\big(\mathbf{y}_d^T\boldsymbol{\Gamma}\mathbf{y}_d + 2\mathbf{x}_d^T\boldsymbol{\Psi}\mathbf{y}_d + \mathbf{x}_d^T\boldsymbol{\Psi}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Psi}^T\mathbf{x}_d\big)\big)$$

with $\mathbf{V} = \boldsymbol{\Omega} + \frac{1}{2}\boldsymbol{\Delta}$, $\mathbf{F} = \left[\frac{1}{2}\boldsymbol{\Pi}^T\ \boldsymbol{\Xi}^T\right]^T$, $\boldsymbol{\Gamma} = \frac{1}{2}\boldsymbol{\Delta}$, *and* $\boldsymbol{\Psi} = \frac{1}{2}\boldsymbol{\Pi}$. *The proof is provided in Supplementary Information A.*

By learning the model parameters based on this alternative representation, it is possible to remove the two computational bottlenecks in the naive application of the optimization method for single-level model. First, during line search and evaluation of the objective, we only need to compute the determinant and inverse of far smaller $q \times q$ matrices $\mathbf{V}$ and $\boldsymbol{\Gamma}$ rather than for a large $mq \times mq$ matrix $\boldsymbol{\Lambda}$.

Second, since the decomposed model is defined on sum and difference variables that collapse the original variables for m individuals in each group, the sufficient statistics that need to be pre-computed for optimization also collapse across the m individuals, significantly reducing the computation time. The sum component of the decomposed representation models the sum data $\mathbf{Y}_s$ of size $n \times q$ and $\mathbf{X}_s$ of size $n \times (p+r)$ obtained from the original data $\mathbf{Y}$ of size $n \times mq$ and $\mathbf{X}$ of size $n \times (mp+r)$. This collapsed data means collapsed covariance matrices $\mathbf{S}_{\mathbf{y}_s}$ of size $q \times q$, $\mathbf{S}_{\mathbf{w}}$ of size $(p+r) \times (p+r)$, and $\mathbf{S}_{\mathbf{y}\mathbf{w}_s}$ of size $q \times (p+r)$. The difference component of the decomposed representation takes the difference data $\mathbf{Y}_{ij}$ of size $nm(m-1)/2 \times q$ and $\mathbf{X}_{ij}$ of size $nm(m-1)/2 \times (p+r)$ derived from $\mathbf{Y}$ and $\mathbf{X}$ for (i,j) pair of individuals in each group. Although the effective sample size increases from n to $nm(m-1)/2$ due to the computation of pairwise differences for m individuals per group, this is one time computation to be completed prior to the optimization, and during the optimization, we again only need to maintain significantly smaller covariance matrices, $\mathbf{S}_{\mathbf{y}_d}$ of size $q \times q$, $\mathbf{S}_{\mathbf{y}\mathbf{x}_d}$ of size $q \times (p+r)$. $\mathbf{S}_{\mathbf{x}_d}$ of size $(p+r) \times (p+r)$.

Then, we extend the alternate Newton coordinate descent method for the single-level model to fit our multi-level model (McCarter and Kim, 2016). In each iteration, we alternately update each of $\boldsymbol{\Pi}$, $\boldsymbol{\Xi}$, $\boldsymbol{\Delta}$, and $\boldsymbol{\Omega}$, while fixing the others. With $\boldsymbol{\Delta}$ and $\boldsymbol{\Omega}$ fixed, updating $\boldsymbol{\Pi}$ and $\boldsymbol{\Xi}$ simply becomes quadratic minimization problems with L_1 regularization, which can be solved efficiently with the coordinate descent method for Lasso. To update $\boldsymbol{\Delta}$ and $\boldsymbol{\Omega}$, we apply Newton's method, where the Newton direction is found by minimizing the second-order approximation of the negative data log-likelihood with L_1 regularization via the Lasso coordinate descent. For problems with large p and q, where the covariance matrices do not fit in the memory, we extend the alternating Newton block coordinate descent method previously developed for single-level conditional Gaussian graphical model optimization to remove the memory requirement. We provide the details of this optimization algorithm in Supple-

mentary Information B.

3.4 Learning with missing individual-level observations

In multi-level data, output observations may be available only at the group-level but not at the individual-level. For example, average scores of certain subjects across all students in a school may be available but scores of individual students may not. In such cases, we show estimating our multi-level model based on the decomposition of the original model leads to significant improvement in computation time, compared to estimating the model based on the original representation. Below, we illustrate this for the case of group size $m = 2$, though this generalizes to arbitrary m.

Theorem 2 below states that with missing individual-level data, our multi-level model in Eq. (2) with $m = 2$ collapses into another conditional Gaussian graphical model.

Theorem 2. *When data are available for individual members of the group for a subset $V \subset \{1, \ldots, q\}$ of the output variables but only for the group-level sums for the other output variables $H = \{1, \ldots, q\} - V$, the multi-level conditional Gaussian graphical model with $m = 2$ collapses into the following conditional Gaussian graphical model:*

$$p(\mathbf{y}_s^H, \mathbf{y}^V \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) = 1/Z(\mathbf{x}) \cdot \exp\left(-\frac{1}{2} \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}^V \end{bmatrix}^T \tilde{\mathbf{\Lambda}} \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}^V \end{bmatrix} - \mathbf{x}^T \tilde{\mathbf{\Theta}} \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}^V \end{bmatrix} \right), \quad (3)$$

where $\tilde{\mathbf{\Lambda}} = \mathbf{C}^{+T} \mathbf{\Lambda} \mathbf{C}^{+}$ and $\tilde{\mathbf{\Theta}}^T = \mathbf{C}^{+T} \mathbf{\Theta}^T$ with $\mathbf{C}^+$ representing the Moore-Penrose inverse of $\mathbf{C} = \begin{bmatrix} \mathbf{C}_s^H \\ \mathbf{C}_a^V \end{bmatrix}$. $\mathbf{C}_s^H$ and $\mathbf{C}_a^V$ are the submatrices of $\mathbf{C}_s = \mathbf{1}_{1\times m} \otimes \mathbf{I}_{q\times q}$ and $\mathbf{C}_a = \mathbf{I}_{mq\times mq}$ for $m = 2$, with only the rows corresponding to the variables in $\mathbf{y}_s^H$ and $\mathbf{y}^V$, respectively. The proof is given in Supplementary Information A.

Theorem 2 suggests the optimization method developed for a single-level conditional Gaussian graphical model could be used, after imputing the missing data (Corollary 2, Supplementary Information A). However, this strategy, in addition to the two challenges discussed in the previous section, poses another computational challenge of evaluating the partition function $Z(\mathbf{x})$ in Eq. (3) that contains a log-determinant and inverse of the collapsed network $\tilde{\mathbf{\Lambda}} = \mathbf{C}^{+T} \mathbf{\Lambda} \mathbf{C}^{+}$. Since the network is collapsed differently via different $\mathbf{C}^+$ due to different sets of observed outputs V for each group-level sample, evaluating $Z(\mathbf{x})$ for each of the n group-level samples in each iteration of optimization would be prohibitively expensive.

The decomposed representation reduces this computational burden from collapsed $mq \times mq$ matrix $\mathbf{\Lambda}$ to smaller collapsed $q \times q$ matrix $\mathbf{\Gamma}$. This can be seen for $m = 2$ by integrating out the missing individual-level outputs in the decomposed representation:

$$p(\mathbf{y}_s^H, \mathbf{y}^V | \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) = \int p(\mathbf{y}_s | \mathbf{x}_s; \mathbf{V}, \mathbf{F}) p(\mathbf{y}_d | \mathbf{x}_d; \mathbf{\Psi}, \mathbf{\Gamma}) d\mathbf{y}^H$$
$$= p(\mathbf{y}_s | \mathbf{x}_s; \mathbf{V}, \mathbf{F}) \int p(\mathbf{y}_d | \mathbf{x}_d; \mathbf{\Psi}, \mathbf{\Gamma}) d\mathbf{y}^H.$$

We notice that the integral now applies only to the difference component, with reduced cost of evaluating the partition function with collapsed $\mathbf{\Gamma}$ for each sample. In particular, in the special case of diagonal $\mathbf{\Gamma}$ with $[\gamma_1, \ldots, \gamma_q]$ in the diagonal elements, the integral above leads to

$$p(\mathbf{y}_s^H, \mathbf{y}^V | \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) = (2\pi)^{-q/2} |\mathbf{V}|^{1/2}$$
$$\cdot \exp\big(-1/2 \big(\mathbf{y}_s^T \mathbf{V} \mathbf{y}_s + 2\mathbf{x}_s^T \mathbf{F} \mathbf{y}_s + \mathbf{x}_s^T \mathbf{F} \mathbf{V}^{-1} \mathbf{F}^T \mathbf{x}_s \big) \big)$$
$$\cdot (2\pi)^{-q/2} \prod_{i \in V} |\gamma_i|^{1/2}$$
$$\cdot \exp\big(-1/2 \big(\mathbf{y}_d^{VT} \boldsymbol{\gamma} \mathbf{y}_d^V + 2\mathbf{x}_d^T \mathbf{\Psi} \mathbf{y}_d^V + \mathbf{x}_d^T \mathbf{\Psi} \boldsymbol{\gamma}^{-1} \mathbf{\Psi}^T \mathbf{x}_d \big) \big),$$

where evaluating the determinant and inverse of collapsed $\mathbf{\Gamma}$ in partition function becomes trivial, eliminating the challenges involved in the network that collapses differently for each sample. Thus, when data contain randomly missing individual-level data, learning the model based on the decomposed representation is significantly more efficient than using the original representation in Eq. (3). It is straightforward to show that this result generalizes to $m > 2$.

4 EXPERIMENTS

4.1 Simulation Study

We compare the performance of our method, single-level conditional Gaussian graphical models, and multi-level MRCE on simulated data in terms of the accuracy of graph structure recovery and prediction. To simulate a dataset, we first generate two scale-free networks of size $q = 50$ for $\mathbf{\Omega}$ and $\mathbf{\Delta}$. We draw edge weights from uniform distribution $U[a, b]$, where a and b are set for each simulation scenario, and set signs randomly to positive or negative. We then add values drawn from $U[1, 4]$ to the diagonal elements of both network matrices. Given $\mathbf{\Omega}$ and $\mathbf{\Delta}$, we construct $\mathbf{\Lambda}$ with group size $m = 4$ or $m = 6$, and then add a constant to diagonals to make the entire matrix $\mathbf{\Lambda}$ positive definite. For $\mathbf{\Pi}$ and $\mathbf{\Xi}$, we generate random sparse matrices of a chosen density of non-zero elements, with values drawn from $U[c, d]$ and signs randomly set to positive or negative. From $\mathbf{\Pi}$ and $\mathbf{\Xi}$, we then construct

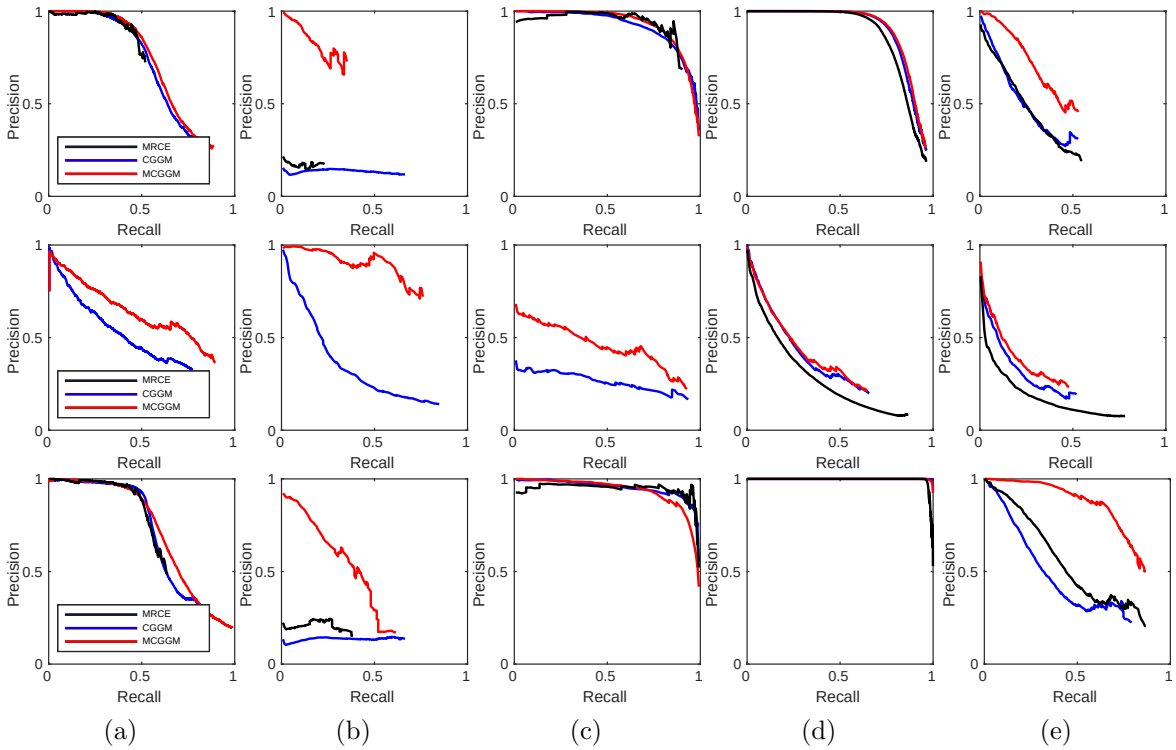


Figure 1: Comparison of methods on simulated data. Precision-recall curves for the recovery of (a) $\mathbf{\Omega} + \mathbf{\Delta}$, (b) $\mathbf{\Omega}$, and (c) $\mathbf{\Delta}$, (d) $\mathbf{\Pi}$, and (e) $\mathbf{\Xi}$. Top row for Case 1, middle row for Case 2, and bottom row for Case 3 with group size $m = 6$.

Table 1: Prediction Errors in Simulation Study

	MCGGM	CGGM	MRCE
Case 1	0.242	0.258	0.294
Case 2	0.054	0.055	0.061
Case 3	0.411	0.604	1.067

the multi-level input-to-output mapping parameter $\mathbf{\Theta}$. We vary the edge strengths and sparsity levels of $\mathbf{\Omega}$, $\mathbf{\Delta}$, $\mathbf{\Pi}$, and $\mathbf{\Xi}$ under each of the five scenarios as follows: Case 1 with $U[0.2, 0.8]$ and with sparsity at 5% for all parameters; Case 2 with $U[0.8, 1.2]$ for $\mathbf{\Omega}$ and $\mathbf{\Delta}$ and $U[0.2, 0.6]$ for $\mathbf{\Pi}$ and $\mathbf{\Xi}$, with sparsity at 5% for all parameters; Case 3 with $U[0.2, 0.6]$ for $\mathbf{\Omega}$ and $\mathbf{\Delta}$ and $U[0.8, 1.2]$ for $\mathbf{\Pi}$ and $\mathbf{\Xi}$, with sparsity 5% for all parameters; Case 4 with $U[0.2, 0.8]$ for all parameters, with sparsity at 2% for $\mathbf{\Omega}$ and 5% for the others; and Case 5 with $U[0.2, 0.8]$ for all parameters, with sparsity at 2% for $\mathbf{\Delta}$ and 5% for the others.

We generate $p = 500$ individual-level input data and $r = 100$ group-level input data as random binary matrices. Given the parameters and input data, we simulate output data from the multi-level conditional Gaussian graphical model based on Eq. (2). Each dataset consists of $n = 300$ samples. We generate 30 datasets under each scenario and report results averaged over these replicates.

Out of 300 samples in each dataset, we use 240 samples to train each model and evaluate its prediction errors on the other 60 samples. During training, to select the optimal regularization parameters, we use three-fold cross validation for multi-level MRCE and 70%-30% split for training and validation data for multi-level and single-level conditional Gaussian graphical models. Since single-level conditional Gaussian graphical models do not consider group structure, we treat the group-level and individual-level observations as independent samples, effectively increasing the sample size per dataset to nm.

We compare the performance of the different methods on the recovery of true parameters under different simulation scenarios with group size $m = 6$, using precision-recall curves. For the models that estimate only a single-level network, we compare their network estimates to $\mathbf{\Omega}$, $\mathbf{\Delta}$, and $\mathbf{\Omega} + \mathbf{\Delta}$. As can be seen in Figure 1, our method recovers network structures and input-to-output mapping parameters at both individual and group level across the first three scenarios with higher accuracy than the other methods. The results from Cases 4 and 5 as well as the same set of five scenarios but with clustered networks for $\mathbf{\Omega}$ (Supplementary Information Figures 4-6) are consistent with those in Figure 1. We also repeated all experiments with group size $m = 4$, and the results were consistent with those from $m = 6$.

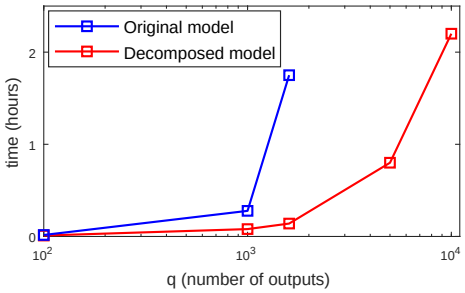


Figure 2: Comparison of computation time of the multi-level model using the original and decomposed representations. The number of inputs was fixed at $p = 10,000$.

We compare the performance of the methods on predicting the outputs in the test set, using the mean-squared error across all group and individuals $(\sum_{n'} \sum_{m'} (\hat{\mathbf{y}}_{nm} - \mathbf{y}_{nm})^2/(nmq))$ for predicted $\hat{\mathbf{y}}_{nm}$ from each method and observed $\mathbf{y}_{nm}$. Table 1 shows that our method achieves lower prediction errors than the other methods across all simulation scenarios.

To assess the advantage of using the decomposition of our model in Theorem 1 for efficient optimization, we compare the computation time of the optimization methods based on the original representation and based on the decomposition, assuming $\boldsymbol{\Delta}$ is a diagonal matrix. As shown in Figure 2, as the problem size increases, the optimization based on the decomposed model is substantially more efficient.

4.2 Financial data

We apply our approach to the daily stock prices of the S&P500 companies from 2013 to 2018 to learn the multi-level model that can predict the stock prices in the future using the prices in the past. We consider the model with group size $m = 4$, where each group member corresponds to the daily close, open, high, and low prices, and use the daily closing prices of various futures, commodities, and foreign major indices as the group-level inputs. From the S&P500, we obtained data for 467 companies, after removing companies with duplicate listing as well as those that underwent significant corporate actions or lacked data from 2013 to 2018. Among these, we selected the members of the S&P100 for $q = 97$ outputs. We considered $r = 45$ group-level inputs composed of futures closing prices for 25 various commodities, bonds, and currencies and major indices of 20 large non-US economies. The open-open prices series was used for European indices because they close during the market hour in the US. We considered the z-scored daily log-return for each input and output variable.

Our task is to forecast the next day's close, open, high, and low prices of 97 companies, using the previous day's close, open, high, and low prices of 467 companies and closing prices of 45 group-level variables. Our model is trained using the data from each full year (250 or 251 days) from 2013 to 2018, and tested on the first 100 days of the succeeding year. For 2018, the test data was the first 74 days of 2019. The mean and standard deviation of training data were used to z-score the respective test data. Both single-level conditional Gaussian graphical model and multi-level MRCE received nm stacked samples as in simulation studies.

Table 2: Prediction Error on S&P100 Data

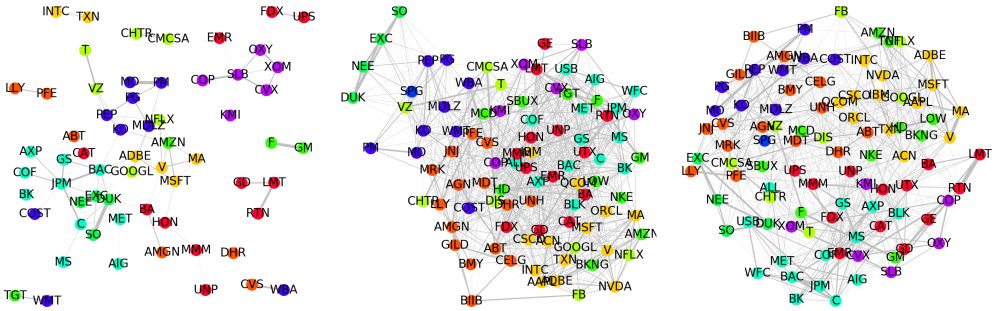
	MCGGM	CGGM	MRCE
2013-14	1.044	1.053	1.098
2014-15	1.131	1.122	1.167
2015-16	1.061	1.064	1.182
2016-17	0.578	0.584	0.718
2017-18	2.365	2.462	2.522
2018-19	0.817	0.832	0.880

Our method almost always achieves lower prediction errors than the methods that do not account for group structure (Table 2). The high prediction errors across different methods for year 2017-2018 can be explained by the high volatility of early 2018. We also compare the $\boldsymbol{\Omega}$ and $\boldsymbol{\Delta}$ in our estimated multi-level model obtained from the training data in year 2018 with the network recovered using the single-level model (Figure 3). We find that $\boldsymbol{\Omega}$ recovers the network of very closely related companies (such as LLY-PFE, FDX-UPS, V-MA, F-GM, TGT-WMT, GD-LMT-RTN, etc) whose daily stock fluctuations closely mirror each other and whose financial performances are often used to forecast each others'. These strong associations among related companies are not immediately apparent in the hairball produced in the network estimate from the single-level conditional Gaussian graphical models.

4.3 Genomics data

We apply single-level and multi-level conditional Gaussian graphical models to the genetic variant and expression data from the Genotype-Tissue Expression (GTEx) project (GTEx Consortium, 2017) to learn the gene network influenced by *cis*-acting and *trans*-acting variants. In diploid organisms with two sets of chromosomes of maternal and paternal origins, *cis*-acting variants (individual-level inputs) influence the expression of genes that reside on the same chromosome as the given variant, whereas *trans*-acting variants (group-level inputs) influence the expression of genes on both chromosomes (Wittkopp et al., 2004; McManus et al., 2010).

We analyze the expression data for 7,650 genes from cultured fibroblasts, and genotype data for 374,334 genetic variants in chromosome 1 from 483 donors. One of the challenges in this problem is that with the expression data, maternal and paternal expression mea-



(a) (b) (c)

Figure 3: Networks recovered from 2018 financial data. (a) $\mathbf{\Omega}$ from multi-level model, (b) $\mathbf{\Delta}$ from multi-level model, and (c) $\mathbf{\Lambda}$ from single-level conditional Gaussian graphical model. The node colors represent sectors: red for industrial; orange for health care; yellow for information technology; green for consumer discretionary; teal for financial; blue for consumer staples; and purple for energy.

surements may be missing for some genes in some samples and only the total expression levels may be available, because the two copies of genes with identical sequences are indistinguishable. Approximately 40% of genes have missing maternal and paternal expression values in this dataset. We use 350 samples as the training data and use the remaining 133 to select the regularization parameters. For the single-level model, we use the total expression values and genotypes summed across maternal and paternal chromosomes.

The runtimes of both methods vary across 5 to 9 hours on 16-core and 32GB-RAM machines, indicating that our multi-level model does not take significantly longer time than the single-level counterpart, because of the decomposition in Theorem 1 for efficient optimization.

Unlike the single-level model, our method is able to distinguish between *cis*-acting and *trans*-acting variants, which leads to the following sets of novel findings that can be made only by our approach. First, our approach identifies potential *cis*-acting variants (pink in Supplementary Information Figure 7), which are distant from the genes they regulate and have not been previously annotated. Second, in our estimated multi-level model, the average absolute effect sizes of *cis*-acting variants was 0.19±0.35, while that of *trans*-acting variants was 0.04±0.11. This corresponds to the widely held view that the effects of *cis*-variants tend to be stronger and localized, while the effects of *trans*-variants spread small effects over a large number of genes. Third, compared to *trans*-acting variants, more *cis*-acting variants were identified as variants that are thought to act mainly through *cis*-regulatory mechanisms, such as promoter and UTR variants (Table 3). Finally, many long non-coding RNAs (lncRNA) are thought to regulate genes in *trans*, and we found many of our potential *trans*-acting variants residing in high-confidence lncRNA regions (Volders et al., 2018).

Table 3: The number of *cis*-acting and *trans*-acting variants with functional annotation

	cis	*trans*
unique genomic variants	252	187
enhancer	1	2
promoter	18	4
open chromatin region	17	4
promoter flanking region	15	5
CTCF binding site	2	2
TF binding site	9	0
3 prime UTR variant	11	1
5 prime UTR variant	11	0
intron variant	121	89
↳ lncRNA region	90	63
missense variant	1	0
non coding transcript exon variant	24	0
synonymous variant	6	4

5 CONCLUSION

In this paper, we proposed sparse multi-level conditional Gaussian graphical models for recovering the underlying output network structure and input features influencing outputs given multi-level grouped data. We introduced an alternative decomposed representation of the model that enables an efficient parameter estimation for large datasets.

Acknowledgements

We thank Calvin McCarter for his contribution to the initial work of this paper. GK was a predoctoral trainee supported by NIH T32 training grant T32 EB009403 as part of the HHMI-NIBIB Interfaces Initiative. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant numbers ACI-1548562 and ACI-1445606.

References

P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas. Identifying influential spreaders in complex multilayer networks: A centrality perspective. *IEEE Transactions on Network Science and Engineering*, 6(1):31–45, 2017.

S. Boccaletti, G. Bianconi, R. Criado, C. I. Del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 39:1–38, 1977.

K. R. Finn, M. J. Silk, M. A. Porter, and N. Pinter-Wollman. The use of multilayer network analysis in animal behaviour. *Animal behaviour*, 149:7–22, 2019.

GTEx Consortium. Genetic effects on gene expression across human tissues. *Nature*, 550(7675):204–213, 2017.

Q. Huang, M. Han, B. Wu, and S. Ioffe. A hierarchical conditional random field model for labeling and segmenting images of street scenes. In *CVPR 2011*, pages 1953–1960. IEEE, 2011.

J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, volume 951, pages 282–289, 2001.

C. McCarter and S. Kim. Large-scale optimization algorithms for sparse conditional Gaussian graphical models. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 528–537, 2016.

C. J. McManus, J. Coolon, M. Duff, J. Eipper-Mains, B. Graveley, and P. Wittkopp. Regulatory divergence in Drosophila revealed by mRNA-seq. *Genome research*, 20(6):816–825, 2010.

A. J. Rothman, E. Levina, and J. Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19 (4):947–962, 2010.

J.-R. Ruiz-Sarmiento, C. Galindo, J. Monroy, F.-A. Moreno, and J. Gonzalez-Jimenez. Ontology-based conditional random fields for object recognition. *Knowledge-Based Systems*, 168:100–108, 2019.

J. R. Silvester. Determinants of block matrices. *The Mathematical Gazette*, 84(501):460–467, 2000.

T. A. Snijders and R. Bosker. Multilevel analysis: An introduction to basic and advanced multilevel modeling, 2012.

K.-A. Sohn and S. Kim. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, pages 1081–1089, 2012.

P.-J. Volders, J. Anckaert, K. Verheggen, J. Nuytens, L. Martens, P. Mestdagh, and J. Vandesompele. LNCipedia 5: towards a reference set of human long non-coding RNAs. *Nucleic Acids Research*, 47(D1): D135–D139, 10 2018.

F. Wang, Y. Wu, M. Li, P. Zhang, and Q. Zhang. Adaptive hybrid conditional random field model for sar image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):537–550, 2016.

P. Wang, G. Robins, P. Pattison, and E. Lazega. Exponential random graph models for multilevel networks. *Social Networks*, 35(1):96–115, 2013.

P. Wittkopp, B. Haerum, and A. Clark. Evolutionary changes in cis and trans gene regulation. *Nature*, 430(6995):85–88, 2004.

M. Wytock and Z. Kolter. Sparse Gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1265–1273, 2013.

X. Xie, G. Xie, X. Xu, L. Cui, and J. Ren. Automatic image segmentation with superpixels and image-level labels. *IEEE Access*, 7:10999–11009, 2019.

Supplementary Information A: Proofs of theorems

Proof of Theorem 1

Proof. We first define the following quantities.

$$\mathbf{x}_s\mathbf{x}_s^T = \sum_i^m \mathbf{x}_i\mathbf{x}_i^T + \sum_{i<j}^m \left(\mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_i^T\right) \tag{4}$$

$$\mathbf{x}_{ij}\mathbf{x}_{ij}^T = \mathbf{x}_i\mathbf{x}_i^T + \mathbf{x}_j\mathbf{x}_j^T - \mathbf{x}_i\mathbf{x}_j^T - \mathbf{x}_j\mathbf{x}_i^T \tag{5}$$

$$\sum_{i<j}^m \mathbf{x}_{ij}\mathbf{x}_{ij}^T = (m-1)\sum_i^m \mathbf{x}_i\mathbf{x}_i^T - \sum_{i<j}^m \left(\mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_i^T\right) \tag{6}$$

$$\sum_i^m \mathbf{x}_i\mathbf{x}_i^T = \frac{1}{m}\left(\mathbf{x}_s\mathbf{x}_s^T + \sum_{i<j}^m \mathbf{x}_{ij}\mathbf{x}_{ij}^T\right) \qquad \text{From Eqs. (4) and (6)} \tag{7}$$

To prove the theorem, we visit each term in the model in Theorem 1, $\mathbf{y}^T\mathbf{\Lambda}\mathbf{y}$, $\mathbf{x}^T\mathbf{\Theta}\mathbf{y}$, $|\mathbf{\Lambda}|$, and $\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{\Theta}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x}\right)$ and re-write it as follows:

$$\begin{aligned}
\operatorname{tr}(\mathbf{y}\mathbf{y}^T\mathbf{\Lambda}) &= \operatorname{tr}(\left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T\right)(\mathbf{\Omega}+\mathbf{\Delta}) + \left(\sum_{i<j}^m \left(\mathbf{y}_i\mathbf{y}_j^T + \mathbf{y}_j\mathbf{y}_i^T\right)\right)\mathbf{\Omega}) \\
&= \operatorname{tr}(\left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T + \sum_{i<j}^m \left(\mathbf{y}_i\mathbf{y}_j^T + \mathbf{y}_j\mathbf{y}_i^T\right)\right)\mathbf{\Omega} + \left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T\right)\mathbf{\Delta}) \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{y}_s^T\mathbf{\Omega} + \left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T\right)\mathbf{\Delta}) \qquad \text{From Eq. (4)} \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{y}_s^T\mathbf{\Omega} + \frac{1}{m}\left(\mathbf{y}_s\mathbf{y}_s^T + \sum_{i<j}^m \mathbf{y}_{ij}\mathbf{y}_{ij}^T\right)\mathbf{\Delta}) \qquad \text{From Eq. (7)} \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{y}_s^T\left(\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}\right) + \left(\sum_{i<j}^m \mathbf{y}_{ij}\mathbf{y}_{ij}^T\right)\frac{1}{m}\mathbf{\Delta}) \\
\operatorname{tr}(\mathbf{y}\mathbf{x}^T\mathbf{\Theta}) &= \operatorname{tr}(\left(\sum_i^m \mathbf{y}_i\mathbf{x}_i^T\right)\mathbf{\Pi} + \left(\sum_i^m \mathbf{y}_i\mathbf{z}^T\right)\mathbf{\Xi}) \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{x}_s^T\left(\frac{1}{m}\mathbf{\Pi}\right) + \left(\sum_{i<j}^m \mathbf{y}_{ij}\mathbf{x}_{ij}^T\right)\frac{1}{m}\mathbf{\Pi} + \mathbf{y}_s\mathbf{z}^T\mathbf{\Xi}) \\
\operatorname{tr}(\mathbf{x}\mathbf{x}^T\mathbf{\Theta}\mathbf{\Sigma}\mathbf{\Theta}^T) &= \operatorname{tr}\Bigg(\mathbf{x}_s\mathbf{x}_s^T\left(\frac{1}{m}\mathbf{\Pi}\right)\left(\mathbf{\Omega}+\frac{1}{m}\mathbf{\Delta}\right)^{-1}\left(\frac{1}{m}\mathbf{\Pi}\right)^T \\
&\quad + \left(\sum_{i<j}^m \mathbf{x}_{ij}\mathbf{x}_{ij}^T\right)\frac{1}{m}\mathbf{\Pi}\left(\frac{1}{m}\mathbf{\Delta}\right)^{-1}\frac{1}{m}\mathbf{\Pi}^T \\
&\quad + \mathbf{z}\mathbf{z}^T\mathbf{\Xi}\left(\mathbf{\Omega}+\frac{1}{m}\mathbf{\Delta}\right)^{-1}\mathbf{\Xi}^T\Bigg) \\
\det(\mathbf{\Lambda}) &= \det(m\mathbf{\Omega}+\mathbf{\Delta})\det(\mathbf{\Delta})^{m-1}
\end{aligned}$$

The equation above on determinants follows from Theorem 3 in Silvester (2000). □

Proof of Theorem 2

We first show that the following lemma holds, which is then used to prove Theorem 2.

Lemma 1. *Let* $\mathbf{K} = \mathbf{C}^{+}\mathbf{C} = \mathbf{C}^{T}\mathbf{C}^{+T} = \mathbf{K}^{T}$. *Then, for* $\mathbf{\Sigma} = \mathbf{\Lambda}^{-1} = [\mathbf{I}_{m\times m} \otimes \mathbf{\Delta} + \mathbf{1}_{m\times m} \otimes \mathbf{\Omega}]^{-1}$, *we have* $\mathbf{K\Sigma} = \mathbf{\Sigma K}$.

Proof. Using a block inversion of a matrix, we can show $\mathbf{\Sigma} = \mathbf{I}_{m\times m} \otimes \mathbf{E} + \mathbf{1}_{m\times m} \otimes \mathbf{\Phi}$, where $\mathbf{\Phi} = -\mathbf{\Delta}^{-1}\mathbf{\Omega}(m\mathbf{\Omega} + \boldsymbol{\delta})^{-1}$ and $\mathbf{E} = \mathbf{\Delta}^{-1}$, and verify this by checking $\mathbf{\Sigma\Lambda} = \mathbf{I}$. $\mathbf{K}$ is an orthogonal projection matrix with $\mathbf{K}_{i,i} = \mathbf{K}_{i,2i} = \mathbf{K}_{2i,i} = \mathbf{K}_{2i,2i} = \frac{1}{2}$ for $i \in H$ for outputs with missing individual-level observations. For observed outputs $i \in V$, $\mathbf{K}_{i,i} = \mathbf{K}_{2i,2i} = 1$, and $\mathbf{K}_{i,2i} = \mathbf{K}_{2i,i} = 0$. Then, using the repeat block structure of $\mathbf{\Sigma}$, we define all the elements of $\mathbf{K\Sigma}$ and $\mathbf{\Sigma K}$:

$$(\mathbf{K\Sigma})_{i,i} = (\mathbf{K\Sigma})_{i,2i} = (\mathbf{K\Sigma})_{2i,i} = (\mathbf{K\Sigma})_{2i,2i} = \mathbf{\Phi}_{i,i} + \frac{1}{2}\boldsymbol{\epsilon}_i \quad \text{for } i \in H, \text{ and}$$
$$(\mathbf{K\Sigma})_{i,i} = (\mathbf{K\Sigma})_{i,2i} = (\mathbf{K\Sigma})_{2i,i} = (\mathbf{K\Sigma})_{2i,2i} = \mathbf{\Phi}_{i,i} + \boldsymbol{\epsilon}_i \quad \text{for } i \in V.$$

For $(i,j) \in \{i,j : i \neq j, i = 1,\ldots,q, j = 1,\ldots,q\}$, $(\mathbf{K\Sigma})_{i,j} = (\mathbf{K\Sigma})_{i,2j} = (\mathbf{K\Sigma})_{2i,j} = (\mathbf{K\Sigma})_{2i,2j} = \mathbf{\Phi}_{i,j}$. The elements of $\mathbf{\Sigma K}$ are identical to those of $\mathbf{K\Sigma}$. □

Now we prove Theorem 2.

Proof. From our model in Eq. (2), we derive the probability distribution for the observed outputs. We begin by re-writing Eq. (2) in the form of Gaussian distribution to make the marginal distribution explicitly represented:

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) \sim \mathcal{N}(-\mathbf{C}_a\mathbf{\Lambda}^{-1}\mathbf{\Theta}^{T}\mathbf{x}, \mathbf{C}_a\mathbf{\Lambda}^{-1}\mathbf{C}_a), \tag{8}$$

where $\mathbf{C}_a = \begin{bmatrix} \mathbf{I}_{q\times q} & \mathbf{0}_{q\times q} \\ \mathbf{0}_{q\times q} & \mathbf{I}_{q\times q} \end{bmatrix}$. Then, the output sum variables are also Gaussian distributed:

$$p(\mathbf{y}_s \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) \sim \mathcal{N}(-\mathbf{C}_s\mathbf{\Lambda}^{-1}\mathbf{\Theta}^{T}\mathbf{x}, \mathbf{C}_s\mathbf{\Lambda}^{-1}\mathbf{C}_s^{T}), \tag{9}$$

where $\mathbf{C}_s = [\mathbf{I}_{q\times q}\ \mathbf{I}_{q\times q}]$.

We combine the marginal distributions for the observed output variables in V from Eq. (8) and for output variables in H with missing individual-level observations but only with group-level sum data from Eq. (9) to form a joint distribution of the observed variables $\mathbf{y}_s^H$ and $\mathbf{y}^V$:

$$p(\mathbf{y}_s^H, \mathbf{y}^V \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) \sim \mathcal{N}(-\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^{T}\mathbf{x}, \mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{C}^{T}), \tag{10}$$

given $\mathbf{C} = \begin{bmatrix} \mathbf{C}_s^H \\ \mathbf{C}_a^V \end{bmatrix}$. $\mathbf{C}_s^H$ and $\mathbf{C}_a^V$ are the submatrices of $\mathbf{C}_s$ and $\mathbf{C}_a$ with only the rows corresponding to the variables in $\mathbf{y}_s^H$ and $\mathbf{y}^V$. To show the distribution in Eq. (10) is a CGGM, we only need to show Eq. (10) can be written as

$$p(\mathbf{y}_s^H, \mathbf{y}^V \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) = \mathcal{N}\big(-\tilde{\mathbf{\Lambda}}^{-1}\tilde{\mathbf{\Theta}}^{T}\mathbf{x}, \tilde{\mathbf{\Lambda}}^{-1}\big), \tag{11}$$

by obtaining the explicit forms of the mean and inverse covariance of the above, since expanding the quadratic term in this distribution above leads to the CGGM in Theorem 2.

We first show $\tilde{\mathbf{\Lambda}} = [\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{C}^{T}]^{-1} = \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^{+}$, where $\mathbf{C}^{+}$ is the Moore-Penrose inverse of $\mathbf{C}$, by showing $\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{C}^{T}\mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^{+} = \mathbf{I}$:

$$\begin{aligned}
\mathbf{C\Sigma C}^{T}\mathbf{C}^{+T}\mathbf{\Lambda C}^{+} &= \mathbf{C}\mathbf{C}^{T}\mathbf{C}^{+T}\mathbf{\Lambda}^{-1}\mathbf{\Lambda}\mathbf{C}^{+} && \text{from Lemma 1} \\
&= \mathbf{C}\mathbf{C}^{T}\mathbf{C}^{+T}\mathbf{C}^{+} && \\
&= \mathbf{C}\mathbf{C}^{+}\mathbf{C}\mathbf{C}^{+} && \text{since } \mathbf{C}^{+}\mathbf{C} \text{ is symmetric from the definition of Moore-Penrose inverse} \\
&= \mathbf{C}\mathbf{C}^{+} && \text{from } \mathbf{C}\mathbf{C}^{+} = \mathbf{I} \\
&= \mathbf{I} &&
\end{aligned}$$

Then the mean of Eq. (10) can be written as

$$\begin{aligned}
-\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x} &= -\mathbf{C}\mathbf{C}^+\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x} && \text{from } \mathbf{C} = \mathbf{C}\mathbf{C}^+\mathbf{C} \\
&= -\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{C}^+\mathbf{C}\mathbf{\Theta}^T\mathbf{x} && \text{from Lemma 1} \\
&= -\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{C}^T\mathbf{C}^{+T}\mathbf{\Theta}^T\mathbf{x} && \text{from } \mathbf{C}^+\mathbf{C} = \mathbf{C}^T\mathbf{C}^{+T} \text{ since } \mathbf{C}^+\mathbf{C} \text{ is symmetric} \\
&= -\tilde{\mathbf{\Lambda}}^{-1}\tilde{\mathbf{\Theta}}^T\mathbf{x} && \tilde{\mathbf{\Theta}} = \mathbf{\Theta}\mathbf{C}^+
\end{aligned}$$

Thus, we have Eq. (11) with $\tilde{\mathbf{\Theta}} = \mathbf{\Theta}\mathbf{C}^+$ and $\tilde{\mathbf{\Lambda}} = \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^+$.

□

The corollary below follows from the theorem above.

Corollary 2. *Learning the collapsed multi-level CGGM in Eq. (3) with randomly missing individual-level output data is equivalent to learning the full multi-level model with imputed data* $\tilde{\mathbf{y}} = \mathbf{C}^+ \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}$.

Proof. From Theorem 2, given $\tilde{\mathbf{\Lambda}} = \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^+$ and $\tilde{\mathbf{\Theta}}^T = \mathbf{C}^{+T}\mathbf{\Theta}^T$, we can re-write the log probability as

$$\begin{aligned}
\log p(\mathbf{y}_s^H, \mathbf{y}^V \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) &= -\frac{1}{2}\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}^V \end{bmatrix}^T \tilde{\mathbf{\Lambda}} \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}^V \end{bmatrix} - \mathbf{x}^T\tilde{\mathbf{\Theta}}\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix} - \log Z(\mathbf{x}) \\
&= -\frac{1}{2}\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}^T \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^+ \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix} - \mathbf{x}^T\mathbf{\Theta}\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix} - \log Z(\mathbf{x}) \\
&= -\frac{1}{2}\left[\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}\right]^T \mathbf{\Lambda}\left[\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}\right] - \mathbf{x}^T\mathbf{\Theta}\left[\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}\right] - \log Z(\mathbf{x}) \\
&= -\frac{1}{2}\tilde{\mathbf{y}}^T\mathbf{\Lambda}\tilde{\mathbf{y}} - \mathbf{x}^T\mathbf{\Theta}\tilde{\mathbf{y}} - \log Z(\mathbf{x}).
\end{aligned}$$

□

Supplementary Information B: Details of the Optimization Method for Multi-level Conditional Gaussian Graphical Models

Alternating Newton coordinate descent

We provide the details of the learning algorithm for our multi-level model, using the representation of the model in Theorem 1. The L_1-regularized negative log-likelihood is given as

$$\begin{aligned}
\underset{\mathbf{\Omega}\succ 0, \mathbf{\Delta}, \mathbf{\Xi}, \mathbf{\Pi}}{\operatorname{argmin}} \; & -\log|\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}| + \operatorname{tr}\Big(\mathbf{S}_{\mathbf{y}_s}(\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}) + 2\mathbf{S}_{\mathbf{y}\mathbf{x}_s}\frac{1}{m}\mathbf{\Pi} + 2\mathbf{S}_{\mathbf{y}\mathbf{z}}\mathbf{\Xi} \\
& + (\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta})^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}\mathbf{z}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{x}\mathbf{z}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\Big) \\
& - (m-1)\log|\mathbf{\Delta}| + \operatorname{tr}\Big(\mathbf{S}_{\mathbf{y}_d}\frac{1}{m}\mathbf{\Delta} + 2\mathbf{S}_{\mathbf{y}\mathbf{x}_d}\frac{1}{m}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\Big) \\
& + \lambda_{\mathbf{\Omega}}\|\mathbf{\Omega}\|_1 + \lambda_{\mathbf{\Delta}}\|\mathbf{\Delta}\|_1 + \lambda_{\mathbf{\Pi}}\|\mathbf{\Pi}\|_1 + \lambda_{\mathbf{\Xi}}\|\mathbf{\Xi}\|_1\,,
\end{aligned} \tag{12}$$

using components of covariances $\mathbf{S}_{\mathbf{y}\mathbf{w}_s} = [\mathbf{S}_{\mathbf{y}\mathbf{x}_s}\mathbf{S}_{\mathbf{y}\mathbf{z}}]$ and $\mathbf{S}_{\mathbf{w}} = \begin{bmatrix} \mathbf{S}_{\mathbf{x}_s} & \mathbf{S}_{\mathbf{x}\mathbf{z}_s} \\ \mathbf{S}_{\mathbf{x}\mathbf{z}_s}^T & \mathbf{S}_{\mathbf{z}} \end{bmatrix}$.

In each iteration, we alternately optimize for each of the parameters $\mathbf{\Omega}$, $\mathbf{\Delta}$, $\mathbf{\Xi}$, and $\mathbf{\Pi}$ by solving the optimization problem above for each of the parameter given the previous estimates of all the other parameters. We maintain and update $\mathbf{V} = \mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}$ after each update of $\mathbf{\Omega}$ and $\mathbf{\Delta}$ in order to avoid repeatedly computing $\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}$. In the rest of this section, we write $\mathbf{V} = \mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}$ whenever this term occurs.

Algorithm 1: Alternating Newton Coordinate Descent for Multi-level Conditional Gaussian Graphical Models

Input: $\mathbf{S}_{\mathbf{y}_d}, \mathbf{S}_{\mathbf{yx}_d}, \mathbf{S}_{\mathbf{x}_d}, \mathbf{Y}_s, \mathbf{X}_s, \mathbf{X}_d, \mathbf{Z}$, regularization parameters $\lambda_{\mathbf{\Omega}}, \lambda_{\mathbf{\Pi}}, \lambda_{\mathbf{\Xi}}$, and α
Initialize: $\mathbf{\Pi}, \mathbf{\Xi} \leftarrow 0, \mathbf{\Omega}, \mathbf{\Delta} \leftarrow \mathbf{I}_{q\times q}$
for $t = 0, 1, \ldots$ **do**
  Determine active sets $\mathcal{S}_{\mathbf{\Omega}}, \mathcal{S}_{\mathbf{\Delta}}, \mathcal{S}_{\mathbf{\Pi}}, \mathcal{S}_{\mathbf{\Xi}}$
  Find Newton directions via coordinate descent
    $D_{\mathbf{\Omega}} = \underset{\delta_{\mathbf{\Omega}}}{\operatorname{argmin}}\, \bar{g}(\mathbf{\Omega} + \delta_{\mathbf{\Omega}}, \mathbf{\Delta}, \mathbf{\Pi}, \mathbf{\Xi}) + h(\mathbf{\Omega} + \delta_{\mathbf{\Omega}}, \mathbf{\Delta}, \mathbf{\Pi}, \mathbf{\Xi})$
  Update $\mathbf{\Omega}^+ = \mathbf{\Omega} + \alpha D_{\mathbf{\Omega}}$, where α is found via line search
    $D_{\mathbf{\Delta}} = \underset{\delta_{\mathbf{\Delta}}}{\operatorname{argmin}}\, \bar{g}(\mathbf{\Omega}, \mathbf{\Delta} + \delta_{\mathbf{\Delta}}, \mathbf{\Pi}, \mathbf{\Xi}) + h(\mathbf{\Omega}, \mathbf{\Delta} + \delta_{\mathbf{\Delta}}, \mathbf{\Pi}, \mathbf{\Xi})$
  Update $\mathbf{\Delta}^+ = \mathbf{\Delta} + \alpha D_{\mathbf{\Delta}}$, where α is found via line search
  Solve via coordinate descent:
    $\mathbf{\Pi}^+ = \underset{\mathbf{\Pi}}{\operatorname{argmin}}\, g(\mathbf{\Pi}) + h(\mathbf{\Pi})$
    $\mathbf{\Xi}^+ = \underset{\mathbf{\Xi}}{\operatorname{argmin}}\, g(\mathbf{\Xi}) + h(\mathbf{\Xi})$

Update for $\mathbf{\Xi}$: Optimizing for $\mathbf{\Xi}$ given all the other parameters corresponds to solving the following optimization problem:

$$\underset{\mathbf{\Xi}}{\operatorname{argmin}}\, g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Pi}}(\mathbf{\Xi}) + \lambda_{\mathbf{\Xi}}\|\mathbf{\Xi}\|_1\,,$$

where

$$g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Pi}}(\mathbf{\Xi}) = \operatorname{tr}\Big(2\mathbf{S}_{\mathbf{yz}}\mathbf{\Xi} + \mathbf{V}^{-1}\big(\frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\Big).$$

Update for $\mathbf{\Pi}$: Optimizing for $\mathbf{\Pi}$ given all the other parameters corresponds to solving the following optimization problem:

$$\underset{\mathbf{\Pi}}{\operatorname{argmin}}\, g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Xi}}(\mathbf{\Pi}) + \lambda_{\mathbf{\Pi}}\|\mathbf{\Pi}\|_1\,,$$

where

$$g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Xi}}(\mathbf{\Pi}) = \operatorname{tr}\Big(2\mathbf{S}_{\mathbf{yx}_s}\frac{1}{m}\mathbf{\Pi} + 2\mathbf{S}_{\mathbf{yx}_d}\frac{1}{m}\mathbf{\Pi} + \mathbf{V}^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi}\big) + \frac{1}{m}\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\Big).$$

The two problems above for $\mathbf{\Xi}$ and $\mathbf{\Pi}$ are Lasso, which can be solved efficiently using a coordiante descent algorithm.

Update for $\mathbf{\Omega}$: To optimize Eq. (12) for $\mathbf{\Omega}$ given all the other parameters, we use the Newton's method. We first find the Newton descent direction by minimizing the second-order approximation of the objective in Eq. (12) with respect to $\mathbf{\Omega}$ and then update $\mathbf{\Omega}$ using this Newton direction and step size found by line search. The Newton direction $D_{\mathbf{\Omega}}$ is found by solving the following optimization problem:

$$D_{\mathbf{\Omega}} = \underset{\delta_{\mathbf{\Omega}}}{\operatorname{argmin}}\, \bar{g}_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\delta_{\mathbf{\Omega}}) + \lambda_{\mathbf{\Omega}}\|\mathbf{\Omega} + \delta_{\mathbf{\Omega}}\|_1\,,$$

where $\bar{g}_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\delta_{\mathbf{\Omega}})$ is the second-order Taylor expansion of the data log-likelihood in Eq. (12) with respect to $\mathbf{\Omega}$ and is given as

$$\bar{g}_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\delta_{\mathbf{\Omega}}) = \operatorname{vec}(\nabla_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega}))^T\operatorname{vec}(\delta_{\mathbf{\Omega}}) + \frac{1}{2}\operatorname{vec}(\delta_{\mathbf{\Omega}})^T\nabla^2_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega})\operatorname{vec}(\delta_{\mathbf{\Omega}}),$$

with the gradient and Hessian

$$\nabla_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega}) = \mathbf{S}_{\mathbf{y}_s} - \mathbf{V}^{-1} - \mathbf{V}^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\mathbf{V}^{-1}$$

$$\nabla^2_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega}) = \mathbf{V}^{-1} \otimes \left(\mathbf{V}^{-1} + 2\mathbf{V}^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\mathbf{V}^{-1}\right).$$

The above optimization problem is a Lasso problem and can be solved using a coordinate descent method.

Update for $\boldsymbol{\Delta}$: To optimize Eq. (12) for $\boldsymbol{\Delta}$ given all the other parameters, we again use the Newton's method. We find the Newton descent direction $D_{\boldsymbol{\Delta}}$ by minimizing a second-order approximation of the objective in Eq. (12) with respect to $\boldsymbol{\Delta}$ and update $\boldsymbol{\Delta}$ with this Newton direction and a step size found by line search. The Newton descent direction $D_{\boldsymbol{\Delta}}$ can be found by solving the following:

$$D_{\boldsymbol{\Delta}} = \underset{\delta_{\boldsymbol{\Delta}}}{\operatorname{argmin}}\, \bar{g}_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\delta_{\boldsymbol{\Delta}}),$$

where

$$\bar{g}_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\delta_{\boldsymbol{\Delta}}) = \operatorname{vec}(\nabla_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}))^T \operatorname{vec}(\delta_{\boldsymbol{\Delta}}) + \frac{1}{2}\operatorname{vec}(\delta_{\boldsymbol{\Delta}})^T \nabla^2_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}) \operatorname{vec}(\delta_{\boldsymbol{\Delta}}),$$

with the gradient and Hessian

$$\begin{aligned}
\nabla_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}) = \frac{1}{m}\Big(&\mathbf{S}_{\mathbf{y}_s} + \mathbf{S}_{\mathbf{y}_d} - \mathbf{V}^{-1} - (m-1)\boldsymbol{\Gamma}^{-1} \\
&- \mathbf{V}^{-1}\big(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\big)\mathbf{V}^{-1} \\
&- \frac{1}{m^2}\boldsymbol{\Delta}^{-1}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\boldsymbol{\Pi}\boldsymbol{\Delta}^{-1}\Big) \\
\nabla^2_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}) = \frac{1}{m^2}\Bigg[&\mathbf{V}^{-1} \otimes \big(\mathbf{V}^{-1} + 2\mathbf{V}^{-1}\big(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\big)\mathbf{V}^{-1}\big) \\
&+ \boldsymbol{\Gamma}^{-1} \otimes \left((m-1)\boldsymbol{\Gamma}^{-1} + \frac{2}{m^2}\boldsymbol{\Delta}^{-1}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\boldsymbol{\Pi}\boldsymbol{\Delta}^{-1}\right)\Bigg].
\end{aligned}$$

This optimization problem for finding a Newton direction is again a Lasso problem, which can be solved using a coordinate descent algorithm.

In order to improve the efficiency of the coordinate descent algorithm, we extend the strategies used in McCarter and Kim (2016). First, we restrict the coordinate descent updates to the active set of variables given as

$$\begin{aligned}
\mathcal{S}_{\boldsymbol{\Omega}} &= \{(\delta_{\boldsymbol{\Omega}})_{ij} : |\left(\nabla_{\boldsymbol{\Omega}} g(\boldsymbol{\Omega})\right)_{ij}| > \lambda_{\boldsymbol{\Omega}} \vee \boldsymbol{\Omega}_{ij} \neq 0\} \\
\mathcal{S}_{\boldsymbol{\Pi}} &= \{\boldsymbol{\Pi}_{ij} : |\left(\nabla_{\boldsymbol{\Pi}} g(\boldsymbol{\Pi})\right)_{ij}| > \lambda_{\boldsymbol{\Pi}} \vee \boldsymbol{\Pi}_{ij} \neq 0\} \\
\mathcal{S}_{\boldsymbol{\Xi}} &= \{\boldsymbol{\Xi}_{ij} : |\left(\nabla_{\boldsymbol{\Xi}} g(\boldsymbol{\Xi})\right)_{ij}| > \lambda_{\boldsymbol{\Xi}} \vee \boldsymbol{\Xi}_{ij} \neq 0\} \\
\mathcal{S}_{\boldsymbol{\Delta}} &= \{(\delta_{\boldsymbol{\Delta}})_{ij} : |\left(\nabla_{\boldsymbol{\Delta}} g(\boldsymbol{\Delta})\right)_{ij}| > \lambda_{\boldsymbol{\Delta}} \vee \boldsymbol{\Delta}_{ij} \neq 0\}.
\end{aligned}$$

Second, we compute and store the intermediate results at the beginning of the coordinate descent updates for $\boldsymbol{\Xi}$ and $\boldsymbol{\Pi}$ and for $\delta_{\boldsymbol{\Omega}}$ and $\delta_{\boldsymbol{\Delta}}$ in each Newton iteration. We store the intermediate results $U_{\boldsymbol{\Xi}} \coloneqq \boldsymbol{\Xi}\mathbf{V}^{-1}$ for $\boldsymbol{\Xi}$, $U_{\boldsymbol{\Pi}_1} \coloneqq \boldsymbol{\Pi}\mathbf{V}^{-1}$ and $U_{\boldsymbol{\Pi}_2} \coloneqq \boldsymbol{\Pi}\boldsymbol{\Delta}^{-1}$ for $\boldsymbol{\Pi}$, $U_{\boldsymbol{\Omega}} \coloneqq \delta_{\boldsymbol{\Omega}}\mathbf{V}^{-1}$ for $\delta_{\boldsymbol{\Omega}}$, and $U_{\boldsymbol{\delta}_1} \coloneqq \delta_{\boldsymbol{\Delta}}\mathbf{V}^{-1}$ and $U_{\boldsymbol{\delta}_2} \coloneqq \delta_{\boldsymbol{\Delta}}\boldsymbol{\Delta}^{-1}$ for $\delta_{\boldsymbol{\Delta}}$. When $\boldsymbol{\Xi}_{ij}$ and $\boldsymbol{\Pi}_{ij}$ are updated, the ith row of the corresponding intermediate results is updated. After $(\delta_{\boldsymbol{\Omega}})_{ij}$ update, the ith and jth rows of $U_{\boldsymbol{\Omega}}$ are updated and after $(\delta_{\boldsymbol{\Delta}})_{ij}$ update, the ith and jth rows of $U_{\boldsymbol{\Delta}}$ is updated.

Alternating Newton block coordinate descent

We further adopt the block-wise optimization method in McCarter and Kim (2016) to remove the memory requirement for storing large dense matrices during coordinate descent optimization. The coordinate descent updates require precomputing and storing the large dense matrices, such as $\mathbf{S}_{\mathbf{x}_s}$, $\mathbf{S}_{\mathbf{x}_d}$, and $\mathbf{V}^{-1}$. Instead, we perform a block-wise update of $\boldsymbol{\Xi}$, $\boldsymbol{\Pi}$, $\delta_{\boldsymbol{\Omega}}$, and $\delta_{\boldsymbol{\Delta}}$, precomputing and reusing the portions of the large matrices that are required for updating the current block of the parameters.

Blockwise optimization for $\boldsymbol{\Omega}$: A coordinate-descent optimization for $(\delta_{\boldsymbol{\Omega}})_{ij}$ requires computing the ith and jth columns of large dense $q \times q$ matrices $\mathbf{V}^{-1}$ and $\mathbf{V}^{-1}\left(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\right)\mathbf{V}^{-1}$ found in the gradient and Hessian of the objective with respect to $\boldsymbol{\Omega}$. We represent $\mathbf{V}^{-1}\left(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\right)\mathbf{V}^{-1}$ as $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)$, where

$R_1 = Q_1\mathbf{V}^{-1}$, $R_2 = Q_2\mathbf{V}^{-1}$, and $R_3 = Q_3\mathbf{V}^{-1}$. We precompute and store $n \times q$ matrices, $Q_1 = \frac{1}{m}\mathbf{X}_s\mathbf{\Pi}$, $Q_2 = \mathbf{X}_s\mathbf{\Xi}$, and $Q_3 = \mathbf{Z}\mathbf{\Xi}$, which are used to update R_1, R_2, and R_3 after each update of $\mathbf{V}$. Instead of storing $\mathbf{V}^{-1}$ and $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)$ persistently, we perform blockwise coordinate descent, whereby the active set $\mathcal{S}_{\mathbf{\Omega}}$ of $\delta_{\mathbf{\Omega}}$ is clustered into smaller blocks, and we perform coordinate-descent updates for elements in the active set in each block. We cluster the rows and columns of $\delta_{\mathbf{\Omega}}$ by partitioning $\{1, \ldots, q\}$ into $k_{\mathbf{\Omega}}$ sets, $C_1, \ldots, C_{k_{\mathbf{\Omega}}}$. For each (C_r, C_z) block of $\delta_{\mathbf{\Omega}}$, we do blockwise computation for $(\mathbf{V}^{-1})_{(:,C_r)}$ and $(\mathbf{V}^{-1})_{(:,C_z)}$, and compute the ith column in each block as $\mathbf{V}(\mathbf{V}^{-1})_{(:,i)} = \mathbf{e}_i$ using conjugate gradient method, where $\mathbf{e}_i$ is a vector of q 0's except for 1 in the ith element. Similarly for $[(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)]_{(:,C_r)}$, and $[(R_1 + R_2)^T(R_1 + R_2)]_{(:,C_z)}$, we obtain the ith column in each block using $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)_{(:,i)}$. After updating each $(\delta_{\mathbf{\Omega}})_{ij}$, we update the corresponding ith and jth rows of the intermediate results $(U_{\mathbf{\Omega}})_{(:,C_z)}$ and $(U_{\mathbf{\Omega}})_{(:,C_r)}$.

Blockwise optimization for $\mathbf{\Delta}$: A coordinate descent optimization for $(\delta_{\mathbf{\Delta}})_{ij}$ requires computing the ith and jth columns of large dense $q \times q$ matrices $\mathbf{V}^{-1}$, $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)$, $\mathbf{\Delta}^{-1}$, and $\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\mathbf{\Delta}^{-1}$ found in the gradient and Hessian of the objective with respect to $\mathbf{\Delta}$. The optimization method for $\mathbf{\Delta}$ is similar to the one for $\mathbf{\Omega}$. We represent $\frac{1}{m^2}\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\mathbf{\Delta}^{-1}$ as $R_4^T R_4$, where $R_4 = Q_4\mathbf{\Delta}^{-1}$. We precompute and store a $n \times q$ matrix, $Q_4 = \frac{1}{m}\mathbf{X}_d\mathbf{\Pi}$, which is used to update R_4 after each update of $\mathbf{\Delta}$.

Blockwise optimization for $\mathbf{\Xi}$: The coordinate descent update for $(\mathbf{\Xi})_{ij}$ requires the ith column of $\mathbf{S}_{\mathbf{x}_s}$ and $\mathbf{S}_{\mathbf{z}}$ and the jth column of $\mathbf{V}^{-1}$. Instead of storing $\mathbf{S}_{\mathbf{x}_s}$, $\mathbf{S}_{\mathbf{z}}$, and $\mathbf{V}$ persistently, we perform blockwise coordinate descent. We cluster the columns of $\mathbf{\Xi}$ by partitioning $\{1, \ldots, q\}$ into $k_{\mathbf{\Xi}}$ sets, $C_1, \ldots, C_{k_{\mathbf{\Xi}}}$. For each block (i, C_r), where $i \in \{1, \ldots, p + r\}$, we compute $(\mathbf{S}_{\mathbf{x}_s})_{(:,i)}$, $(\mathbf{S}_{\mathbf{z}})_{(:,i)}$, and $(\mathbf{V}^{-1})_{(:,C_r)}$. After updating each $(\mathbf{\Xi})_{ij}$, we update the corresponding ith row of the intermediate result $(U_{\mathbf{\Xi}})_{(:,C_r)}$.

Blockwise optimization for $\mathbf{\Pi}$: The coordinate descent update for $(\mathbf{\Pi})_{ij}$ requires the ith column of $\mathbf{S}_{\mathbf{x}_s}$ and $\mathbf{S}_{\mathbf{x}_d}$ and the jth column of $\mathbf{V}^{-1}$ and $\mathbf{\Delta}^{-1}$. Instead of storing $\mathbf{S}_{\mathbf{x}_s}$, $\mathbf{S}_{\mathbf{x}_d}$, $\mathbf{V}$, and $\mathbf{\Delta}$ persistently, we perform blockwise coordinate descent. We cluster the columns of $\mathbf{\Pi}$ by partitioning $\{1, \ldots, q\}$ into $k_{\mathbf{\Pi}}$ sets, $C_1, \ldots, C_{k_{\mathbf{\Pi}}}$. For each block (i, C_r), where $i \in \{1, \ldots, p\}$, we compute $(\mathbf{S}_{\mathbf{x}_s})_{(:,i)}$, $(\mathbf{S}_{\mathbf{x}_d})_{(:,i)}$, $(\mathbf{V}^{-1})_{(:,C_r)}$, and $(\mathbf{\Delta}^{-1})_{(:,C_r)}$. After updating each $(\mathbf{\Pi})_{ij}$, we update the corresponding ith row of the intermediate results $(U_{\mathbf{\Pi}_1})_{(:,C_r)}$ and $(U_{\mathbf{\Pi}_2})_{(:,C_r)}$.

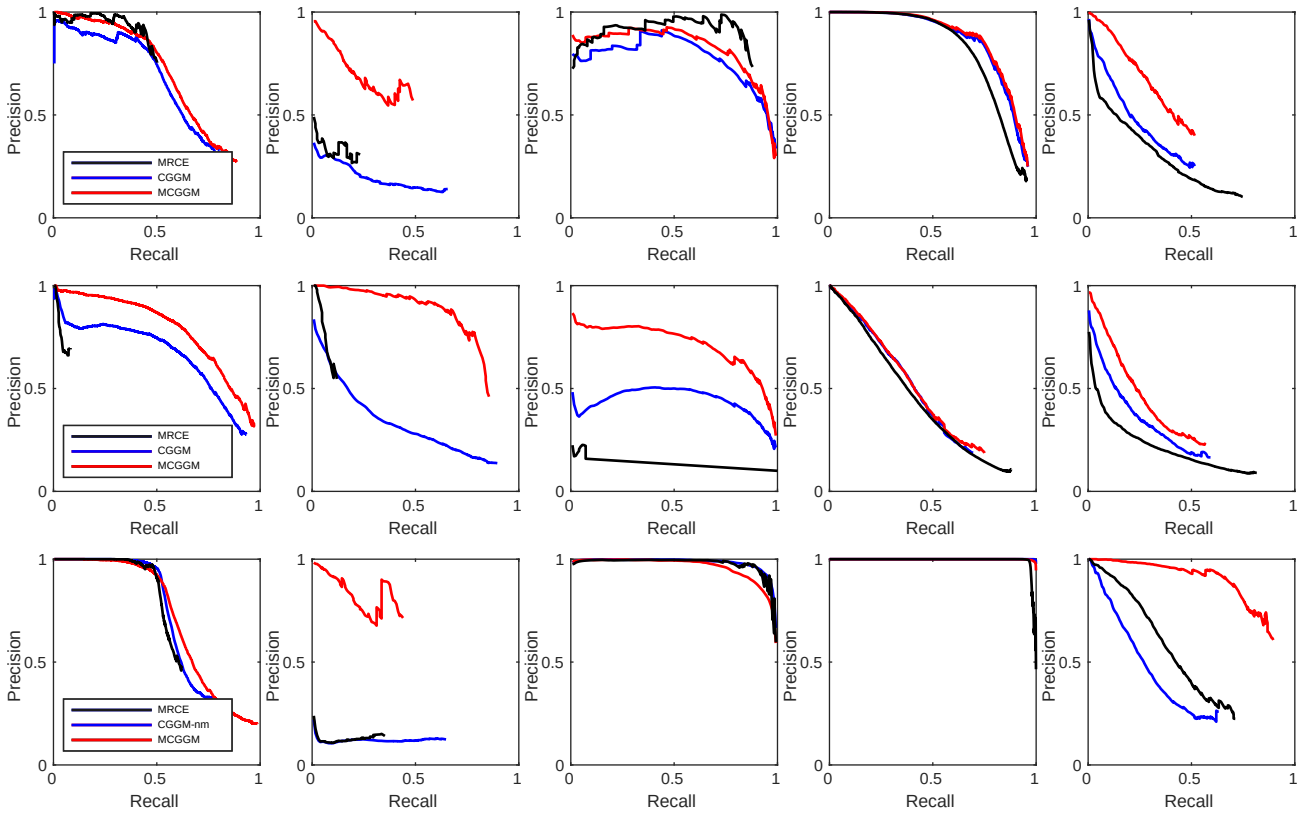


Figure 4: Comparison of methods on simulated data. Precision-recall curves for the recovery of (a) $\mathbf{\Omega} + \mathbf{\Delta}$, (b) $\mathbf{\Omega}$, and (c) $\mathbf{\Delta}$, (d) $\mathbf{\Pi}$, and (e) $\mathbf{\Xi}$. Top row for Case 1, middle row for Case 2, and bottom row for Case 3 with group size $m = 6$. $\mathbf{\Omega}$ is set as a clustered network.

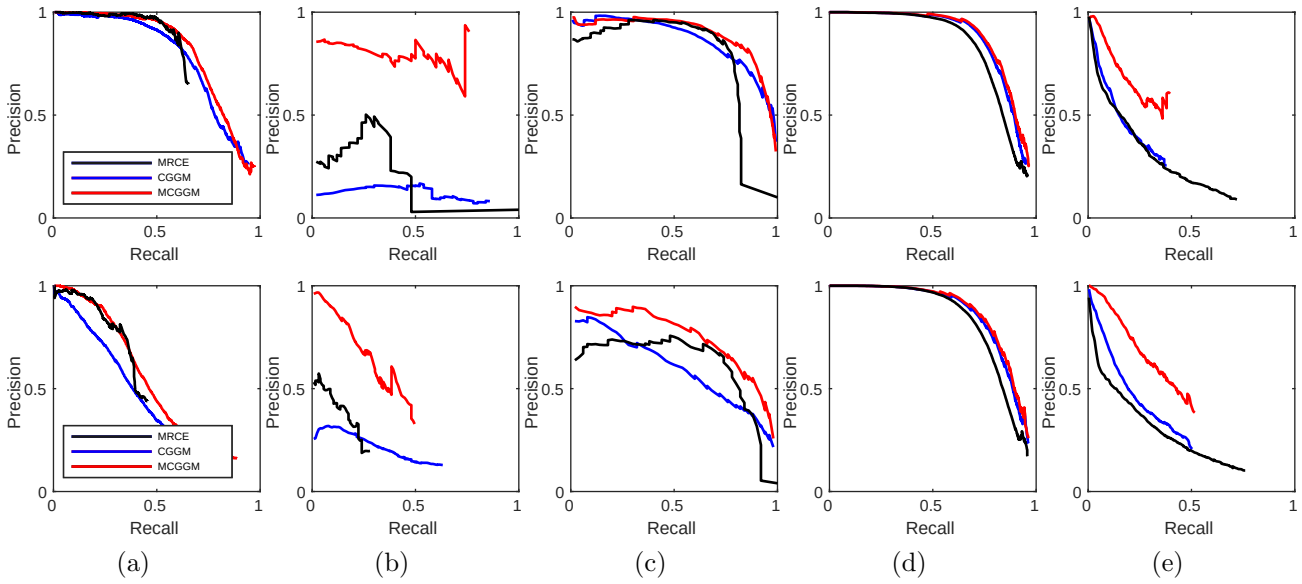


Figure 5: Comparison of methods on simulated data. Precision-recall curves for the recovery of (a) $\mathbf{\Omega} + \mathbf{\Delta}$, (b) $\mathbf{\Omega}$, and (c) $\mathbf{\Delta}$, (d) $\mathbf{\Pi}$, and (e) $\mathbf{\Xi}$. The results from Case 4 (top) and Case 5 (bottom) are shown for group size $m = 6$ and for scale-free networks for $\mathbf{\Omega}$ and $\mathbf{\Delta}$.

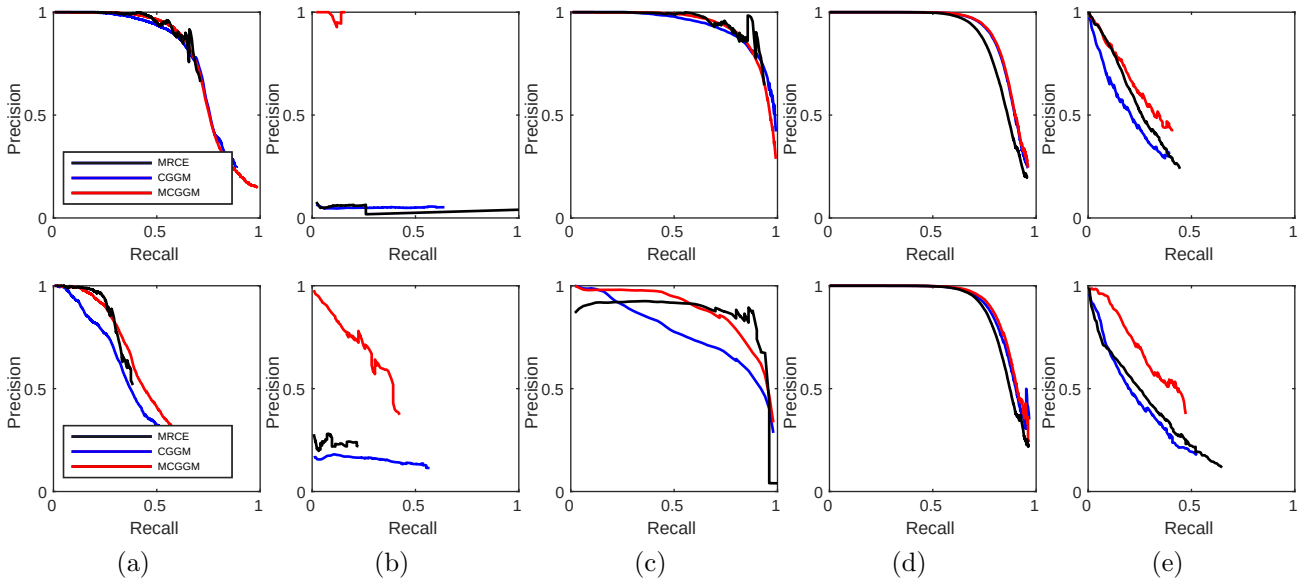


Figure 6: Comparison of methods on simulated data. Precision-recall curves for the recovery of (a) $\mathbf{\Omega} + \mathbf{\Delta}$, (b) $\mathbf{\Omega}$, and (c) $\mathbf{\Delta}$, (d) $\mathbf{\Pi}$, and (e) $\mathbf{\Xi}$. Results from Case 4 (top) and Case 5 (bottom) are shown for group size $m = 6$ and for $\mathbf{\Omega}$ with a clustered network.

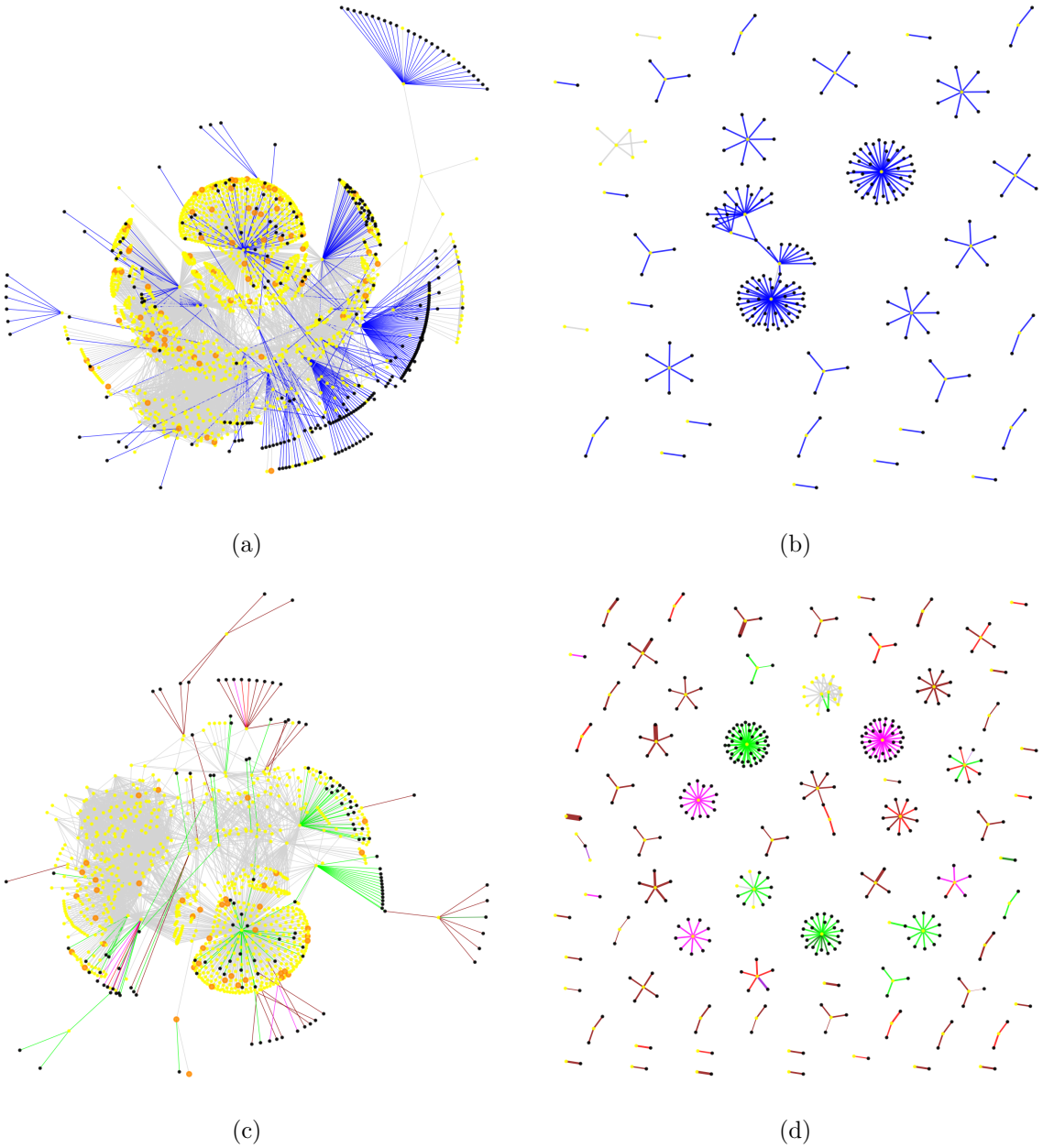


Figure 7: Single-level and multi-level conditional Gaussian graphical models estimated from GTEx data. The estimated single-level model is shown for (a) the largest connected component and (b) the rest of the gene networks, where gene network edges are shown as gray and the effects of variants on gene expressions are shown as blue. The estimated multi-level model is shown for (c) the largest connected component and (d) the rest of the gene networks. In the multi-level model, the influence of *cis*-acting variants is shown as brown if they have been previously annotated as having impact on gene expression, red if they have not been annotated, and pink if the distance to the gene it is influencing is greater than 1 megabase. The influence of *trans*-acting edges is shown as green. Known transcription factors are colored as big orange dots.