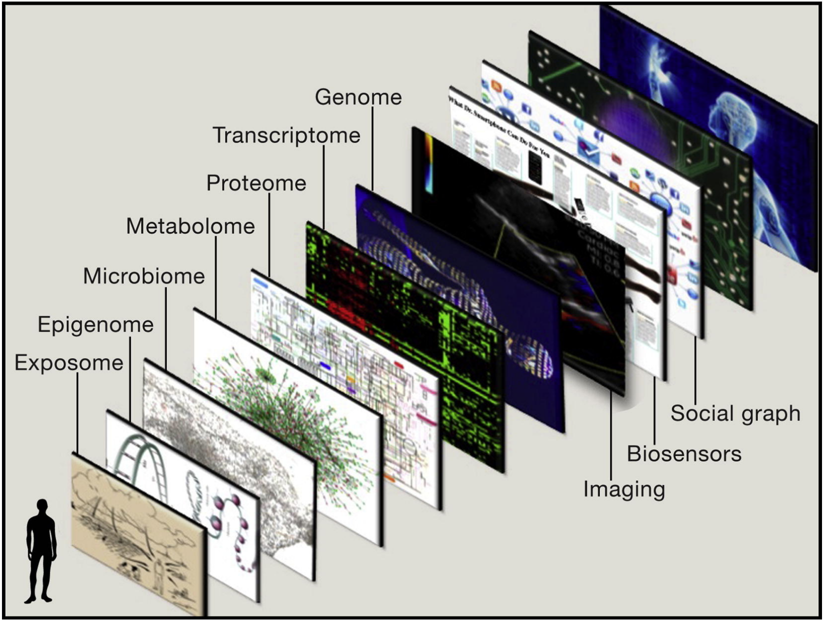
Statistical Inference with Multi-layered Graphical Models

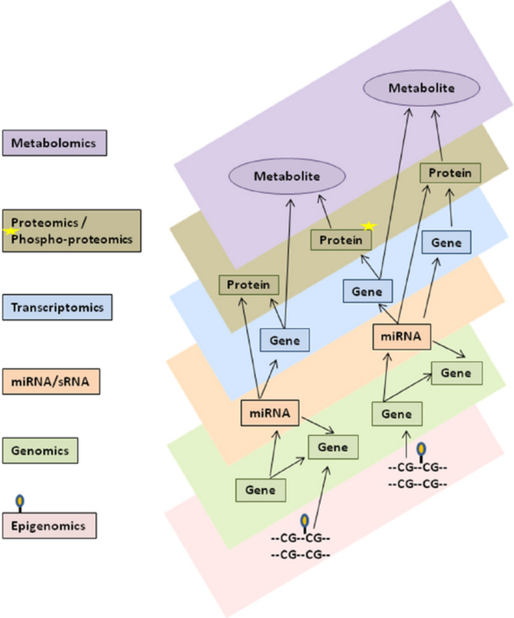
Subhabrata Majumdar
Joint work with George Michailidis
University of Florida Informatics Institute

Savvysherpa, Inc., Minneapolis, MN
May 3, 2018

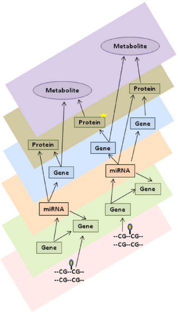
Summary of the talk

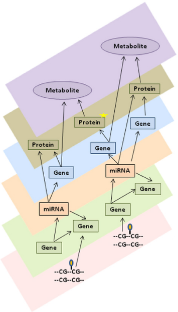
- Estimation of graphs from high-dimensional data is of importance for *biological processes, financial systems or social interactions*;
- Nodes in such data can have a *natural hierarchical structure*, e.g. Genes affecting proteins affecting metabolites, or macroeconomic indicators like interest rates or price indices affecting stock prices;
- There are *within layer and between-layer connections* in such structures.

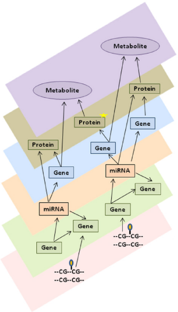

Genome
Transcriptome
Proteome
Metabolome
Microbiome
Epigenome
Exposome
Social graph
Biosensors
Imaging


Metabolomics
Proteomics / Phospho-proteomics
Transcriptomics
miRNA/sRNA
Genomics
Epigenomics
Metabolite
Metabolite
Protein
Protein
Protein
Gene
Gene
Gene
miRNA
miRNA
Gene
Gene
Gene
Gene
--CG--CG--
--CG--CG--
--CG--CG--
--CG--CG--

These connections can be different inside different organs, experimental conditions, or for different subtypes of the same disease;



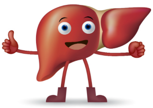




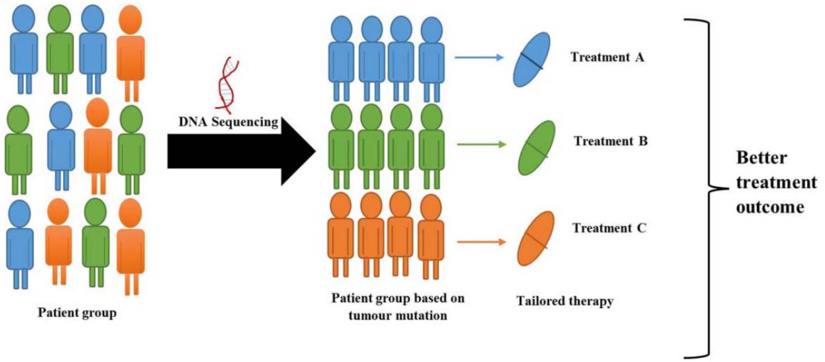
Liver

Kidney

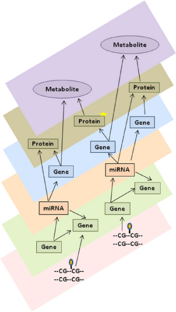
Lungs

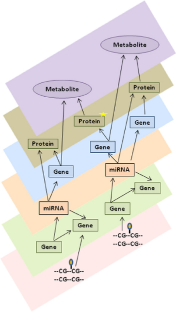


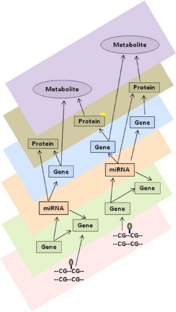
Connection to precision medicine



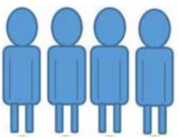
The connections between layers can be different for different patient profiles.







Treatment 1



Treatment 2

Treatment 3

Statistical inference for hierarchical graphical models.

In this work we propose a general statistical framework based on graphical models for *horizontal* (i.e. across conditions or subtypes) *and vertical* (i.e. across different layers containing data on molecular compartments) *integration of information* in data from such complex biological structures.

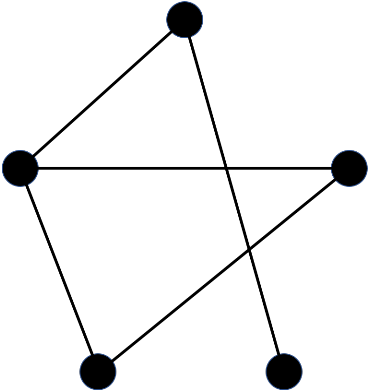
Specifically, we perform *joint estimation and hypothesis testing* for all the connections in these structures.

Table of contents

Outline

$$\mathbb{X} = (X_1, \ldots, X_p)^T \sim \mathcal{N}_p(0, \Sigma_x)$$

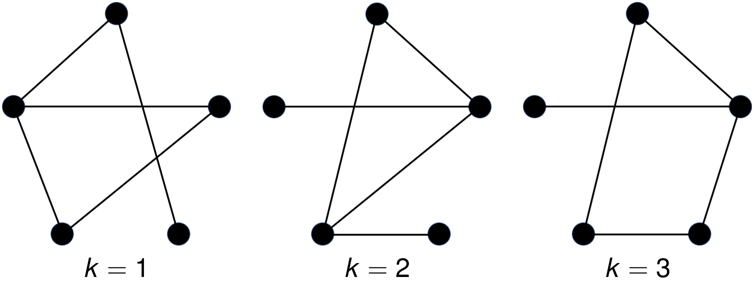
Non-zero entries in the precision matrix $\Omega_x = \Sigma_x^{-1}$ gives edges of the network.



Sparse estimation of Ω_x: Meinshausen and Bühlmann (2006)
Multiple testing and error control: Drton and Perlman (2007).

Multiple Gaussian Graphical models

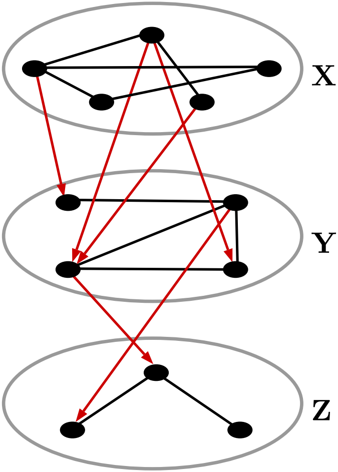
$$\mathbb{X}^k = (X_1^k, \ldots, X_p^k)^T \sim \mathcal{N}_p(0, \Sigma_x^k); \quad \Omega_x^k = (\Sigma_x^k)^{-1}$$
$$k = 1, 2, \ldots, K$$



- Joint estimation of $\{\Omega_x^k\}$: Guo et al. (2011); Ma and Michailidis (2016)
- Difference and similarity testing with FDR control: Liu (2017)

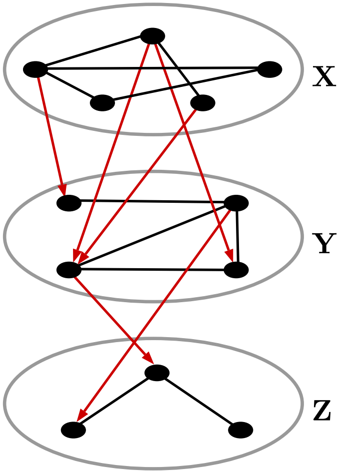
$$\mathbb{E} = (E_1, \ldots, E_q)^T \sim \mathcal{N}_q(0, \Sigma_y);$$
$$\mathbb{F} = (F_1, \ldots, F_r)^T \sim \mathcal{N}_r(0, \Sigma_z);$$
$$\Omega_y = (\Sigma_y)^{-1}, \Omega_z = (\Sigma_z)^{-1}$$
$$\mathbb{Y} = \mathbb{X}^T \mathbf{B} + \mathbb{E},$$
$$\mathbb{Z} = \mathbb{Y}^T \mathbf{C} + \mathbb{F}.$$

- $\Omega_x, \Omega_y, \Omega_z$ give undirected within-layer edges, while **B**, **C** gives directed between-layer edges.
- Sparse estimation of the components: Lin et al. (2016).
- Testing: **??**



$$\mathbb{E} = (E_1, \ldots, E_q)^T \sim \mathcal{N}_q(0, \Sigma_y);$$
$$\mathbb{F} = (F_1, \ldots, F_r)^T \sim \mathcal{N}_r(0, \Sigma_z);$$
$$\Omega_y = (\Sigma_y)^{-1}, \Omega_z = (\Sigma_z)^{-1}$$
$$\mathbb{Y} = \mathbb{X}^T \mathbf{B} + \mathbb{E},$$
$$\mathbb{Z} = \mathbb{Y}^T \mathbf{C} + \mathbb{F}.$$

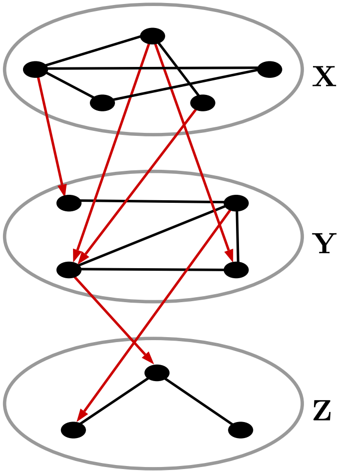
- $\Omega_x, \Omega_y, \Omega_z$ give undirected within-layer edges, while **B**, **C** gives directed between-layer edges.
- Sparse estimation of the components: Lin et al. (2016).
- Testing: **??**



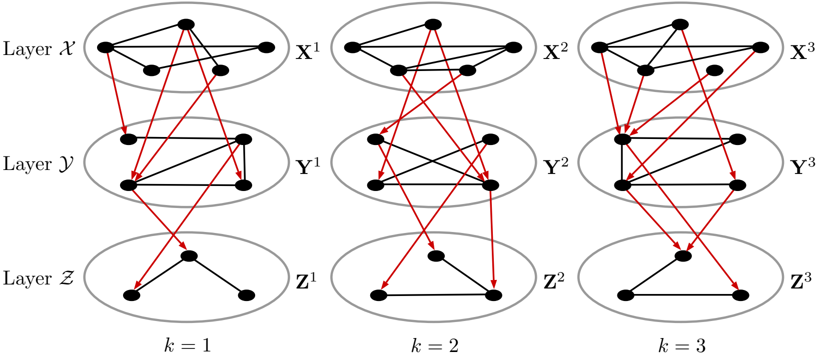
Multi-Layered Gaussian Graphical models

$$\mathbb{E} = (E_1, \ldots, E_q)^T \sim \mathcal{N}_q(0, \Sigma_y);$$
$$\mathbb{F} = (F_1, \ldots, F_r)^T \sim \mathcal{N}_r(0, \Sigma_z);$$
$$\Omega_y = (\Sigma_y)^{-1}, \Omega_z = (\Sigma_z)^{-1}$$
$$\mathbb{Y} = \mathbb{X}^T \mathbf{B} + \mathbb{E},$$
$$\mathbb{Z} = \mathbb{Y}^T \mathbf{C} + \mathbb{F}.$$

- $\Omega_x, \Omega_y, \Omega_z$ give undirected within-layer edges, while **B**, **C** gives directed between-layer edges.
- Sparse estimation of the components: Lin et al. (2016).
- Testing: **??**



Multiple Multi-layered Gaussian Graphical models



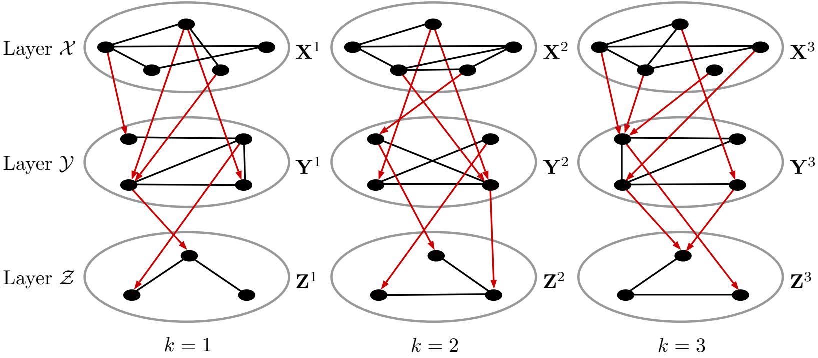
$$\mathbb{E}^k = (E_1^k, \ldots, E_q^k)^T \sim \mathcal{N}_q(0, \Sigma_y^k); \quad \Omega_y^k = (\Sigma_y^k)^{-1}$$

$$\mathbb{F}^k = (F_1^k, \ldots, F_r^k)^T \sim \mathcal{N}_r(0, \Sigma_z^k); \quad \Omega_y^k = (\Sigma_y^k)^{-1}$$

$$\mathbb{Y}^k = (\mathbb{X}^k)^T \mathbf{B}^k + \mathbb{E}^k,$$

$$\mathbb{Z}^k = (\mathbb{Y}^k)^T \mathbf{C}^k + \mathbb{F}^k; \quad k = 1, 2, \ldots, K$$

Multiple Multi-layered Gaussian Graphical models



$$\mathbb{E}^k = (E_1^k, \ldots, E_q^k)^T \sim \mathcal{N}_q(0, \Sigma_y^k); \quad \Omega_y^k = (\Sigma_y^k)^{-1}$$
$$\mathbb{F}^k = (F_1^k, \ldots, F_r^k)^T \sim \mathcal{N}_r(0, \Sigma_z^k); \quad \Omega_y^k = (\Sigma_y^k)^{-1}$$
$$\mathbb{Y}^k = (\mathbb{X}^k)^T \mathbf{B}^k + \mathbb{E}^k,$$
$$\mathbb{Z}^k = (\mathbb{Y}^k)^T \mathbf{C}^k + \mathbb{F}^k; \quad k = 1, 2, \ldots, K$$

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

What we do

- We decompose the multi-layer problem into a series of two layer problems.
- We estimate $\{\Omega_x^k, \Omega_y^k, \mathbf{B}^k\}$ jointly for all k from a single model;
  - Incorporate structural informartion using group sparsity,
  - Propose algorithm to compute solutions, derive their convergence properties.
- Devise a full pairwise testing procedure for rows of $\mathbf{B}^k$;
  - For $K = 2$, propose a test for row-wise differences $\mathbf{b}_i^1 - \mathbf{b}_i^2$;
  - Perform multiple testing for elementwise differences $b_{ij}^1 = b_{ij}^2, j = 1, 2, \ldots, q$ within a row.
- Use simulations for performance evaluation.

Outline

Preliminaries

Preliminaries

- $\mathcal{Y} = \{\mathbf{Y}^1, \ldots, \mathbf{Y}^K\}, \mathcal{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^K\}$;
- $\Omega_x = \{\Omega_x^1, \ldots, \Omega_x^K\}, \Omega_y = \{\Omega_y^1, \ldots, \Omega_y^K\}, \mathcal{B} = \{\mathbf{B}^1, \ldots, \mathbf{B}^K\}$;

- $\mathcal{Y} = \{\mathbf{Y}^1, \ldots, \mathbf{Y}^K\}, \mathcal{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^K\}$;
- $\Omega_x = \{\Omega_x^1, \ldots, \Omega_x^K\}, \Omega_y = \{\Omega_y^1, \ldots, \Omega_y^K\}, \mathcal{B} = \{\mathbf{B}^1, \ldots, \mathbf{B}^K\}$;

Linear model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, with $\mathbf{X} \in \mathbb{R}^{n \times p}, \boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ with $\sigma > 0$;

Lasso: $\widehat{\beta} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\beta\|^2/n + \lambda \|\boldsymbol{\beta}\|_1$;

Preliminaries

- $\mathcal{Y} = \{\mathbf{Y}^1, \ldots, \mathbf{Y}^K\}, \mathcal{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^K\}$;
- $\Omega_x = \{\Omega_x^1, \ldots, \Omega_x^K\}, \Omega_y = \{\Omega_y^1, \ldots, \Omega_y^K\}, \mathcal{B} = \{\mathbf{B}^1, \ldots, \mathbf{B}^K\}$;

Linear model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, with $\mathbf{X} \in \mathbb{R}^{n \times p}, \boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ with $\sigma > 0$;

Lasso: $\widehat{\beta} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\beta\|^2/n + \lambda \|\boldsymbol{\beta}\|_1$;

Group lasso:

$$\widehat{\beta} = \underset{\boldsymbol{\beta}}{\text{argmin}} \frac{1}{n} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \sum_{g \in \mathcal{G}} \|\boldsymbol{\beta}_g\|$$

where $\mathcal{G}$ is a *partition* of $\{1, 2, \ldots, p\}$.

Example: $p = 7, \mathcal{G} = \{[1, 2], [3, 4], [5, 6, 7]\}$. Then

$$\sum_{g \in \mathcal{G}} \|\boldsymbol{\beta}_g\| = \sqrt{\beta_1^2 + \beta_2^2} + \sqrt{\beta_3^2 + \beta_4^2} + \sqrt{\beta_5^2 + \beta_6^2 + \beta_7^2}$$

Outline

Estimation of X-network

1. **Trick:** Take a node- figure out who its neighbors are. Repeat this for all nodes. This infers the full graph structure.
2. *Estimate neighborhood coefficients* of each X-node, say $\zeta_i = (\zeta_i^1, \ldots, \zeta_i^K)$ using the group information on the X-network:

$$\widehat{\zeta}_i = \underset{\zeta_i}{\text{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \|\mathbf{X}_i^k - \mathbf{X}_{-i}^k \zeta_i^k\|^2 + \nu_n P(\zeta) \right\}$$

3. Non-zero supports of $\zeta_i, i = 1, ..., p$ give a skeleton set for the corresponding graphs. *Recover precision matrices* as maximum likelihood estimates over these restricted skeleton sets.

Joint Structural Estimation Method (JSEM)
Ma and Michailidis (2016)

1. **Trick:** Take a node- figure out who its neighbors are. Repeat this for all nodes. This infers the full graph structure.
2. *Estimate neighborhood coefficients* of each X-node, say $\zeta_i = (\zeta_i^1, \ldots, \zeta_i^K)$ using the group information on the X-network:

$$\widehat{\zeta}_i = \underset{\zeta_i}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \|\mathbf{X}_i^k - \mathbf{X}_{-i}^k \zeta_i^k\|^2 + \nu_n P(\zeta) \right\}$$

3. Non-zero supports of $\zeta_i, i = 1, ..., p$ give a skeleton set for the corresponding graphs. *Recover precision matrices* as maximum likelihood estimates over these restricted skeleton sets.

Joint Structural Estimation Method (JSEM)
Ma and Michailidis (2016)

Estimation of X-network

1. **Trick:** Take a node- figure out who its neighbors are. Repeat this for all nodes. This infers the full graph structure.
2. *Estimate neighborhood coefficients* of each X-node, say $\zeta_i = (\zeta_i^1, \ldots, \zeta_i^K)$ using the group information on the X-network:

$$\widehat{\zeta_i} = \underset{\zeta_i}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \| \mathbf{X}_i^k - \mathbf{X}_{-i}^k \zeta_i^k \|^2 + \nu_n P(\zeta) \right\}$$

3. Non-zero supports of $\zeta_i, i = 1, ..., p$ give a skeleton set for the corresponding graphs. *Recover precision matrices* as maximum likelihood estimates over these restricted skeleton sets.

Joint Structural Estimation Method (JSEM)
Ma and Michailidis (2016)

Estimation of X-network

1. **Trick:** Take a node- figure out who its neighbors are. Repeat this for all nodes. This infers the full graph structure.
2. *Estimate neighborhood coefficients* of each X-node, say $\zeta_i = (\zeta_i^1, \ldots, \zeta_i^K)$ using the group information on the X-network:

$$\widehat{\zeta_i} = \underset{\zeta_i}{\text{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \| \mathbf{X}_i^k - \mathbf{X}_{-i}^k \zeta_i^k \|^2 + \nu_n P(\zeta) \right\}$$

3. Non-zero supports of $\zeta_i, i = 1, ..., p$ give a skeleton set for the corresponding graphs. *Recover precision matrices* as maximum likelihood estimates over these restricted skeleton sets.

Joint Structural Estimation Method (JSEM)
Ma and Michailidis (2016)

1. **Trick:** Take a node- figure out who its neighbors are. Repeat this for all nodes. This infers the full graph structure.
2. *Estimate neighborhood coefficients* of each X-node, say $\zeta_i = (\zeta_i^1, \ldots, \zeta_i^K)$ using the group information on the X-network:

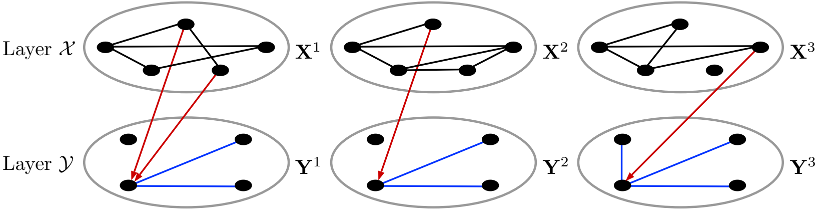
$$\widehat{\zeta_i} = \underset{\zeta_i}{\text{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \| \mathbf{X}_i^k - \mathbf{X}_{-i}^k \zeta_i^k \|^2 + \nu_n P(\zeta) \right\}$$

3. Non-zero supports of $\zeta_i, i = 1, ..., p$ give a skeleton set for the corresponding graphs. *Recover precision matrices* as maximum likelihood estimates over these restricted skeleton sets.

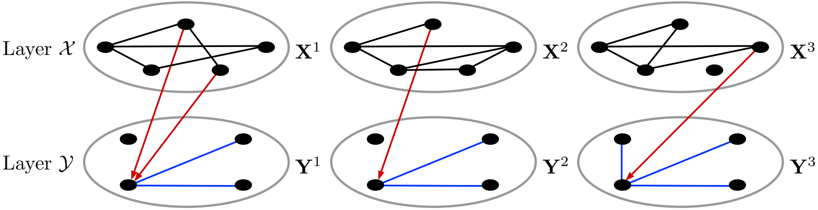
Joint Structural Estimation Method (JSEM)
Ma and Michailidis (2016))

Estimating XY and Y-networks: the objective function

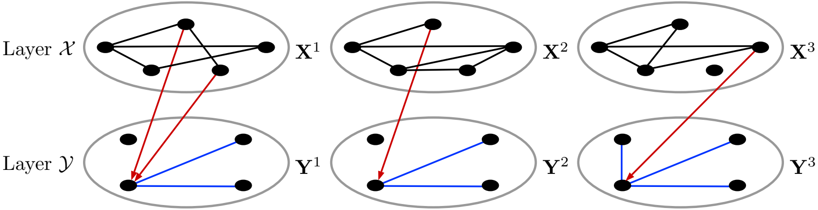
Estimating XY and Y-networks: the objective function



Estimating XY and Y-networks: the objective function



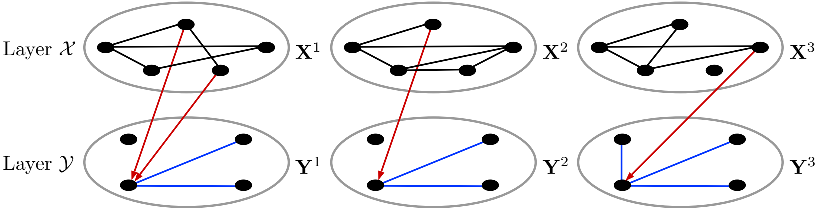
$\mathbf{Y}_j^k$



$$\mathbf{Y}_j^k \quad - \quad \mathbf{E}_{-j}^k \boldsymbol{\theta}_j^k$$

$$\mathbf{E}^k = \mathbf{Y}^k - \mathbf{X}^k \mathbf{B}^k$$

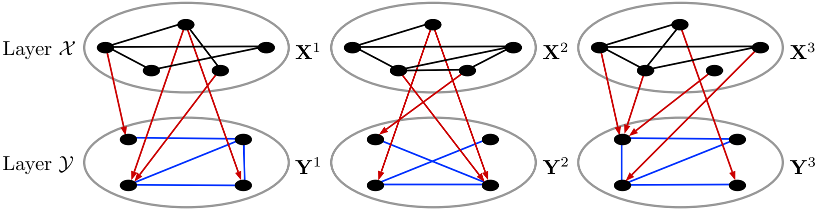
Estimating XY and Y-networks: the objective function



$$\mathbf{Y}_j^k \quad - \quad \mathbf{E}_{-j}^k \boldsymbol{\theta}_j^k \quad - \quad \mathbf{X}^k \mathbf{B}_j^k$$

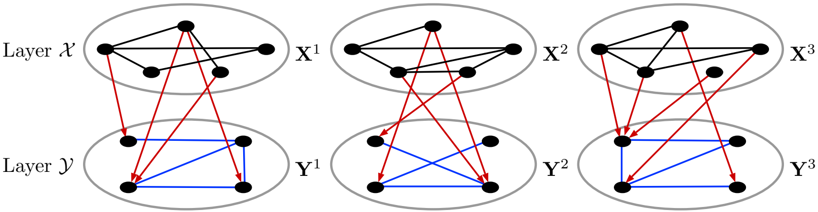
$$\mathbf{E}^k = \mathbf{Y}^k - \mathbf{X}^k \mathbf{B}^k$$

The objective function



$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{q} \left\| \mathbf{Y}_j^k - (\mathbf{Y}_{-j}^k - \mathbf{X}^k \mathbf{B}_{-j}^k) \boldsymbol{\theta}_j^k - \mathbf{X}^k \mathbf{B}_j^k \right\|^2$$

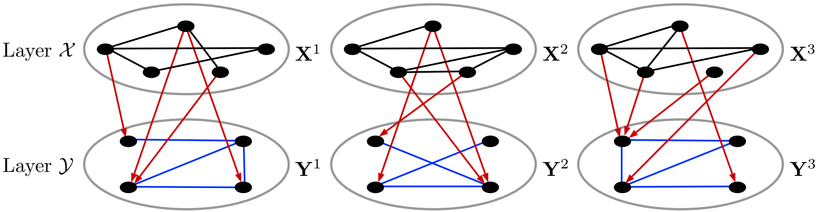
The objective function



$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{q} \left\| \mathbf{Y}_j^k - (\mathbf{Y}_{-j}^k - \mathbf{X}^k \mathbf{B}_{-j}^k) \boldsymbol{\theta}_j^k - \mathbf{X}^k \mathbf{B}_j^k \right\|^2$$

$$+ \lambda_n \sum_{h \in \mathcal{H}} \| \mathbf{B}^{[h]} \|$$

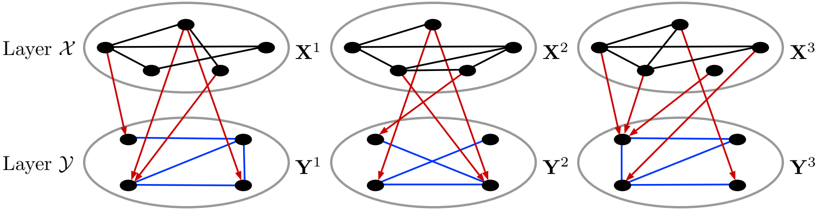
The objective function



$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{q} \left\| \mathbf{Y}_j^k - (\mathbf{Y}_{-j}^k - \mathbf{X}^k \mathbf{B}_{-j}^k) \boldsymbol{\theta}_j^k - \mathbf{X}^k \mathbf{B}_j^k \right\|^2$$

$$+ \lambda_n \sum_{h \in \mathcal{H}} \| \mathbf{B}^{[h]} \| + \gamma_n \sum_{j' \neq j, g \in \mathcal{G}_{jj'}} \| \boldsymbol{\theta}_{jj'}^{[g]} \|$$

The objective function



$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{q} \left\| \mathbf{Y}_j^k - (\mathbf{Y}_{-j}^k - \mathbf{X}^k \mathbf{B}_{-j}^k) \boldsymbol{\theta}_j^k - \mathbf{X}^k \mathbf{B}_j^k \right\|^2$$

$$+ \lambda_n \sum_{h \in \mathcal{H}} \| \mathbf{B}^{[h]} \| + \gamma_n \sum_{j' \neq j, g \in \mathcal{G}_{jj'}} \| \boldsymbol{\theta}_{jj'}^{[g]} \|$$

(Ask me what they are later)

Joint Multiple Multi-Level Estimation (JMMLE)

Joint Multiple Multi-Level Estimation (JMMLE)

1. Solve for $\{\mathcal{B}, \Theta\}$:

$$\{\widehat{\mathcal{B}}, \widehat{\Theta}\} = \underset{\mathcal{B},\Theta}{\text{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{q} \left\| \mathbf{Y}_j^k - (\mathbf{Y}_{-j}^k - \mathbf{X}^k \mathbf{B}_{-j}^k) \boldsymbol{\theta}_j^k - \mathbf{X}^k \mathbf{B}_j^k \right\|^2 + \lambda_n \sum_{h \in \mathcal{H}} \|\mathbf{B}^{[h]}\| + \gamma_n \sum_{j' \neq j, g \in \mathcal{G}_{jj'}} \|\boldsymbol{\theta}_{jj'}^{[g]}\| \right\}$$

Joint Multiple Multi-Level Estimation (JMMLE)

1. Solve for $\{\mathcal{B}, \Theta\}$:

$$\{\widehat{\mathcal{B}}, \widehat{\Theta}\} = \underset{\mathcal{B},\Theta}{\text{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{q} \left\| \mathbf{Y}_j^k - (\mathbf{Y}_{-j}^k - \mathbf{X}^k \mathbf{B}_{-j}^k) \boldsymbol{\theta}_j^k - \mathbf{X}^k \mathbf{B}_j^k \right\|^2 + \lambda_n \sum_{h \in \mathcal{H}} \| \mathbf{B}^{[h]} \| + \gamma_n \sum_{j' \neq j, g \in \mathcal{G}_{jj'}} \| \boldsymbol{\theta}_{jj'}^{[g]} \| \right\}$$

2. Recover Y-precision matrices as MLE over the Y-network skeleton sets

$$\{\widehat{\mathcal{B}}, \widehat{\Theta}\} = \operatorname*{argmin}_{\mathcal{B},\Theta} \{f(\mathcal{Y}, \mathcal{X}, \mathcal{B}, \Theta) + P(\mathcal{B}) + Q(\Theta)\}$$

The objective function is biconvex, so we solve the above by the following alternating iterative algorithm:

1. Start with initial estimates of $\mathcal{B}$ and Θ, say $\mathcal{B}^{(0)}, \Theta^{(0)}$.
2. Iterate:

$$\mathcal{B}^{(t+1)} = \operatorname*{argmin}_{\mathcal{B}} \left\{ f(\mathcal{Y}, \mathcal{X}, \mathcal{B}, \Theta^{(t)}) + Q(\mathcal{B}) \right\}$$

$$\Theta^{(t+1)} = \operatorname*{argmin}_{\Theta} \left\{ f(\mathcal{Y}, \mathcal{X}, \mathcal{B}^{(t+1)}, \Theta) + P(\Theta) \right\}$$

3. Continue till convergence.

For $\lambda_n \geq 4\sqrt{|h_{\max}|}\mathbb{R}_0\sqrt{\frac{\log(pq)}{n}}$, the following hold with probability approaching 1 as $n \to \infty$,

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_1 \leq \frac{48\sqrt{|h_{\max}|}s_\beta \lambda_n}{\psi^*}$$

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| \leq \frac{12\sqrt{s_\beta}\lambda_n}{\psi^*}$$

$$\sum_{h \in \mathcal{H}} \|\boldsymbol{\beta}^{[h]} - \boldsymbol{\beta}_0^{[h]}\| \leq \frac{48 s_\beta \lambda_n}{\psi^*}$$

with $\psi^*, \mathbb{R}_0$ being constants, and $\boldsymbol{\beta} = (\text{vec}(\mathbf{B}^1)^T, \ldots, \text{vec}(\mathbf{B}^K)^T)^T$, $|h_{\max}|$ the maximum group size in $\boldsymbol{\beta}_0$ (the true $\boldsymbol{\beta}$) and s_β the sparsity of $\boldsymbol{\beta}_0$.

Error bounds for $\widehat{\Theta}, \widehat{\Omega}$

For $\gamma_n = 4\sqrt{|g_{\max}|}\mathbb{Q}_0\sqrt{\frac{\log(pq)}{n}}$, the following hold with probability approaching 1 as $n \to \infty$,

$$\|\widehat{\Theta}_j - \Theta_{0,j}\|_F \leq \frac{12\sqrt{s_j}\gamma_n}{\psi}$$

$$\sum_{j \neq j', g \in \mathcal{G}_y^{jj'}} \|\widehat{\boldsymbol{\theta}}_{jj'}^{[g]} - \boldsymbol{\theta}_{0,jj'}^{[g]}\| \leq \frac{48 s_j \gamma_n}{\psi}$$

$$\frac{1}{K}\sum_{k=1}^{K} \|\widehat{\Omega}_y^k - \Omega_y^k\|_F \leq O\left(\frac{\sqrt{S}\gamma_n}{\sqrt{K}}\right)$$

with $\psi, \mathbb{Q}_0$ being constants, $|g_{\max}|$ the maximum group size in Θ_0, s_j the sparsity of Θ_j and $S = \sum_j s_j$.

Outline

Motivation

- Find out if an upper layer variable has a significant downstream effects, e.g. if a gene influences the activity of *any* protein.
- How does this downstream effect vary across different horizontal category, e.g. gene has downstream effect on patient profile 1 but not on profile 2.
- Which of the downstream effects are significant? How do they differ across subtypes? e.g. which exact proteins does the gene affect for each patient profile

Motivation

- Find out if an upper layer variable has a significant downstream effects, e.g. if a gene influences the activity of *any* protein.
- How does this downstream effect vary across different horizontal category, e.g. gene has downstream effect on patient profile 1 but not on profile 2.
- Which of the downstream effects are significant? How do they differ across subtypes? e.g. which exact proteins does the gene affect for each patient profile

Motivation

- Find out if an upper layer variable has a significant downstream effects, e.g. if a gene influences the activity of *any* protein.
- How does this downstream effect vary across different horizontal category, e.g. gene has downstream effect on patient profile 1 but not on profile 2.
- Which of the downstream effects are significant? How do they differ across subtypes? e.g. which exact proteins does the gene affect for each patient profile

- Find out if an upper layer variable has a significant downstream effects, e.g. if a gene influences the activity of *any* protein.
- How does this downstream effect vary across different horizontal category, e.g. gene has downstream effect on patient profile 1 but not on profile 2.
- Which of the downstream effects are significant? How do they differ across subtypes? e.g. which exact proteins does the gene affect for each patient profile

Debiased estimators (Zhang and Zhang, 2014)

Debiased estimators (Zhang and Zhang, 2014)

- Lasso: $\widehat{\beta} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\beta\|^2/n + \lambda\|\boldsymbol{\beta}\|_1$;

Debiased estimators (Zhang and Zhang, 2014)

- Lasso: $\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\beta\|^2/n + \lambda\|\boldsymbol{\beta}\|_1$;
- Debiased estimator:

$$\hat{\beta}_j^{(\text{deb})} = \hat{\beta}_j + \frac{\mathbf{z}_j^T(\mathbf{y} - \mathbf{X}\widehat{\beta})}{\mathbf{z}^T\mathbf{x}_j},$$

where $\mathbf{z}_j$ is the vector of residuals from the ℓ_1-penalized regression of $\mathbf{x}_j$ on $\mathbf{X}_{-j}$.

Debiased estimators (Zhang and Zhang, 2014)

- Lasso: $\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\beta\|^2/n + \lambda\|\boldsymbol{\beta}\|_1$;
- Debiased estimator:

$$\hat{\beta}_j^{(\text{deb})} = \hat{\beta}_j + \frac{\mathbf{z}_j^T(\mathbf{y} - \mathbf{X}\widehat{\beta})}{\mathbf{z}^T\mathbf{x}_j},$$

  where $\mathbf{z}_j$ is the vector of residuals from the ℓ_1-penalized regression of $\mathbf{x}_j$ on $\mathbf{X}_{-j}$.
- Asymptotic normal distribution:

$$\frac{\hat{\beta}_j^{(\text{deb})} - \beta_j^0}{\|\mathbf{z}_j\|/|\mathbf{z}_j^T\mathbf{x}_j|} \sim N(0, \sigma^2)$$

Debiased estimators (Zhang and Zhang, 2014)

- Lasso: $\widehat{\beta} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\beta\|^2/n + \lambda\|\boldsymbol{\beta}\|_1$;
- Debiased estimator:

$$\hat{\beta}_j^{(\text{deb})} = \hat{\beta}_j + \frac{\mathbf{z}_j^T(\mathbf{y} - \mathbf{X}\widehat{\beta})}{\mathbf{z}^T\mathbf{x}_j},$$

  where $\mathbf{z}_j$ is the vector of residuals from the ℓ_1-penalized regression of $\mathbf{x}_j$ on $\mathbf{X}_{-j}$.
- Asymptotic normal distribution:

$$\frac{\hat{\beta}_j^{(\text{deb})} - \beta_j^0}{\|\mathbf{z}_j\|/|\mathbf{z}_j^T\mathbf{x}_j|} \sim N(0, \sigma^2)$$

The debiasing factor for the j^{th} coefficient is obtained by taking residuals from the regularized regression and scale them using the *projection of $\mathbf{x}_j$ onto a space approximately orthogonal to it*.

What we do

- We propose a debiased estimator for $\mathbf{b}_i^k$ that makes use of already computed model quantities, and establish asymptotic properties of its scaled version,
- We assume $K = 2$, and propose an asymptotic test for detecting differential effects of a variable in the upper layer, i.e. testing for the null hypothesis $H_0 : \mathbf{b}_{0i}^1 = \mathbf{b}_{0i}^2$,
- We also propose pairwise simultaneous tests with False Discovery Rate (FDR) control across $j = 1, \ldots, q$ for detecting the elementwise differences $b_{0ij}^1 = b_{0ij}^2$.

- We propose a debiased estimator for $\mathbf{b}_i^k$ that makes use of already computed model quantities, and establish asymptotic properties of its scaled version,
- We assume $K = 2$, and propose an asymptotic test for detecting differential effects of a variable in the upper layer, i.e. testing for the null hypothesis $H_0 : \mathbf{b}_{0i}^1 = \mathbf{b}_{0i}^2$,
- We also propose pairwise simultaneous tests with False Discovery Rate (FDR) control across $j = 1, \ldots, q$ for detecting the elementwise differences $b_{0ij}^1 = b_{0ij}^2$.

What we do

- We propose a debiased estimator for $\mathbf{b}_i^k$ that makes use of already computed model quantities, and establish asymptotic properties of its scaled version,
- We assume $K = 2$, and propose an asymptotic test for detecting differential effects of a variable in the upper layer, i.e. testing for the null hypothesis $H_0 : \mathbf{b}_{0i}^1 = \mathbf{b}_{0i}^2$,
- We also propose pairwise simultaneous tests with False Discovery Rate (FDR) control across $j = 1, \ldots, q$ for detecting the elementwise differences $b_{0ij}^1 = b_{0ij}^2$.

- We propose a debiased estimator for $\mathbf{b}_i^k$ that makes use of already computed model quantities, and establish asymptotic properties of its scaled version,
- We assume $K = 2$, and propose an asymptotic test for detecting differential effects of a variable in the upper layer, i.e. testing for the null hypothesis $H_0 : \mathbf{b}_{0i}^1 = \mathbf{b}_{0i}^2$,
- We also propose pairwise simultaneous tests with False Discovery Rate (FDR) control across $j = 1, \ldots, q$ for detecting the elementwise differences $b_{0ij}^1 = b_{0ij}^2$.

Outline

Simulation setup

- Number of categories (K) = 5;
- Structured $\{\Omega_x\}, \{\Omega_y\}, \mathcal{B}$;
- Groups in $\mathcal{B}, \Omega_x$ are non-zero with probability $5/p$, and their elements come from Unif$[-1, -0.5] \cup [0.5, 1]$;
- Groups in Ω_y are non-zero with probability $5/q$, and their elements come from Unif$[-1, -0.5] \cup [0.5, 1]$;
- We generate size-n i.i.d. samples $\mathbf{X}^k$ from $\mathcal{N}_p(0, \Sigma_x^k)$, and $\mathbf{E}^k$ from $\mathcal{N}_p(0, \Sigma_y^k)$, then obtain $\mathbf{Y}^k = \mathbf{X}^k \mathbf{B}^k + \mathbf{E}^k$;
- 50 Replications.
- Tuning parameters:

$$\gamma_n \in \{0.3, 0.4, ..., 1\} \sqrt{\frac{\log q}{n}}, \lambda_n \in \{0.4, 0.6, ..., 1.8\} \sqrt{\frac{\log p}{n}}$$

Evaluation metrics

1. True positive Rate-

$$\text{TPR}(\widehat{\mathcal{B}}) = \frac{1}{K}\sum_{k=1}^{K} \frac{|\,\text{supp}(\widehat{\mathbf{B}}^k) \cup \text{supp}(\mathbf{B}_0^k)|}{|\,\text{supp}(\mathbf{B}_0^k)|}$$

2. True negatives-

$$\text{TNR}(\widehat{\mathcal{B}}) = \frac{1}{K}\sum_{k=1}^{K} \frac{|\,\text{supp}^c(\widehat{\mathbf{B}}^k) \cup \text{supp}^c(\mathbf{B}_0^k)|}{|\,\text{supp}^c(\mathbf{B}_0^k)|}$$

3. Relative error in Frobenius norm-

$$\text{RF}(\widehat{\mathcal{B}}) = \frac{1}{K}\sum_{k=1}^{K} \frac{\|\widehat{\mathbf{B}}^k - \mathbf{B}_0^k\|_F}{\|\mathbf{B}_0^k\|_F}$$

4. Matthews correlation coefficient (MCC).

Same metrics are used for $\widehat{\Theta}$.

Results

(π_x, π_y)	(p, q, n)	Method	TPR	TNR	MCC	RF
$(5/p, 5/q)$	(60,30,100)	JMMLE	**0.97(0.02)**	0.99(0.003)	**0.96(0.014)**	0.24(0.033)
		Separate	0.96(0.018)	0.99(0.004)	0.93(0.014)	0.22(0.029)
	(30,60,100)	JMMLE	0.97(0.013)	0.99(0.002)	**0.96(0.008)**	0.27(0.024)
		Separate	0.99(0.009)	0.99(0.003)	0.93(0.017)	0.18(0.021)
	(200,200,150)	JMMLE	0.98(0.011)	**1.0(0)**	**0.99(0.005)**	**0.16(0.025)**
		Separate	0.99(0.001)	0.99 (0.001)	0.88(0.009)	0.18(0.007)
	(300,300,150)	JMMLE	**1.0(0.001)**	**1.0(0)**	**0.99(0.001)**	**0.14 (0.015)**
		Separate	1.0(0.001)	0.99(0.001)	0.84(0.01)	0.21(0.007)
$(30/p, 30/q)$	(200,200,100)	JMMLE	**0.97(0.017)**	**1.0(0)**	**0.98(0.008)**	**0.21(0.032)**
		Separate	0.32(0.01)	0.99(0.001)	0.49(0.009)	0.85(0.06)
	(200,200,200)	JMMLE	**0.99(0.006)**	**1.0(0)**	**0.99(0.007)**	**0.13(0.016)**
		Separate	0.97(0.004)	0.98(0.001)	0.93(0.002)	0.19(0.07)

Table of outputs for estimation of regression matrices, giving empirical mean and standard deviation (in brackets) of each evaluation metric over 50 replications.

Results

(π_x, π_y)	(p, q, n)	Method	TPR	TNR	MCC	RF
$(5/p, 5/q)$	(60,30,100)	JMMLE	0.76(0.018)	0.90(0.006)	**0.61(0.024)**	**0.32(0.008)**
		Separate	0.77(0.031)	0.92(0.007)	0.56(0.03)	0.51(0.017)
		JSEM	0.24(0.013)	0.8(0.003)	0.05(0.015)	1.03(0.002)
	(30,60,100)	JMMLE	0.7(0.018)	**0.94(0.002)**	0.55(0.018)	**0.3(0.005)**
		Separate	0.76(0.041)	0.89(0.015)	0.59(0.039)	0.49(0.014)
		JSEM	0.13(0.005)	0.9(0.001)	0.03(0.007)	1.04(0.001)
	(200,200,150)	JMMLE	0.68(0.017)	**0.98(0)**	0.48(0.013)	**0.26(0.002)**
		Separate	0.78(0.019)	0.97(0.001)	0.55(0.012)	0.6(0.007)
		JSEM	0.05(0.002)	0.97(0)	0.02(0.002)	1.01(0)
	(300,300,150)	JMMLE	0.**71(0.014)**	**0.98(0)**	0.44(0.008)	**0.25(0.002)**
		Separate	0.71(0.017)	0.98(0.001)	0.51(0.011)	0.59(0.005)
		JSEM	0.04(0.002)	0.98(0)	0.02(0.002)	1.01(0)
$(30/p, 30/q)$	(200,200,100)	JMMLE	**0.77(0.016)**	**0.98(0)**	**0.46(0.013)**	**0.31(0.003)**
		Separate	0.57(0.027)	0.44(0.007)	0.04(0.008)	0.84(0.002)
		JSEM	0.05(0.002)	0.97(0)	0.01(0.002)	1.01(0)
	(200,200,200)	JMMLE	**0.76(0.018)**	**0.98(0)**	**0.55(0.015)**	**0.27(0.004)**
		Separate	0.73(0.023)	0.94(0.003)	0.39(0.017)	0.62(0.011)
		JSEM	0.05(0.002)	0.97(0)	0.03(0.003)	1.01(0)

Table of outputs for estimation of lower layer precision matrices over 50 replications.

- Set $K = 2$, then randomly assign each element of $\mathbf{B}_0^1$ as non-zero w.p. π, then draw their values from Unif$\{[-1, -0.5] \cup [0.5, 1]\}$ independently.
- Generate a matrix of differences $\mathbf{D}$, where $(\mathbf{D})_{ij}$ takes values –1, 1, 0 w.p. 0.1, 0.1 and 0.8, respectively. Finally set $\mathbf{B}_0^2 = \mathbf{B}_0^1 + \mathbf{D}$.
- Identical sparsity structures for the pairs of X- and Y-precision matrices.
- Type-I error set at 0.05, FDR controlled at 0.2.
- Empirical sizes of global tests are calculated from estimators obtained from a separate set of data generated by setting all elements of $\mathbf{D}$ to 0.

Results

(π_x, π_y)	(p, q)	n	Global test		Simultaneous tests	
			Power	Size	Power	FDR
$(5/p, 5/q)$	(60,30)	100	0.977 (0.018)	0.058 (0.035)	0.937 (0.021)	0.237 (0.028)
		200	0.987 (0.016)	0.046 (0.032)	0.968 (0.013)	0.218 (0.032)
	(30,60)	100	0.985 (0.018)	0.097 (0.069)	0.925 (0.022)	0.24 (0.034)
		200	0.990 (0.02)	0.119 (0.059)	0.958 (0.024)	0.245 (0.041)
	(200,200)	150	0.987 (0.005)	0.004 (0.004)	0.841 (0.13)	0.213 (0.007)
	(300,300)	150	0.988 (0.002)	0.002 (0.003)	0.546 (0.035)	0.347 (0.017)
		300	0.998 (0.003)	0.000 (0.001)	0.989 (0.003)	0.117 (0.006)
$(30/p, 30/q)$	(200,200)	100	0.994 (0.005)	0.262 (0.06)	0.479 (0.01)	0.557 (0.006)
		200	0.998 (0.004)	0.020 (0.01)	0.962 (0.003)	0.266 (0.007)
		300	0.999 (0.002)	0.011 (0.008)	0.990 (0.004)	0.185 (0.009)

Table of outputs for hypothesis testing.

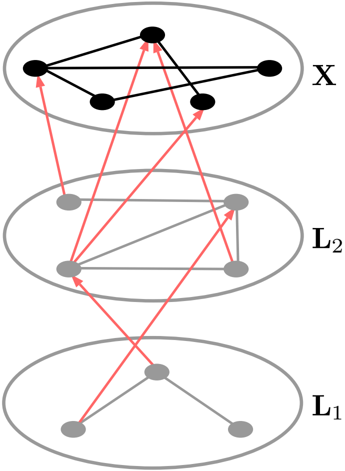
Future work

- **Application to multi-omics data**;
- Beyond pairwise testing: global and simultaneous tests for $K > 2$;
- Multi-level estimation and testing for model assumptions other than structured sparsity;
- Hypothesis testing for complex high-dimensional models;
- Non-gaussian data;
- **Graphical models with non-linear interactions**.

Outline

Graphical models with non-linear interactions

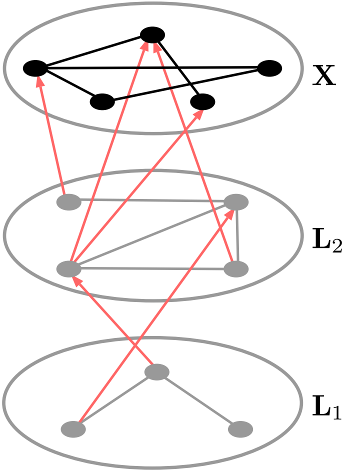
- Take the multi-layer structure as a *generative model*.
- Only the top layer is observed, other layers are composed of latent variables.



- Take the multi-layer structure as a *generative model*.
- Only the top layer is observed, other layers are composed of latent variables.

$$\mathbb{L}_1 = (L_{11}, \ldots, L_{1r})^T \sim \mathcal{N}_r(0, \Sigma_1);$$
$$\mathbb{L}_2 = \phi(\mathbb{L}_1^T \mathbf{B}) + \mathbb{E},$$
$$\mathbb{X} = \phi(\mathbb{L}_2^T \mathbf{C}) + \mathbb{F},$$
$$\mathbb{E} = (E_1, \ldots, E_q)^T \sim \mathcal{N}_q(0, \Sigma_2);$$
$$\mathbb{F} = (F_1, \ldots, F_p)^T \sim \mathcal{N}_p(0, \Sigma_x).$$

where ϕ is a known activation function.

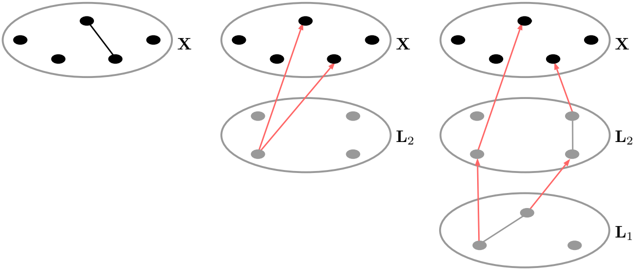


Connections

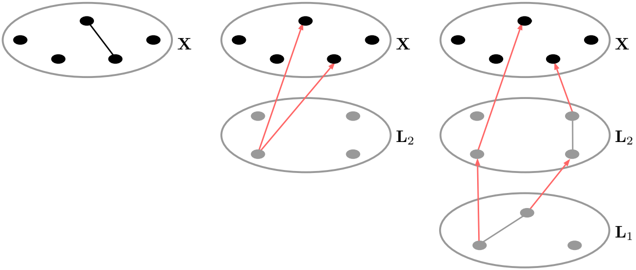
- Non-linear generalization of a factor model.
- A general version: $\mathbb{L}_2 = f_1(\mathbb{L}_1) + \mathbb{E}$ etc. for unknown function f_1, has been proposed as Deep Latent Gaussian Model (Rezende et al., 2014).
- The choice $\phi(\mathbb{L}^T \mathbf{B}) \equiv \phi(\mathbb{L})^T \mathbf{B}$ corresponds to Non-linear Gaussian belief networks (Frey and Hinton, 1999).

Our plan

Incorporate *sparse* estimation of the model parameters to model non-linear interactions.



Incorporate *sparse* estimation of the model parameters to model non-linear interactions.



- Monte-Carlo EM to maximize a variational lower bound of the likelihood,
- Theoretical properties of estimates

Conclusion

- We proposed a general framework to model data in *complex hierarchical structures*, with a focus on multi-level biological Omics datasets;
- We provide an *estimation algorithm* and *testing methodology* for the parameters involved, with theoretical results ensuring the validity of the methods;
- The general nature of the work leaves many *directions for future developments*.

- We proposed a general framework to model data in *complex hierarchical structures*, with a focus on multi-level biological Omics datasets;
- We provide an *estimation algorithm* and *testing methodology* for the parameters involved, with theoretical results ensuring the validity of the methods;
- The general nature of the work leaves many *directions for future developments*.

Conclusion

- We proposed a general framework to model data in *complex hierarchical structures*, with a focus on multi-level biological Omics datasets;
- We provide an *estimation algorithm* and *testing methodology* for the parameters involved, with theoretical results ensuring the validity of the methods;
- The general nature of the work leaves many *directions for future developments*.

Conclusion

- We proposed a general framework to model data in *complex hierarchical structures*, with a focus on multi-level biological Omics datasets;
- We provide an *estimation algorithm* and *testing methodology* for the parameters involved, with theoretical results ensuring the validity of the methods;
- The general nature of the work leaves many *directions for future developments*.

References

Preprint available at: https://arxiv.org/abs/1803.03348

Drton, M. and Perlman, M. D. (2007). Multiple Testing and Error Control in Gaussian Graphical Model Selection. *Statist. Sci.*, 22(3):430–449.

Frey, B. J. and Hinton, G. E. (1999). Variational learning in nonlinear Gaussian belief networks. *Neural Comput.*, 11(1):193–213.

Gligorijević, V. and Pržulj, N. (2015). Methods for biological data integration: perspectives and challenges. *J. R. Soc. Interface*, 12(112):20150571.

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15.

Lin, J., Basu, S., Banerjee, M., and Michailidis, G. (2016). Penalized Maximum Likelihood Estimation of Multi-layered Gaussian Graphical Models. *J. Mach. Learn. Res.*, 17:5097–5147.

Liu, W. (2017). Structural similarity and difference testing on multiple sparse Gaussian graphical models. *Ann. Statist.*, 45(6):2680–2707.

Ma, J. and Michailidis, G. (2016). Joint Structural Estimation of Multiple Graphical Models. *J. Mach. Learn. Res.*, 17:5777–5824.

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the Â¨ lasso. *Ann. Statist.*, 34(3):1436–1462.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML Proceedings*, volume 32, pages 1278–1286.

References

Preprint available at: https://arxiv.org/abs/1803.03348

Drton, M. and Perlman, M. D. (2007). Multiple Testing and Error Control in Gaussian Graphical Model Selection. *Statist. Sci.*, 22(3):430–449.

Frey, B. J. and Hinton, G. E. (1999). Variational learning in nonlinear Gaussian belief networks. *Neural Comput.*, 11(1):193–213.

Gligorijević, V. and Pržulj, N. (2015). Methods for biological data integration: perspectives and challenges. *J. R. Soc. Interface*, 12(112):20150571.

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15.

Lin, J., Basu, S., Banerjee, M., and Michailidis, G. (2016). Penalized Maximum Likelihood Estimation of Multi-layered Gaussian Graphical Models. *J. Mach. Learn. Res.*, 17:5097–5147.

Liu, W. (2017). Structural similarity and difference testing on multiple sparse Gaussian graphical models. *Ann. Statist.*, 45(6):2680–2707.

Ma, J. and Michailidis, G. (2016). Joint Structural Estimation of Multiple Graphical Models. *J. Mach. Learn. Res.*, 17:5777–5824.

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the Â¨ lasso. *Ann. Statist.*, 34(3):1436–1462.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML Proceedings*, volume 32, pages 1278–1286.

THANK YOU!