Statistics, Computations, and Genomics Applications

AMCS PhD Candidacy Oral Exam

Changhao Ge

University of Pennsylvania

May, 2025

Table of Contents

Introduction

Estimation of many gene expression networks

CNV prediction with Neural Network

Future projects - Statistical and Computational Methods for Perturb Seq Data

Overview

Statistics

&

Computation

&

Applications

- Modeling: Linear, Additive, Graphical models...
- Bayesian Analysis: Sparse priors, Hierarchical models...
- Causal Inference: DAGs, Linear SEMs...
- Other: Neural Networks...

- Efficient Algorithms: VI, EM, ADMM...
- HPC: Parallel computing, GPU acceleration...
- Programming Language optimization: C/C++ as core, R/Python as wrapper...

- Estimation: Gene co-expression networks, DAGs...
- Prediction: Copy Number Variation of tumor...
- Inference: Mediation Effect analysis of Perturb Seq...
- Other: Multi-task Learning...

Table of Contents

Introduction

Estimation of many gene expression networks

CNV prediction with Neural Network

Future projects - Statistical and Computational Methods for Perturb Seq Data

Progress in publication

Ge C and Li H (2025): Simultaneous Estimation of Many Sparse Networks via Hierarchical Poisson Log-Normal Model. *Journal of Computational and Graphical Statistics*, revision submitted. https://arxiv.org/abs/2409.12275

Single-cell gene expression networks

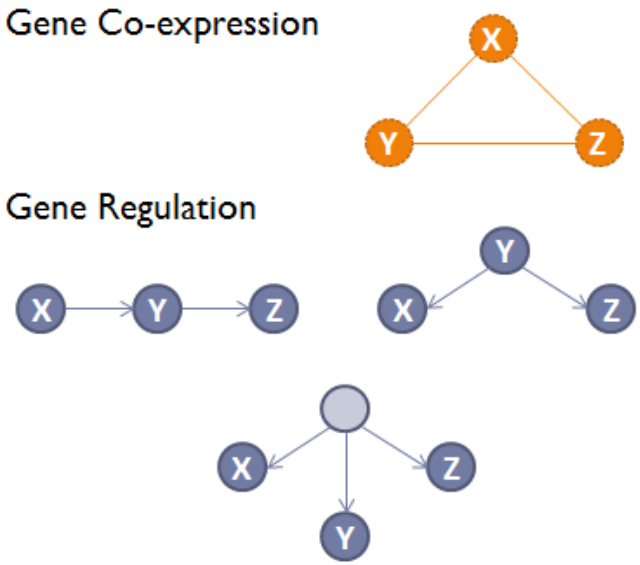


Figure: By Smh.oloomi, CC BY-SA 4.0

	YDL229W	YDL204W	YDL198C	YDL192W	YDL191W
1	4	0	1	5	4
2	4	3	0	1	3
3	18	0	2	7	8
4	3	2	0	4	1
5	3	0	0	1	8
6	10	2	4	6	17
7	3	0	0	3	0
8	1	0	0	3	2
9	5	1	2	16	11
10	13	1	3	5	8
11	3	7	1	9	8
12	3	0	0	4	2
13	8	0	1	8	4
14	3	0	0	5	7
15	0	3	1	1	5
16	3	3	0	2	4

Figure: Single-cell RNA raw read counts

Poisson Log-Normal model

- $\boldsymbol{y}_i^{(k)} = (y_{i1}^{(k)}, \cdots, y_{ip}^{(k)})^\intercal \in \mathbb{N}^p$ is the read counts of p genes for the i-th cell in the k-th group
- $\boldsymbol{z}_i^{(k)} = (z_{i1}^{(k)}, \cdots, z_{id}^{(k)})^\intercal \in \mathbb{R}^d$ represents the d-dimensional covariates
- the offset $\boldsymbol{o}_i^{(k)} = (o_{i1}^{(k)}, \cdots, o_{ip}^{(k)})^\intercal \in \mathbb{R}^p$ represents the library sizes
- Poisson Log-Normal model:

$$\boldsymbol{X}_i^{(k)} \sim N(\boldsymbol{\lambda}_i^{(k)}, (\boldsymbol{\Omega}^{(k)})^{-1}),$$
$$y_{ij}^{(k)} \sim \mathrm{Pois}(\exp(X_{ij}^{(k)})),$$

$\boldsymbol{\lambda}_i^{(k)} = \boldsymbol{o}_i^{(k)} + \boldsymbol{\beta}^{(k)} \boldsymbol{z}_i^{(k)}$ is the mean vector with coefficient $\boldsymbol{\beta}^{(k)} = (\boldsymbol{\beta}_1^{(k)}, \cdots, \boldsymbol{\beta}_p^{(k)})^\intercal \in \mathbb{R}^{p \times d}$, and $\boldsymbol{\Omega}^{(k)}$ is the sparse precision matrix.

Poisson Log-Normal model

- $\boldsymbol{y}_i^{(k)} = (y_{i1}^{(k)}, \cdots, y_{ip}^{(k)})^\intercal \in \mathbb{N}^p$ is the read counts of p genes for the i-th cell in the k-th group
- $\boldsymbol{z}_i^{(k)} = (z_{i1}^{(k)}, \cdots, z_{id}^{(k)})^\intercal \in \mathbb{R}^d$ represents the d-dimensional covariates
- the offset $\boldsymbol{o}_i^{(k)} = (o_{i1}^{(k)}, \cdots, o_{ip}^{(k)})^\intercal \in \mathbb{R}^p$ represents the library sizes
- Poisson Log-Normal model:

$$\boldsymbol{X}_i^{(k)} \sim N(\boldsymbol{\lambda}_i^{(k)}, (\boldsymbol{\Omega}^{(k)})^{-1}),$$
$$y_{ij}^{(k)} \sim \mathrm{Pois}(\exp(X_{ij}^{(k)})),$$

$\boldsymbol{\lambda}_i^{(k)} = \boldsymbol{o}_i^{(k)} + \boldsymbol{\beta}^{(k)} \boldsymbol{z}_i^{(k)}$ is the mean vector with coefficient $\boldsymbol{\beta}^{(k)} = (\boldsymbol{\beta}_1^{(k)}, \cdots, \boldsymbol{\beta}_p^{(k)})^\intercal \in \mathbb{R}^{p \times d}$, and $\boldsymbol{\Omega}^{(k)}$ is the sparse precision matrix.

Multi-task Learning of Many Networks

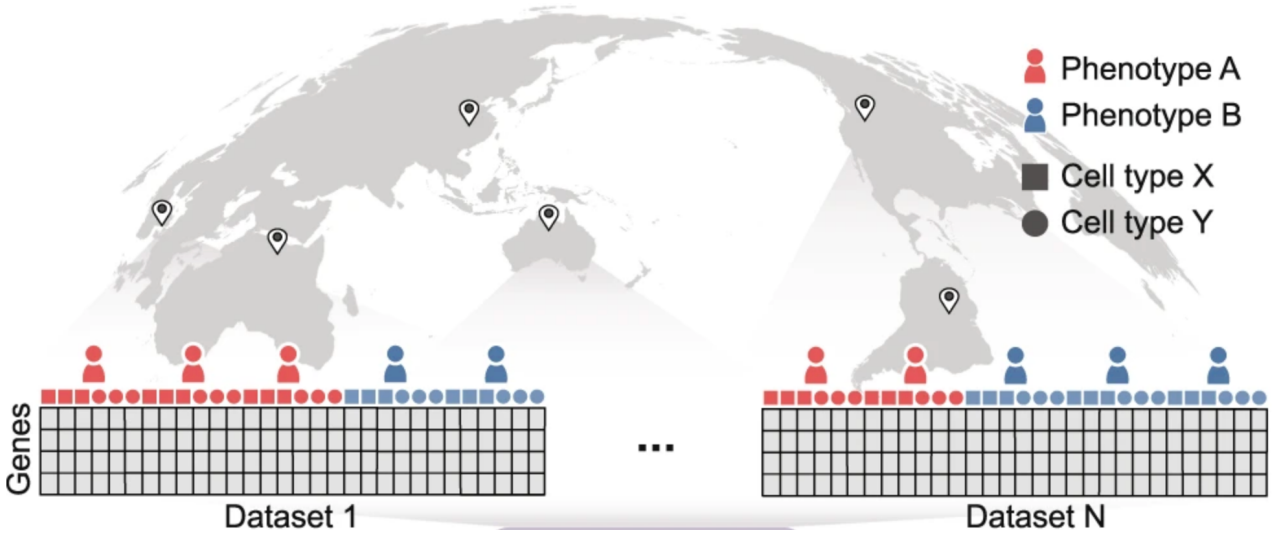


Figure: Illustration of multi-task Learning in biology studies (Lin et al., 2023)

Examples:
Different individuals with individual-specific networks;
Single cell networks under different growth conditions;
Networks of cells of different cell types;
Single cell data of different tissues.

Bayesian Multi-task Learning of Networks

- The hierarchical prior for the (i, j)-th element $(i \neq j)$ of the precision matrix in group k has the form of:

$$\mathbb{P}(\Omega_{ij}^{(k)}|\gamma_{ij}) = \begin{cases} \frac{1}{2v_1}\exp(-|\Omega_{ij}^{(k)}|/v_1), & \text{if } \gamma_{ij} = 1, \\ \frac{1}{2v_0}\exp(-|\Omega_{ij}^{(k)}|/v_0), & \text{if } \gamma_{ij} = 0, \end{cases}$$

where $\gamma_{ij} \sim \mathrm{Bern}(p_0)$, $v_1 > v_0 > 0$ ensures the model is well-defined.

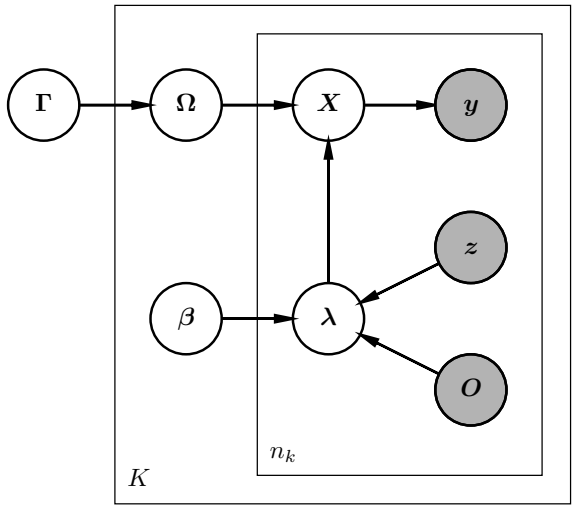


Figure: Graphical representation of the SPLN model

Variational EM Algorithm: Overview

- The posterior

$$\mathbb{P}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{y}) \propto \mathbb{P}(\boldsymbol{\Omega}) \cdot \prod_{k,i} \int \left(\prod_{j=1}^{p} \exp(y_{ij}^{(k)} X_{ij}^{(k)} - \exp(X_{ij}^{(k)})) \right) \cdot N(\boldsymbol{X}_i^{(k)}; \boldsymbol{\lambda}_i^{(k)}, (\boldsymbol{\Omega}^{(k)})^{-1}) d\boldsymbol{X}_i^{(k)}$$

  is intractable.

- EM algorithm: computing

$$\begin{aligned} Q(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)}) &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot|\boldsymbol{y},\boldsymbol{\Omega}^{(\cdot,t)},\boldsymbol{\beta}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Gamma}) \\ &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot|\boldsymbol{y},\boldsymbol{\Omega}^{(\cdot,t)},\boldsymbol{\beta}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{y}, \boldsymbol{X}|\boldsymbol{\Omega}, \boldsymbol{\beta}) \mathbb{P}(\boldsymbol{\Omega}|\boldsymbol{\Gamma}) + C, \end{aligned}$$

  where

$$\log \mathbb{P}(\boldsymbol{y}, \boldsymbol{X}|\boldsymbol{\Omega}, \boldsymbol{\beta}) = \sum_{k=1}^{K} \log \mathbb{P}(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)}) \mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)}).$$

Variational EM Algorithm: Overview

- The posterior

$$\mathbb{P}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{y}) \propto \mathbb{P}(\boldsymbol{\Omega}) \cdot \prod_{k,i} \int \left(\prod_{j=1}^{p} \exp(y_{ij}^{(k)} X_{ij}^{(k)} - \exp(X_{ij}^{(k)})) \right) \cdot N(\boldsymbol{X}_i^{(k)}; \boldsymbol{\lambda}_i^{(k)}, (\boldsymbol{\Omega}^{(k)})^{-1}) d\boldsymbol{X}_i^{(k)}$$

  is intractable.

- EM algorithm: computing

$$\begin{aligned} Q(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)}) &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot|\boldsymbol{y},\boldsymbol{\Omega}^{(\cdot,t)},\boldsymbol{\beta}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Gamma}) \\ &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot|\boldsymbol{y},\boldsymbol{\Omega}^{(\cdot,t)},\boldsymbol{\beta}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{y}, \boldsymbol{X}|\boldsymbol{\Omega}, \boldsymbol{\beta}) \mathbb{P}(\boldsymbol{\Omega}|\boldsymbol{\Gamma}) + C, \end{aligned}$$

  where

$$\log \mathbb{P}(\boldsymbol{y}, \boldsymbol{X}|\boldsymbol{\Omega}, \boldsymbol{\beta}) = \sum_{k=1}^{K} \log \mathbb{P}(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)}) \mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)}).$$

Variational EM Algorithm: E step

- Therefore, the Q function can be derived as (up to a constant):

$$Q(\boldsymbol{\Omega}, \boldsymbol{\beta} | \boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)}) = \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{X}^{(k)} \sim \mathbb{P}(\cdot | \boldsymbol{y}^{(k)}, \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)})} \log \mathbb{P}(\boldsymbol{X}^{(k)} | \boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)}) + \mathbb{E}_{\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot | \boldsymbol{\Omega}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{\Omega} | \boldsymbol{\Gamma}).$$

- However, this Q function still lacks an explicit form, because the normalizing constant has the expression of

$$\mathbb{P}(\boldsymbol{y}_i^{(k)} | \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)}) = \int \prod_{j=1}^{p} \mathrm{Pois}(y_{ij}^{(k)}; \exp(-e^{X_{ij}^{(k)}})) \cdot N(\boldsymbol{X}_i^{(k)}; \boldsymbol{\lambda}_i^{(k,t)}, (\boldsymbol{\Omega}^{(k,t)})^{-1}) d\boldsymbol{X}_i^{(k)},$$

  which includes an Exponential integral (Ei) that is not an elementary function.

Variational EM Algorithm: E step

- Therefore, the Q function can be derived as (up to a constant):

$$Q(\boldsymbol{\Omega}, \boldsymbol{\beta} | \boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)}) = \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{X}^{(k)} \sim \mathbb{P}(\cdot | \boldsymbol{y}^{(k)}, \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)})} \log \mathbb{P}(\boldsymbol{X}^{(k)} | \boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)}) + \mathbb{E}_{\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot | \boldsymbol{\Omega}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{\Omega} | \boldsymbol{\Gamma}).$$

- However, this Q function still lacks an explicit form, because the normalizing constant has the expression of

$$\mathbb{P}(\boldsymbol{y}_i^{(k)} | \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)}) = \int \prod_{j=1}^{p} \mathrm{Pois}(y_{ij}^{(k)}; \exp(-e^{X_{ij}^{(k)}})) \cdot N(\boldsymbol{X}_i^{(k)}; \boldsymbol{\lambda}_i^{(k,t)}, (\boldsymbol{\Omega}^{(k,t)})^{-1}) d\boldsymbol{X}_i^{(k)},$$

  which includes an Exponential integral (Ei) that is not an elementary function.

Variational EM Algorithm: Variational step

- We adopt a variational approximation of $\boldsymbol{X}$:

$$\mathcal{P}_{VI} = \left\{ P(\boldsymbol{X}) = \prod_{k=1}^{K} \prod_{i=1}^{n_k} \prod_{j=1}^{p} N(X_{ij}^{(k)}; \mu_{ij}^{(k)}, \sigma_{ij}^{2(k)}) \right\}.$$

- The variational posterior of $\boldsymbol{X}$ is defined as

$$\widetilde{\Pi}^{(\cdot,t)} := \underset{P \in \mathcal{P}_{VI}}{\arg\min} \quad D_{\mathrm{KL}}(P(\boldsymbol{X}) \| \mathbb{P}(\boldsymbol{X} | \boldsymbol{y}, \boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)})),$$

where $D_{\mathrm{KL}}(\cdot \| \cdot)$ is the KL divergence.

Variational EM Algorithm: Variational step

- We adopt a variational approximation of $\boldsymbol{X}$:

$$\mathcal{P}_{VI} = \left\{ P(\boldsymbol{X}) = \prod_{k=1}^{K} \prod_{i=1}^{n_k} \prod_{j=1}^{p} N(X_{ij}^{(k)}; \mu_{ij}^{(k)}, \sigma_{ij}^{2(k)}) \right\}.$$

- The variational posterior of $\boldsymbol{X}$ is defined as

$$\widetilde{\Pi}^{(\cdot,t)} := \underset{P \in \mathcal{P}_{VI}}{\arg\min} \quad D_{\mathrm{KL}}(P(\boldsymbol{X}) \| \mathbb{P}(\boldsymbol{X} | \boldsymbol{y}, \boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)})),$$

  where $D_{\mathrm{KL}}(\cdot\|\cdot)$ is the KL divergence.

Variational EM Algorithm: E and M step

- Replacing the posterior $\mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{y}^{(k)}, \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)})$ in (11) by its variational surrogate $\widetilde{\Pi}^{(k,t)}$, we can derive an approximation of Q function:

$$\widetilde{Q}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)}) = \underbrace{\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{X}^{(k)} \sim \widetilde{\Pi}^{(k,t)}} \log \mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)})}_{\text{likelihood}} + \underbrace{\mathbb{E}_{\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot|\boldsymbol{\Omega}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{\Omega}|\boldsymbol{\Gamma})}_{\text{prior}},$$

where

$$\log \mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)}) = \frac{n_k}{2} \log |\boldsymbol{\Omega}^{(k)}| - \frac{1}{2} \sum_{i=1}^{n_k} (\boldsymbol{X}_i^{(k)} - \boldsymbol{\lambda}_i^{(k)})^\intercal \boldsymbol{\Omega}^{(k)} (\boldsymbol{X}_i^{(k)} - \boldsymbol{\lambda}_i^{(k)}).$$

- The M-step of the above variational EM algorithm solves $(\boldsymbol{\Omega}^{(\cdot,t+1)}, \boldsymbol{\beta}^{(\cdot,t+1)}) = \arg\max_{(\boldsymbol{\Omega},\boldsymbol{\beta})} \widetilde{Q}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)})$.
  This process is repeated until convergence.

Variational EM Algorithm: E and M step

- Replacing the posterior $\mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{y}^{(k)}, \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)})$ in (11) by its variational surrogate $\widetilde{\Pi}^{(k,t)}$, we can derive an approximation of Q function:

$$\widetilde{Q}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)}) = \underbrace{\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{X}^{(k)} \sim \widetilde{\Pi}^{(k,t)}} \log \mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)})}_{\text{likelihood}} + \underbrace{\mathbb{E}_{\boldsymbol{\Gamma} \sim \mathbb{P}(\cdot|\boldsymbol{\Omega}^{(\cdot,t)})} \log \mathbb{P}(\boldsymbol{\Omega}|\boldsymbol{\Gamma})}_{\text{prior}},$$

  where

$$\log \mathbb{P}(\boldsymbol{X}^{(k)}|\boldsymbol{\Omega}^{(k)}, \boldsymbol{\beta}^{(k)}) = \frac{n_k}{2} \log |\boldsymbol{\Omega}^{(k)}| - \frac{1}{2} \sum_{i=1}^{n_k} (\boldsymbol{X}_i^{(k)} - \boldsymbol{\lambda}_i^{(k)})^\intercal \boldsymbol{\Omega}^{(k)} (\boldsymbol{X}_i^{(k)} - \boldsymbol{\lambda}_i^{(k)}).$$

- The M-step of the above variational EM algorithm solves $(\boldsymbol{\Omega}^{(\cdot,t+1)}, \boldsymbol{\beta}^{(\cdot,t+1)}) = \arg\max_{(\boldsymbol{\Omega},\boldsymbol{\beta})} \widetilde{Q}(\boldsymbol{\Omega}, \boldsymbol{\beta}|\boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)})$.
  This process is repeated until convergence.

Solve VEM: E step

- The conditional posterior for γ_{ij} can be calculated as:

$$\mathbb{P}(\gamma_{ij} = 1 | \mathbf{\Omega}^{(\cdot,t)}) = 1/\left(1 + \frac{1-p_0}{p_0}(\frac{v_1}{v_0})^K \exp\left(-(\frac{1}{v_0} - \frac{1}{v_1})\sum_{k=1}^{K} |\Omega_{ij}^{(k,t)}|\right)\right).$$

- Denoting this probability as $p_{ij}(\mathbf{\Omega}^{(\cdot,t)})$, the prior term is given by:

$$\mathbb{E}_{\mathbf{\Gamma} \sim \mathbb{P}(\cdot|\mathbf{\Omega}^{(\cdot,t)})} \log \mathbb{P}(\mathbf{\Omega}|\mathbf{\Gamma}) = \sum_{i<j}\left(\frac{p_{ij}(\mathbf{\Omega}^{(\cdot,t)})}{v_1} + \frac{1-p_{ij}(\mathbf{\Omega}^{(\cdot,t)})}{v_0}\right)|\Omega_{ij}| + \sum_{i=1}^{p}\frac{1}{\tau}\Omega_{ii},$$

which is a weighted graphical lasso penalty.

Solve VEM: E step

- The conditional posterior for γ_{ij} can be calculated as:

$$\mathbb{P}(\gamma_{ij} = 1|\mathbf{\Omega}^{(\cdot,t)}) = 1/\left(1 + \frac{1-p_0}{p_0}(\frac{v_1}{v_0})^K \exp\left(-(\frac{1}{v_0} - \frac{1}{v_1})\sum_{k=1}^{K}|\Omega_{ij}^{(k,t)}|\right)\right).$$

- Denoting this probability as $p_{ij}(\mathbf{\Omega}^{(\cdot,t)})$, the prior term is given by:

$$\mathbb{E}_{\mathbf{\Gamma}\sim\mathbb{P}(\cdot|\mathbf{\Omega}^{(\cdot,t)})} \log \mathbb{P}(\mathbf{\Omega}|\mathbf{\Gamma}) = \sum_{i<j}\left(\frac{p_{ij}(\mathbf{\Omega}^{(\cdot,t)})}{v_1} + \frac{1-p_{ij}(\mathbf{\Omega}^{(\cdot,t)})}{v_0}\right)|\Omega_{ij}| + \sum_{i=1}^{p}\frac{1}{\tau}\Omega_{ii},$$

which is a weighted graphical lasso penalty.

Solve VEM: Variational step

- Defining $\boldsymbol{\mu}_i^{(k)} = (\mu_{i1}^{(k)}, \mu_{i2}^{(k)}, \cdots, \mu_{ip}^{(k)})^\intercal$ and $\boldsymbol{\sigma}_i^{2(k)} = (\sigma_{i1}^{2(k)}, \sigma_{i2}^{2(k)}, \cdots, \sigma_{ip}^{2(k)})^\intercal$, the target function in (12) is:

$$D_{\mathrm{KL}}(P(\boldsymbol{X}) \| \mathbb{P}(\boldsymbol{X}|\boldsymbol{y}, \boldsymbol{\Omega}^{(\cdot,t)}, \boldsymbol{\beta}^{(\cdot,t)})) = \sum_{k=1}^{K} \sum_{i=1}^{n_k} D_{\mathrm{KL}}(P(\boldsymbol{X}_i^{(k)}) \| \mathbb{P}(\boldsymbol{X}_i^{(k)}|\boldsymbol{y}_i^{(k)}, \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)})),$$

where

$$\begin{aligned} D_{\mathrm{KL}}(P(\boldsymbol{X}_i^{(k)}) \| \mathbb{P}(\boldsymbol{X}_i^{(k)}|\boldsymbol{y}_i^{(k)}, \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\beta}^{(k,t)})) &= \frac{1}{2}(\boldsymbol{\mu}_i^{(k)} - \boldsymbol{\lambda}_i^{(k,t)})^\intercal \boldsymbol{\Omega}^{(k,t)} (\boldsymbol{\mu}_i^{(k)} - \boldsymbol{\lambda}_i^{(k,t)}) \\ &\quad - \boldsymbol{y}_i^{(k)\intercal} \boldsymbol{\mu}_i^{(k)} + \frac{1}{2} \sum_{j=1}^{p} \left(\sigma_{ij}^{2(k)} \Omega_{jj}^{(k,t)} + 2 \exp(\mu_{ij}^{(k)} + \frac{\sigma_{ij}^{2(k)}}{2}) - \log \sigma_{ij}^{2(k)} \right). \end{aligned}$$

Solve VEM: ADMM

- Solving $\boldsymbol{\sigma}_i^{2(k)}$ is straightforward but solving $\boldsymbol{\mu}_i^{(k)}$ is not.
- The optimization problem for the variational mean vector $\boldsymbol{\mu}_i^{(k,t)}$ can be reformulated as

$$\min_{\boldsymbol{\mu}_N, \boldsymbol{\mu}_M} \ell_1(\boldsymbol{\mu}_N, \boldsymbol{\Omega}^{(k,t)}, \boldsymbol{\lambda}_i^{(k,t)}) + \sum_{j=1}^{p} \ell_2(\mu_{Mj}, y_{ij}^{(k)}, \sigma_{ij}^{(k,t-1)})$$

$$\text{subject to } \boldsymbol{\mu}_N = \boldsymbol{\mu}_M,$$

where

$$\ell_1(\boldsymbol{\mu}, \boldsymbol{\Omega}, \boldsymbol{\lambda}) = \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\lambda})^\intercal \boldsymbol{\Omega}(\boldsymbol{\mu} - \boldsymbol{\lambda}),$$

and

$$\ell_2(\mu, y, \sigma^2) = -\mu y + \exp\left(\mu + \frac{\sigma^2}{2}\right).$$

Solve VEM: ADMM

- The ADMM algorithm finds a global minimum by optimizing the augmented Lagrangian

$$\ell^{(k)}(\boldsymbol{\mu}_N, \boldsymbol{\mu}_M) + \boldsymbol{\alpha}^\intercal (\boldsymbol{\mu}_N - \boldsymbol{\mu}_M) + \frac{\rho}{2}\|\boldsymbol{\mu}_N - \boldsymbol{\mu}_M\|^2,$$

  with respect to $\boldsymbol{\mu}_N$, $\boldsymbol{\mu}_M$ and $\boldsymbol{\alpha}$ in a coordinate-wise manner.

- Algorithm:

2: **for** $j = 1$ to p **do**

3:

$$\mu_{M,j}^{(\tilde{t}+1)} = \underset{\mu_{M,j}}{\arg\min} \left\{ \ell_2(\mu_{M,j}, y_{ij}^{(k)}, \sigma_{ij}^{2(k,t-1)}) + \alpha_i^{(\tilde{t})}(\mu_{M,j} - \mu_{N,j}^{(\tilde{t})}) + \frac{\rho}{2}(\mu_{M,j} - \mu_{N,j}^{(\tilde{t})})^2 \right\}.$$

4: **end for**

5: Update

$$\boldsymbol{\mu}_N^{(\tilde{t}+1)} = \left(\boldsymbol{\Omega}^{(k,t)} + \rho \boldsymbol{I}\right)^{-1} \left(\rho \boldsymbol{\mu}_M^{(\tilde{t}+1)} + \boldsymbol{\alpha}^{(\tilde{t})} + \boldsymbol{\Omega}^{(k,t)} \boldsymbol{\lambda}_i^{(k,t)}\right).$$

6: Update

$$\boldsymbol{\alpha}^{(\tilde{t}+1)} = \boldsymbol{\alpha}^{(\tilde{t})} + \rho \left(\boldsymbol{\mu}_M^{(\tilde{t}+1)} - \boldsymbol{\mu}_N^{(\tilde{t}+1)}\right).$$

Parallel Computing

- Within each sample, the updates of $\mu_{Mj}, \quad j = 1, 2, \ldots, p$ do not interfere.
- Similar to $\sigma^2_{ij}, \quad j = 1, 2, \ldots, p$.
- Different samples within a group are independent.
- K precision matrices only share the penalty matrix $\boldsymbol{P}$.

```
@cython.boundscheck(False)
@cython.wraparound(False)
def update_py_par(double[:, :] mu, double[:, :] y
    cdef int p = y.shape[1]
    cdef int n = y.shape[0]
    cdef Py_ssize_t i

    if threads == 0:
        for i in prange(n, nogil=True):
            update_mu(&mu[i,0], &y[i,0], &sigma[i
        for i in prange(n, nogil=True):
            update_sigma(&sigma[i,0], &Omega[0,0]
```

Figure: Cython code for parallel computing

Parallel Computing

- Within each sample, the updates of $\mu_{Mj}, \quad j = 1, 2, \ldots, p$ do not interfere.
- Similar to $\sigma^2_{ij}, \quad j = 1, 2, \ldots, p$.
- Different samples within a group are independent.
- K precision matrices only share the penalty matrix $\boldsymbol{P}$.

```
@cython.boundscheck(False)
@cython.wraparound(False)
def update_py_par(double[:, :] mu, double[:, :] y
    cdef int p = y.shape[1]
    cdef int n = y.shape[0]
    cdef Py_ssize_t i

    if threads == 0:
        for i in prange(n, nogil=True):
            update_mu(&mu[i,0], &y[i,0], &sigma[i
        for i in prange(n, nogil=True):
            update_sigma(&sigma[i,0], &Omega[0,0]
```

Figure: Cython code for parallel computing

Parallel Computing

- Within each sample, the updates of $\mu_{Mj}, \quad j = 1, 2, \ldots, p$ do not interfere.
- Similar to $\sigma^2_{ij}, \quad j = 1, 2, \ldots, p$.
- Different samples within a group are independent.
- K precision matrices only share the penalty matrix $\boldsymbol{P}$.

```
@cython.boundscheck(False)
@cython.wraparound(False)
def update_py_par(double[:, :] mu, double[:, :] y
    cdef int p = y.shape[1]
    cdef int n = y.shape[0]
    cdef Py_ssize_t i

    if threads == 0:
        for i in prange(n, nogil=True):
            update_mu(&mu[i,0], &y[i,0], &sigma[i
        for i in prange(n, nogil=True):
            update_sigma(&sigma[i,0], &Omega[0,0]
```

Figure: Cython code for parallel computing

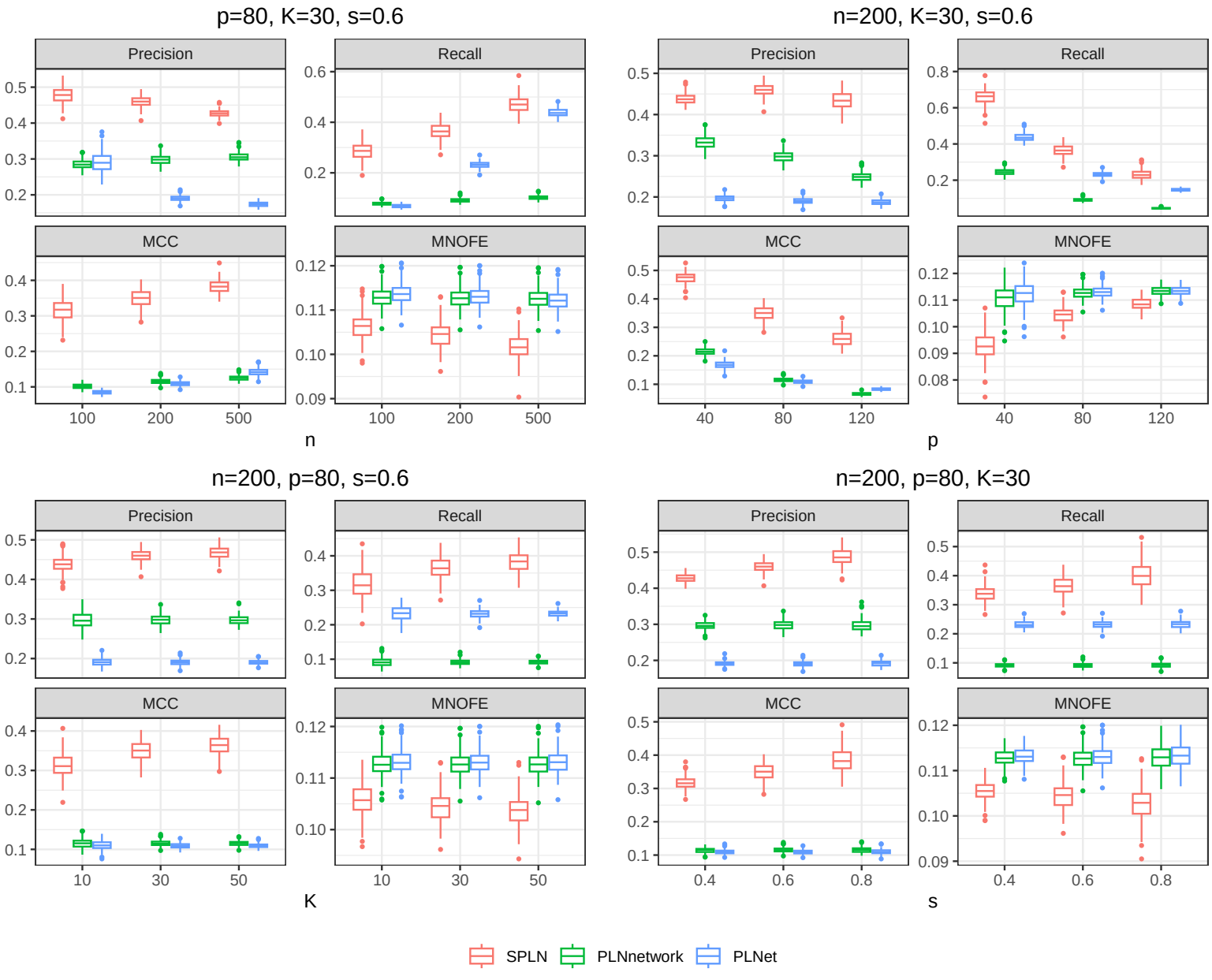
Parallel Computing

- Within each sample, the updates of $\mu_{Mj}, \quad j = 1, 2, \ldots, p$ do not interfere.
- Similar to $\sigma^2_{ij}, \quad j = 1, 2, \ldots, p$.
- Different samples within a group are independent.
- K precision matrices only share the penalty matrix $\boldsymbol{P}$.

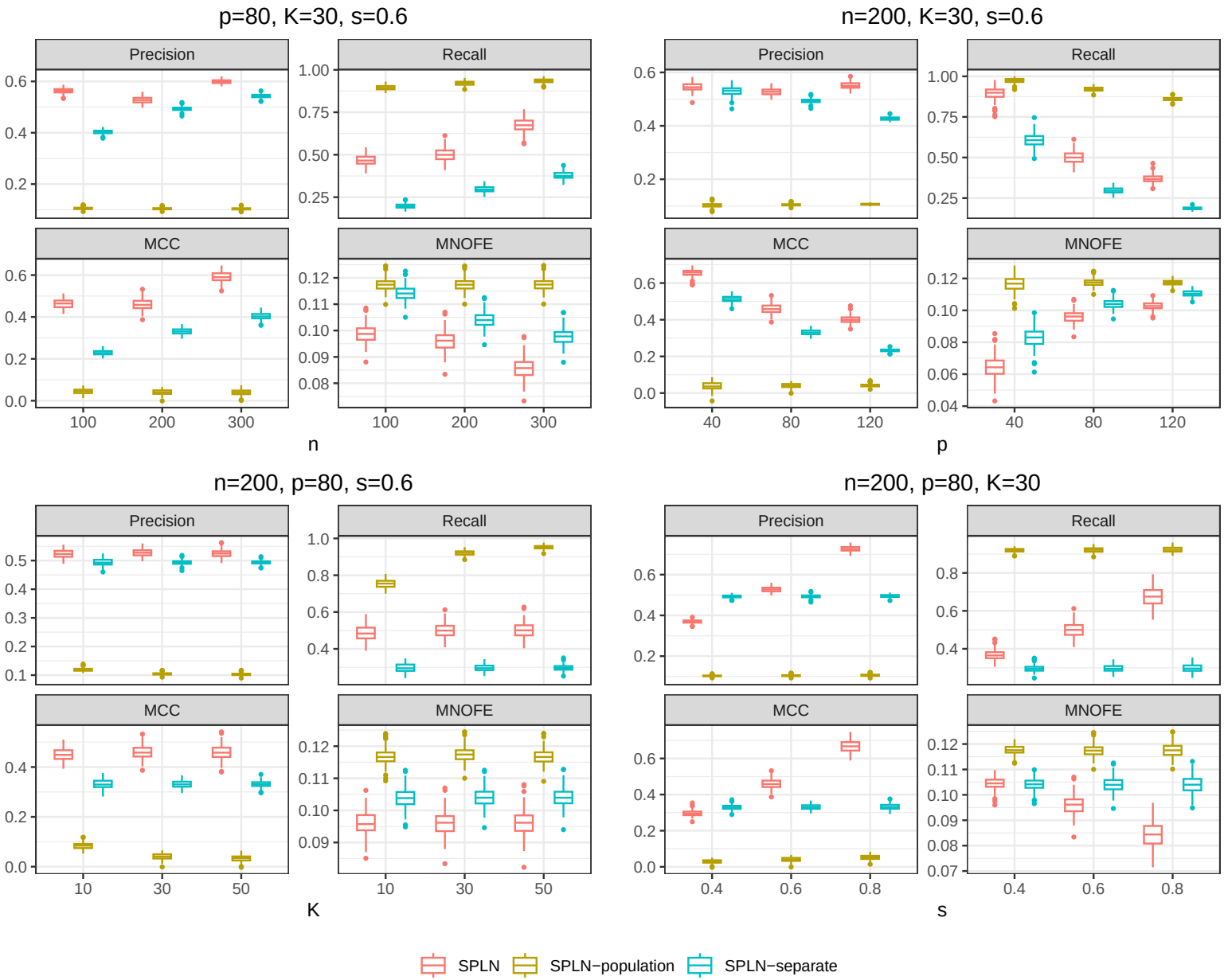
```
@cython.boundscheck(False)
@cython.wraparound(False)
def update_py_par(double[:, :] mu, double[:, :] y
    cdef int p = y.shape[1]
    cdef int n = y.shape[0]
    cdef Py_ssize_t i

    if threads == 0:
        for i in prange(n, nogil=True):
            update_mu(&mu[i,0], &y[i,0], &sigma[i
        for i in prange(n, nogil=True):
            update_sigma(&sigma[i,0], &Omega[0,0]
```

Figure: Cython code for parallel computing

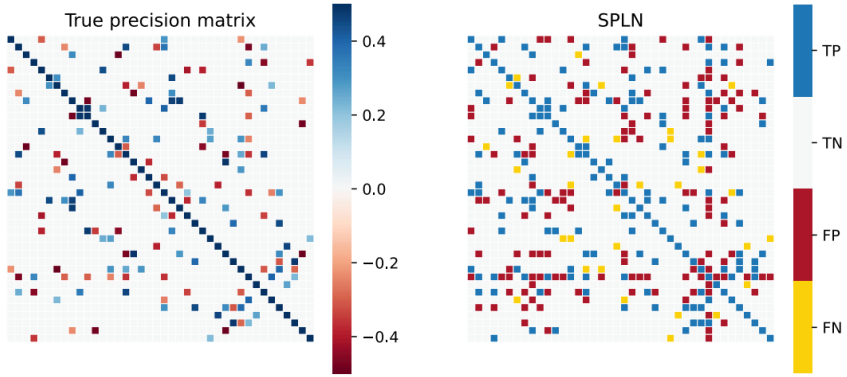
Simulation Study (Existing Methods)

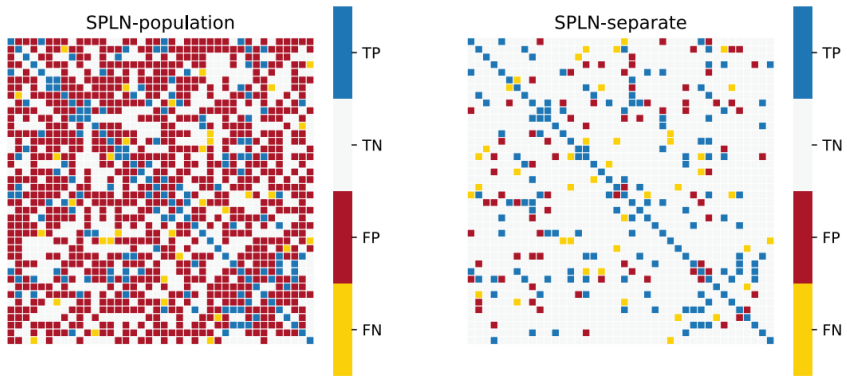


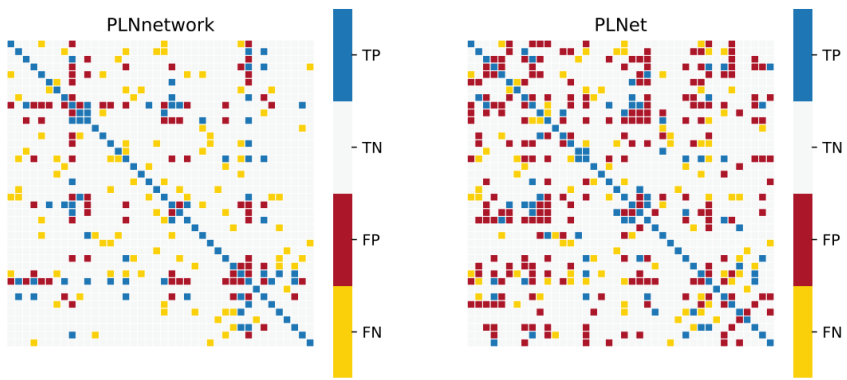
Simulation Study (Separate Estimation)



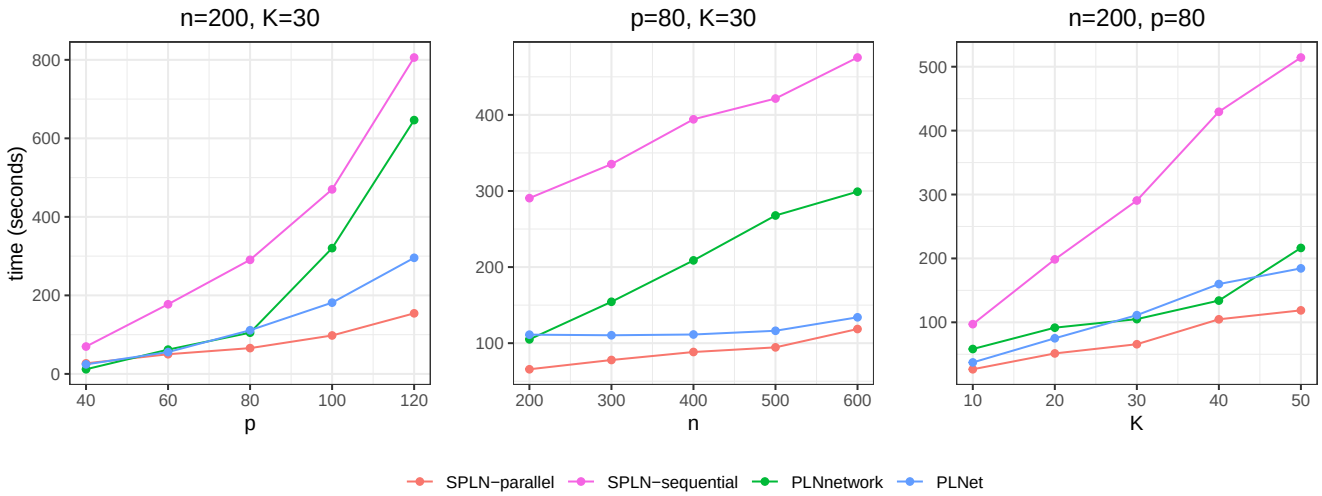
Simulation Study







Simulation Study (Running Time)



Yeast Single-cell Data Analysis

- $p = 612$ genes across $K = 11$ different growth conditions, a total of 38,285 cells (Jackson et al., 2020)

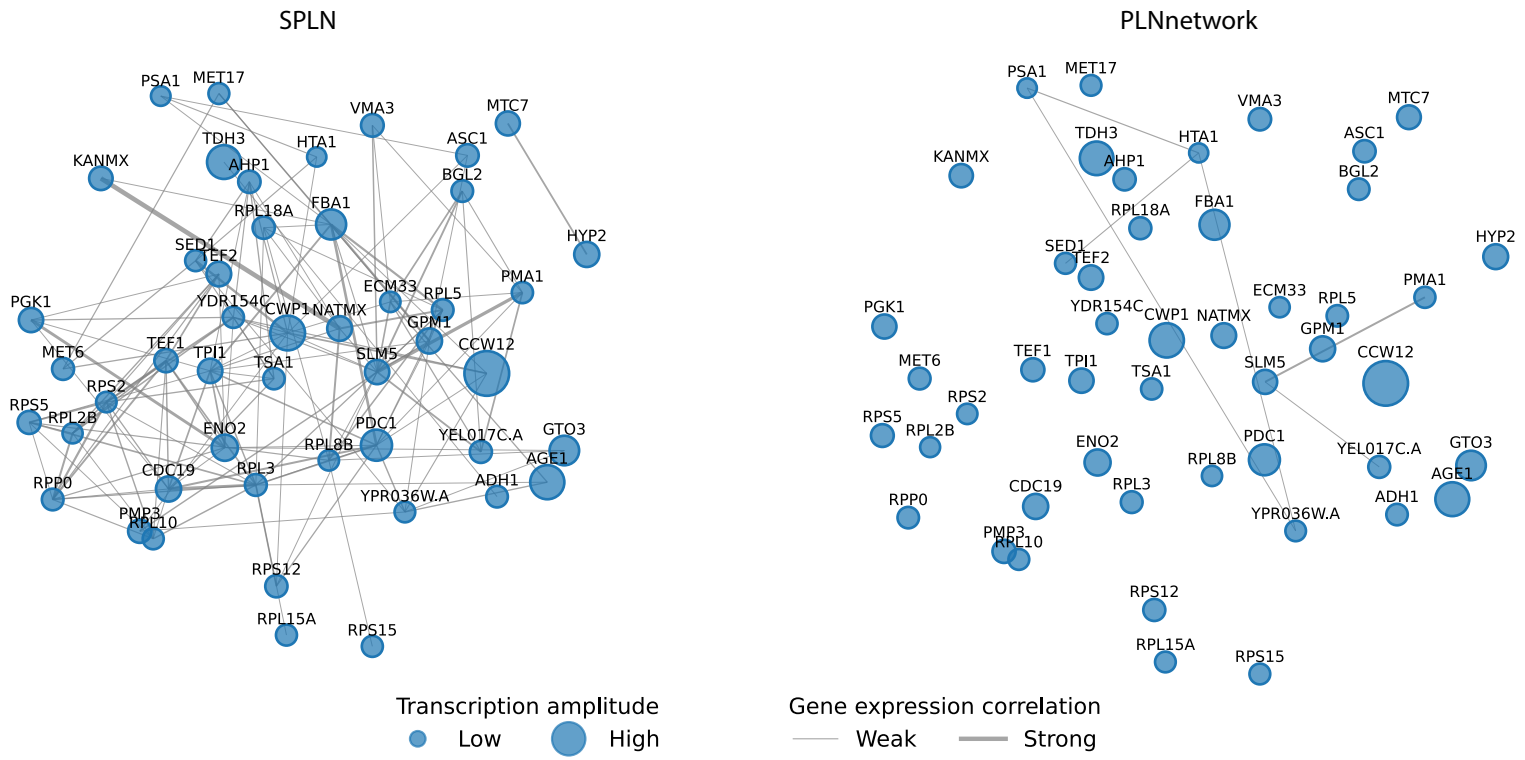


Figure: Gene co-expression network estimation of Yeast gene expression data under AmmoniumSulfate condition.

Yeast Single-cell Data Analysis

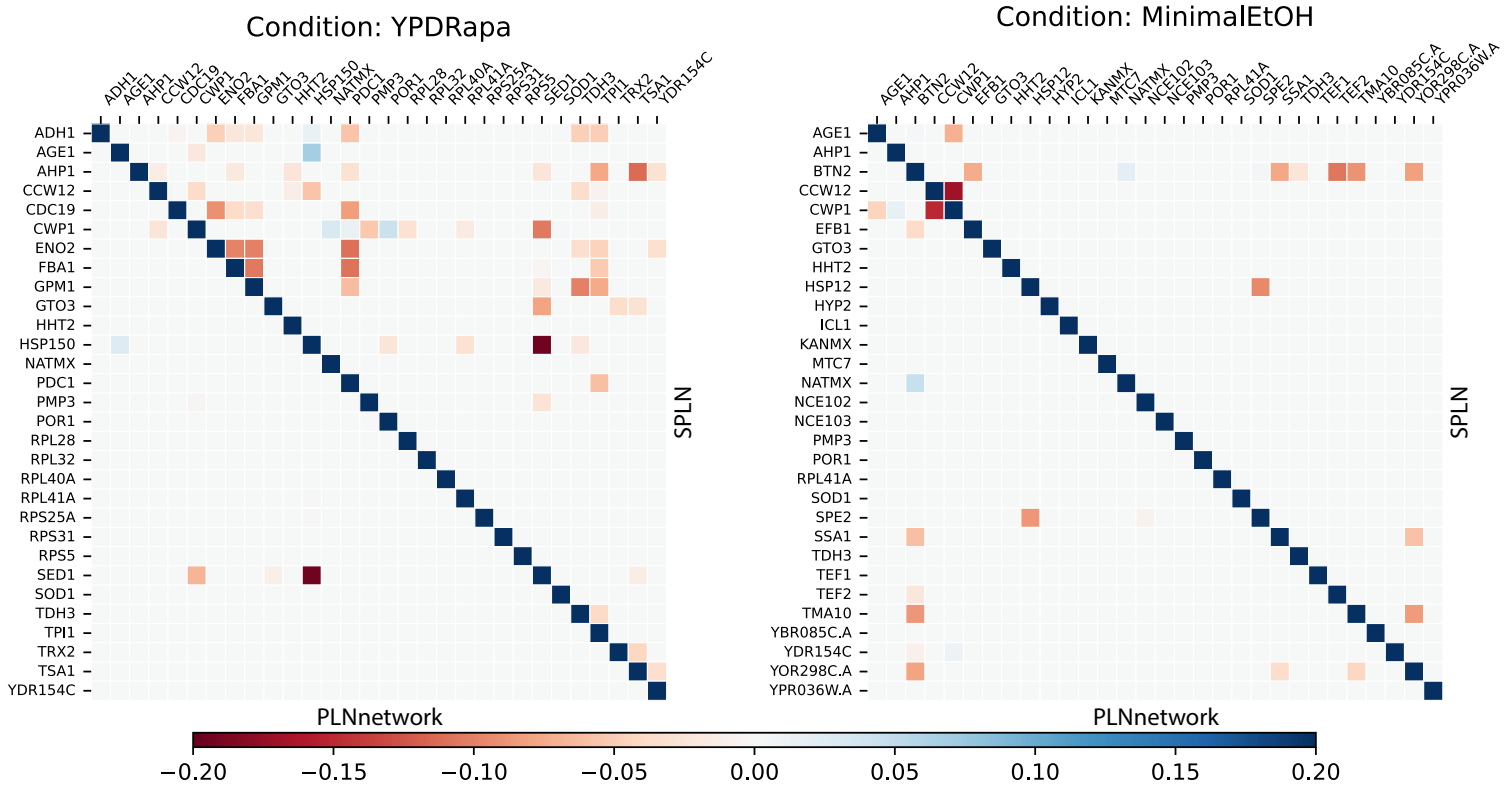
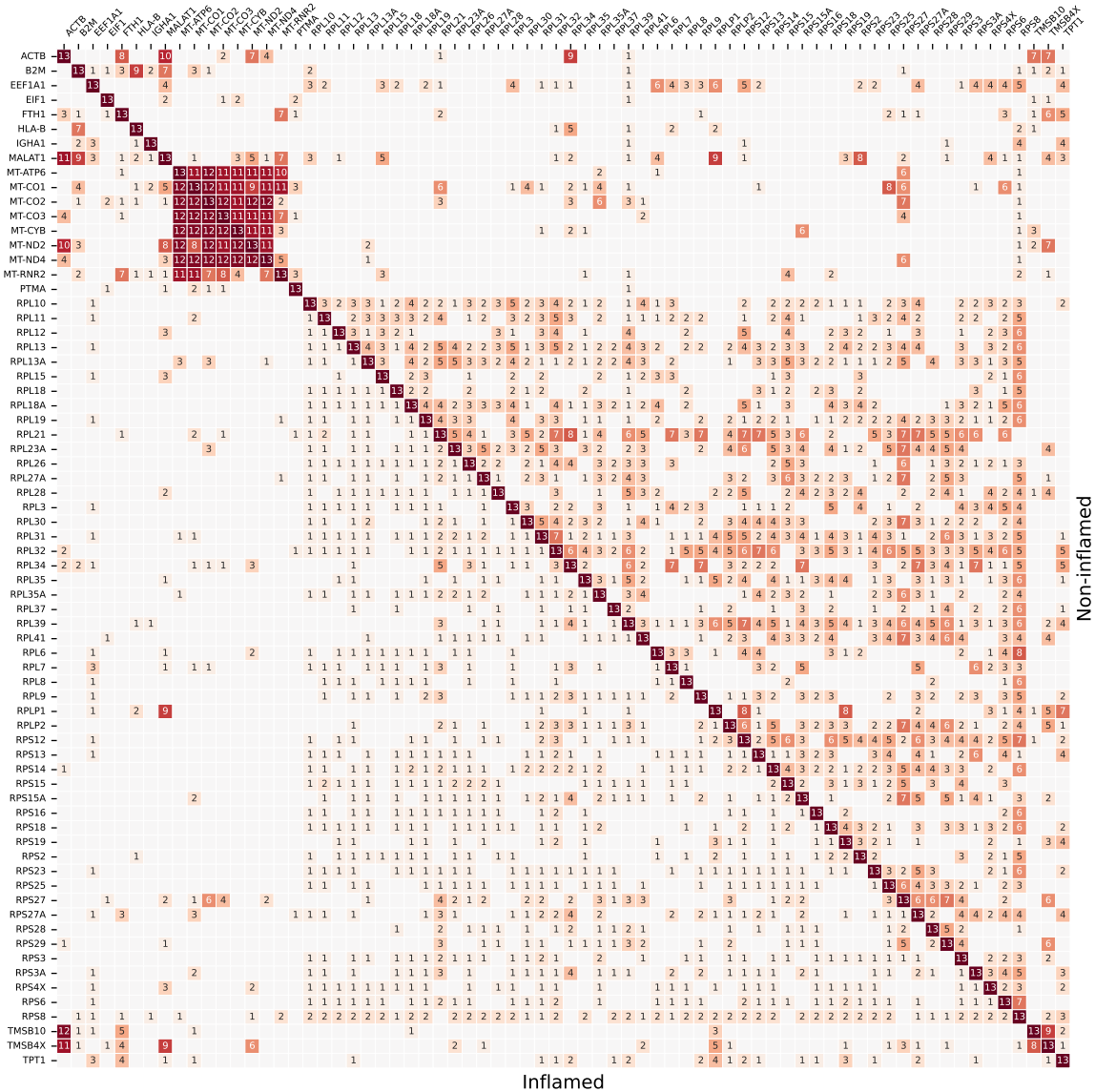


Figure: Precision matrix extimation of Yeast gene expression data under two conditions.

T-cells in Patients with Ulcerative Colitis (SPLN)

- $K = 13$ inflamed and $K = 13$ non-inflamed colon tissues, p =368 genes (Smillie et al., 2019)



T-cells in Patients with Ulcerative Colitis (PLNnetwork)

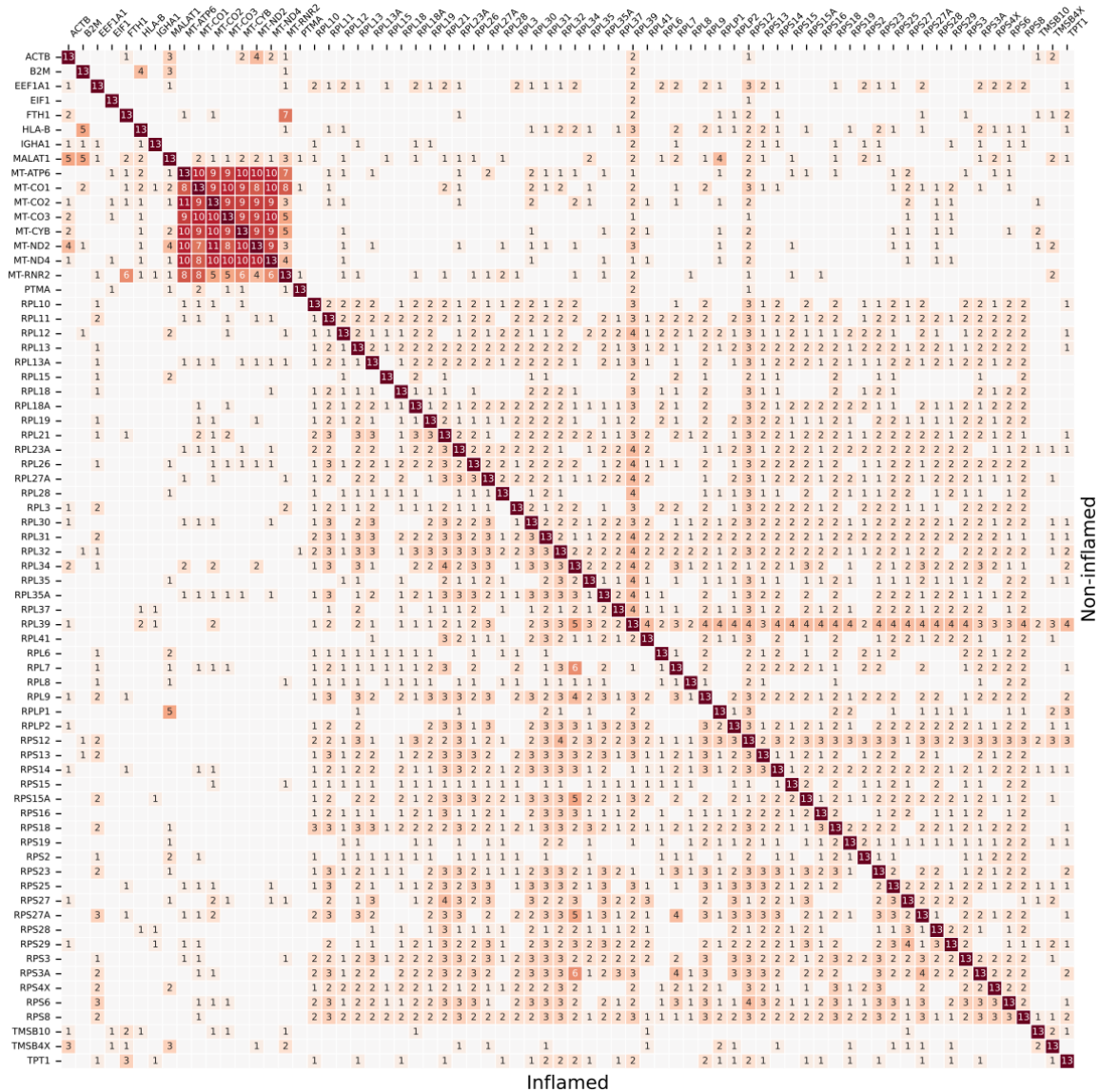


Table of Contents

Introduction

Estimation of many gene expression networks

CNV prediction with Neural Network

Future projects - Statistical and Computational Methods for Perturb Seq Data

Progress in publication

Ge C, He X, Zhang L and Li H (2025): RCANE: A Deep Learning Algorithm for Whole-genome Pan-Cancer Somatic Copy Number Aberration Prediction using RNA-seq Data. *Communications Biology*, in revision.
https://www.biorxiv.org/content/10.1101/2024.11.03.621681v1.full

Copy Number Variation in Cancer Tissues

- Copy number variation (CNV) is a phenomenon in which sections of the genome are repeated and the number of repeats in the genome varies between individuals.
- CNV involves large-scale genomic alterations that drive tumorigenesis and cancer progression by affecting gene dosage and altering the expression of oncogenes and tumor suppressor genes.

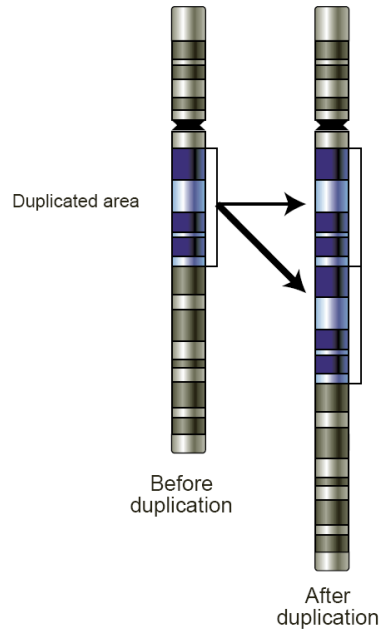


Figure: Illustration of Copy Number Duplication

Data sets: 11,000 TCGA samples of 33 cancer types and 410 DepMap cell lines, with both RNA-seq data and SNP-array-based $log_2 R$ data (copy number intensity).

Neural Network Architecture

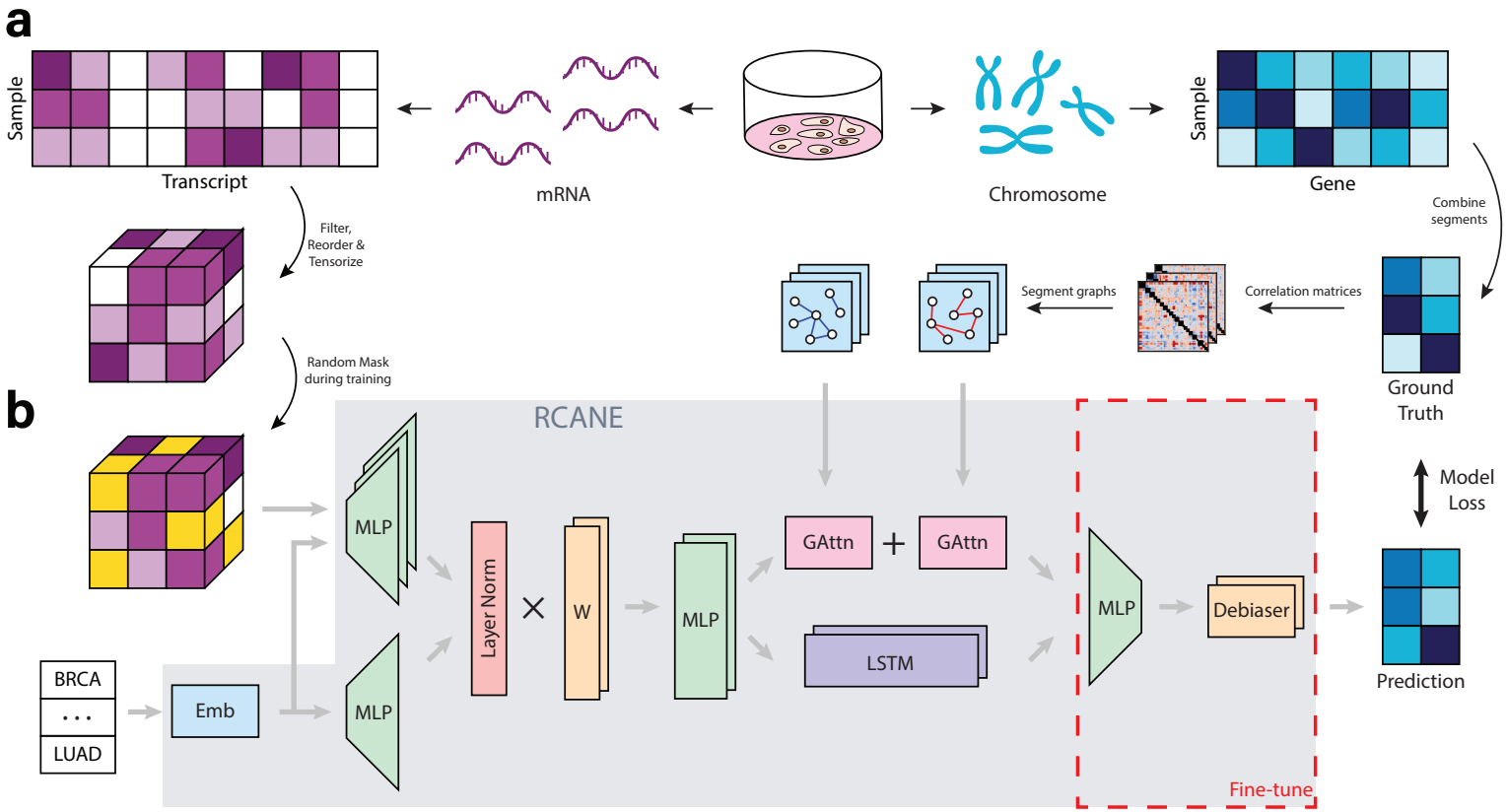


Figure: Overview of RCANE architecture

TCGA Results on Testing Samples - All genes/segments

- 33 cancer types, 30,096 transcripts, 1,514 segments, 2,226 testing samples (Weinstein et al., 2013)

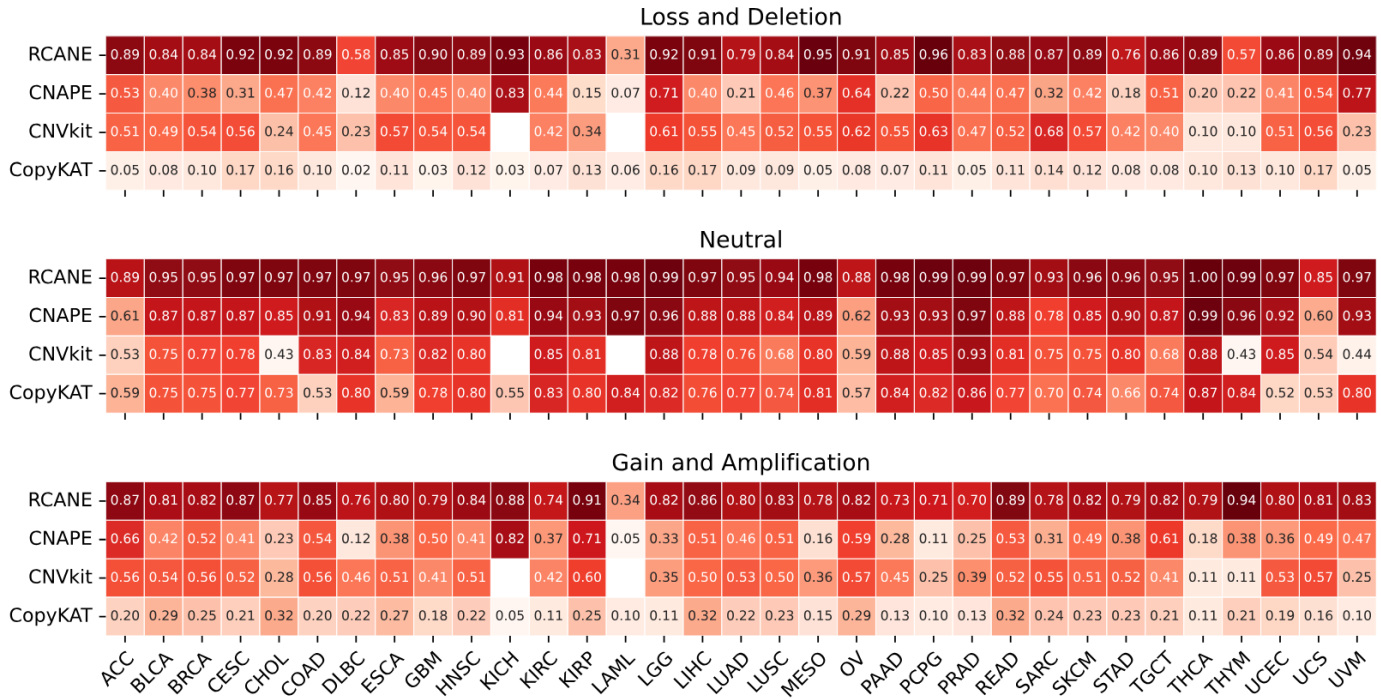


Figure: F1 score of TCGA data for each cancer type

TCGA Results on Testing Samples - All genes/segments

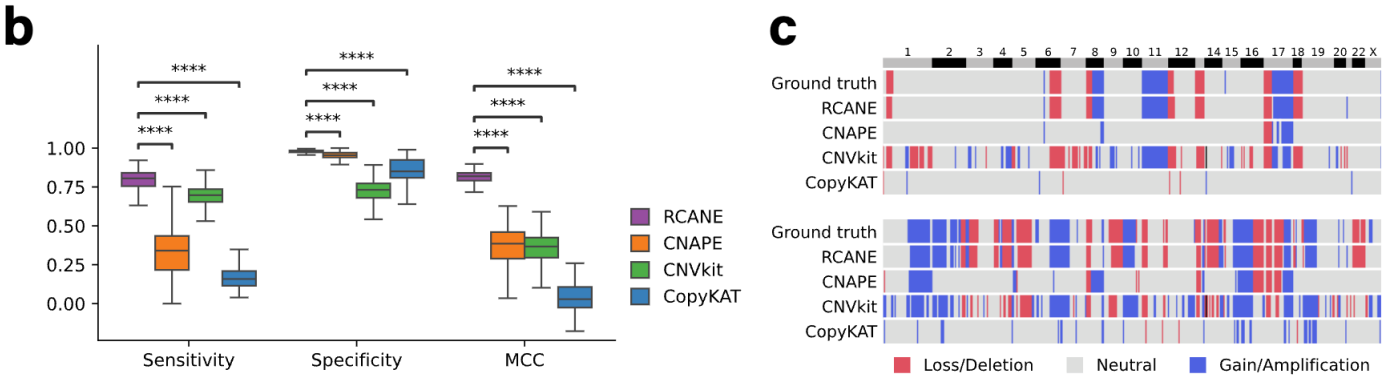


Figure: MCC and whole-genome prediction results of TCGA data

TCGA Results - interpretability

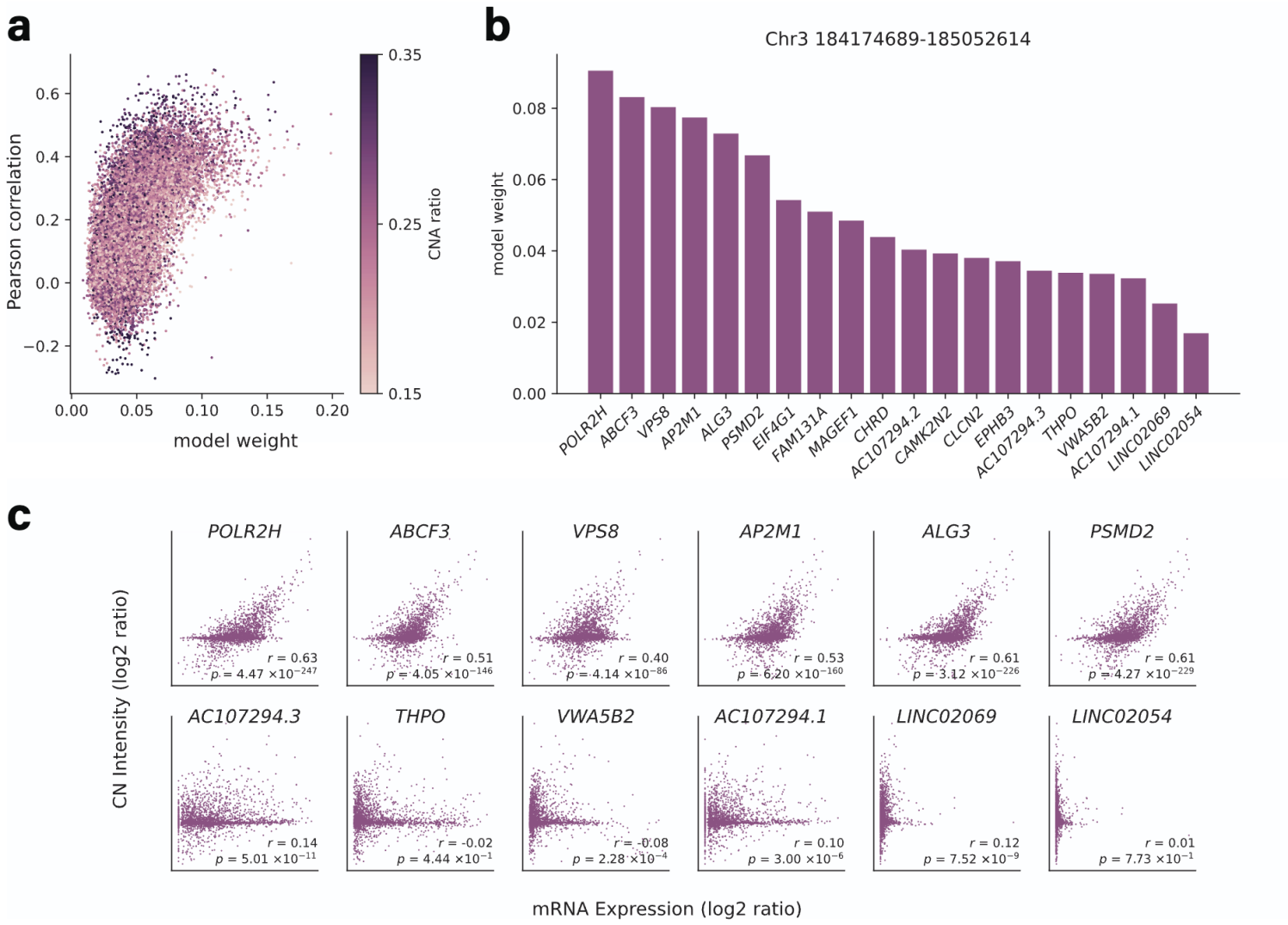


Figure: Model weights in RCANE capture CNA-RNA correlations

Table of Contents

Introduction

Estimation of many gene expression networks

CNV prediction with Neural Network

Future projects - Statistical and Computational Methods for Perturb Seq Data

Statistical Methods for Perturb-Seq Data

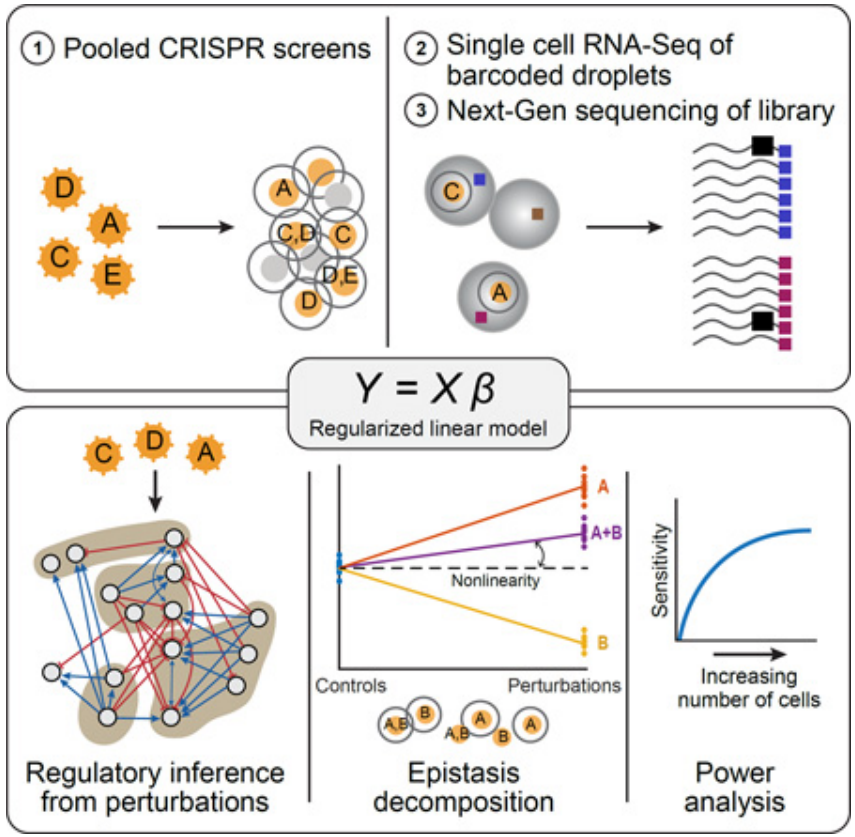


Figure: Perturb-seq data (Replogle et al., 2022)

Perturb-Seq Data

```
1  adata.obs.gene
✓ 0.0s
cell_barcode
AAACCCAAGAAACTAC-53           MRPS31
AAACCCAAGAAGCCAC-51         LRRC37A3
AAACCCAAGAAGCGAA-32            SRCAP
AAACCCAAGAATACAC-44             WBP1
AAACCCAAGAATCGAT-43            RRP12
                              ...
TTTGTTGTCTGCACCT-44              MAX
TTTGTTGTCTGGGCAC-32           ATP6V0C
TTTGTTGTCTGTCCCA-37             MLST8
TTTGTTGTCTTGGTCC-17              CDK6
TTTGTTGTCTTTCCGG-22     non-targeting
Name: gene, Length: 247914, dtype: category
Categories (2394, object): ['AAAS', 'AAMP',
```

Figure: Interfered genes of each cell

```
1  adata.X
✓ 0.0s
array([[  4.,   1.,   5., ...,  45.,  31., 131.],
       [  0.,   0.,   3., ...,  13.,   7.,  55.],
       [  0.,   0.,   2., ...,  27.,   4.,  72.],
       ...,
       [  0.,   0.,   0., ...,  14.,   1.,  57.],
       [  0.,   0.,   0., ...,  10.,   0.,  53.],
       [  4.,   0.,   0., ...,  38.,   6.,  85.]],
```

Figure: Raw read counts

Causal Inference and DAGs

- Linear SEM for observational data:

$$\boldsymbol{X} \leftarrow \boldsymbol{X}\boldsymbol{W} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{W} \in \mathcal{D} \subset \mathbb{R}^{p \times p}$, $\mathcal{D}$ is the set of weighted adjacency matrices on p nodes whose support is a directed acyclic graph (DAG).

- Low multiplicity of infection (MOI, Barry et al. (2024)):

$$\boldsymbol{X}^{(k)} \leftarrow \boldsymbol{X}^{(k)}\boldsymbol{W}^{(k)} + \boldsymbol{\epsilon}^{(k)}, \qquad k = 0, 1, \cdots, p, \tag{2}$$

where $\boldsymbol{X}^{(k)} \in \mathbb{R}^{n_k \times p}$ represents n_k observations of X when the k-th variable is intervened,

$$W_{ij}^{(k)} := \begin{cases} W_{ij}, & j \neq k \\ 0, & j = k \end{cases}. \tag{3}$$

Causal Inference and DAGs

- Linear SEM for observational data:

$$\boldsymbol{X} \leftarrow \boldsymbol{X}\boldsymbol{W} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{W} \in \mathcal{D} \subset \mathbb{R}^{p \times p}$, $\mathcal{D}$ is the set of weighted adjacency matrices on p nodes whose support is a directed acyclic graph (DAG).

- Low multiplicity of infection (MOI, Barry et al. (2024)):

$$\boldsymbol{X}^{(k)} \leftarrow \boldsymbol{X}^{(k)}\boldsymbol{W}^{(k)} + \boldsymbol{\epsilon}^{(k)}, \qquad k = 0, 1, \cdots, p, \tag{2}$$

where $\boldsymbol{X}^{(k)} \in \mathbb{R}^{n_k \times p}$ represents n_k observations of X when the k-th variable is intervened,

$$W_{ij}^{(k)} := \begin{cases} W_{ij}, & j \neq k \\ 0, & j = k \end{cases}. \tag{3}$$

Causal Inference and DAGs

Definition (d-separation, Nandy et al. (2018))

Let $\boldsymbol{S}$ be a subset of nodes in a DAG $\mathcal{H}$, where $\boldsymbol{S}$ does not contain X_i and X_k. Then $\boldsymbol{S}$ *blocks* a path $\pi_{\mathcal{H}}(X_i, \ldots, X_k)$ if at least one of the following holds: (i) $\pi_{\mathcal{H}}$ contains a noncollider that is in $\boldsymbol{S}$, or (ii) $\pi_{\mathcal{H}}$ contains a collider that has no descendant in $\boldsymbol{S}$. Otherwise $\pi_{\mathcal{H}}$ is *open* given $\boldsymbol{S}$. For pairwise disjoint sets of nodes $\boldsymbol{W}_1, \boldsymbol{W}_2$ and $\boldsymbol{S}$, we say that $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are *d-separated* by $\boldsymbol{S}$ if every path between a node in $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ is blocked by $\boldsymbol{S}$. This is denoted by $\boldsymbol{W}_1 \perp_{\mathcal{H}} \boldsymbol{W}_2 | \boldsymbol{S}$. Otherwise, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are *d-connected* given $\boldsymbol{S}$ in $\mathcal{H}$, denoted by $\boldsymbol{W}_1 \not\perp_{\mathcal{H}} \boldsymbol{W}_2 | \boldsymbol{S}$.

Figure: r blocks x and s.

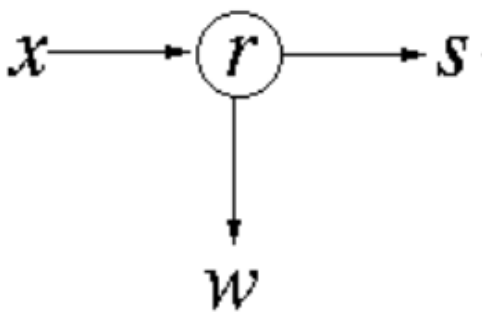
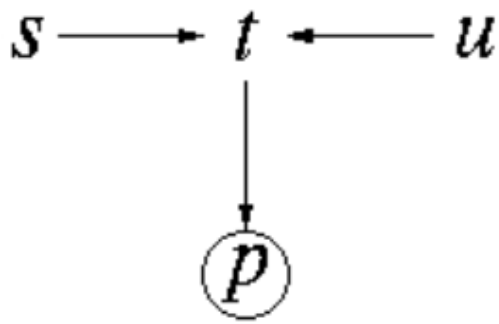


Figure: p does NOT block s and u.



Causal Inference and DAGs

- The causal DAG is typically not identifiable from observational data from the distribution of X.
- Several DAGs can encode the same conditional independence relationships, and such DAGs form a Markov equivalence class.
- A Markov equivalence class of DAGs can be uniquely represented by a completed partially directed acyclic graph (CPDAG).
- A CPDAG satisfies: $X_i \to X_j$ in the CPDAG if $X_i \to X_j$ in every DAG in the Markov equivalence class, and $X_i - X_j$ in the CPDAG if the Markov equivalence class contains a DAG for which $X_i \to X_j$ as well as a DAG for which $X_i \leftarrow X_j$.
- Question: How to extend these methods for Poisson count data and can we reduce CPDAGs given perturb-seq interventional data?

Causal Inference and DAGs

- The causal DAG is typically not identifiable from observational data from the distribution of X.
- Several DAGs can encode the same conditional independence relationships, and such DAGs form a Markov equivalence class.
- A Markov equivalence class of DAGs can be uniquely represented by a completed partially directed acyclic graph (CPDAG).
- A CPDAG satisfies: $X_i \to X_j$ in the CPDAG if $X_i \to X_j$ in every DAG in the Markov equivalence class, and $X_i - X_j$ in the CPDAG if the Markov equivalence class contains a DAG for which $X_i \to X_j$ as well as a DAG for which $X_i \leftarrow X_j$.
- Question: How to extend these methods for Poisson count data and can we reduce CPDAGs given perturb-seq interventional data?

Causal Inference and DAGs

- The causal DAG is typically not identifiable from observational data from the distribution of X.
- Several DAGs can encode the same conditional independence relationships, and such DAGs form a Markov equivalence class.
- A Markov equivalence class of DAGs can be uniquely represented by a completed partially directed acyclic graph (CPDAG).
- A CPDAG satisfies: $X_i \to X_j$ in the CPDAG if $X_i \to X_j$ in every DAG in the Markov equivalence class, and $X_i - X_j$ in the CPDAG if the Markov equivalence class contains a DAG for which $X_i \to X_j$ as well as a DAG for which $X_i \leftarrow X_j$.
- Question: How to extend these methods for Poisson count data and can we reduce CPDAGs given perturb-seq interventional data?

Causal Inference and DAGs

- The causal DAG is typically not identifiable from observational data from the distribution of X.
- Several DAGs can encode the same conditional independence relationships, and such DAGs form a Markov equivalence class.
- A Markov equivalence class of DAGs can be uniquely represented by a completed partially directed acyclic graph (CPDAG).
- A CPDAG satisfies: $X_i \to X_j$ in the CPDAG if $X_i \to X_j$ in every DAG in the Markov equivalence class, and $X_i - X_j$ in the CPDAG if the Markov equivalence class contains a DAG for which $X_i \to X_j$ as well as a DAG for which $X_i \leftarrow X_j$.
- Question: How to extend these methods for Poisson count data and can we reduce CPDAGs given perturb-seq interventional data?

Causal Inference and DAGs

- The causal DAG is typically not identifiable from observational data from the distribution of X.
- Several DAGs can encode the same conditional independence relationships, and such DAGs form a Markov equivalence class.
- A Markov equivalence class of DAGs can be uniquely represented by a completed partially directed acyclic graph (CPDAG).
- A CPDAG satisfies: $X_i \to X_j$ in the CPDAG if $X_i \to X_j$ in every DAG in the Markov equivalence class, and $X_i - X_j$ in the CPDAG if the Markov equivalence class contains a DAG for which $X_i \to X_j$ as well as a DAG for which $X_i \leftarrow X_j$.
- Question: How to extend these methods for Poisson count data and can we reduce CPDAGs given perturb-seq interventional data?

Problem: Estimating DAGs with Poisson count data

- Reall the PLN model:

$$\begin{aligned} \boldsymbol{X}_i^{(k)} &\sim \text{Normal}, \\ y_{ij}^{(k)} &\sim \text{Pois}(\exp(X_{ij}^{(k)})), \end{aligned} \tag{4}$$

- In the LSEMs:

$$\boldsymbol{X}^{(k)} \leftarrow \boldsymbol{X}^{(k)}\boldsymbol{W}^{(k)} + \epsilon^{(k)}, \qquad k = 0, 1, \cdots, p, \tag{5}$$

- **dotears** (Xue et al., 2025) algorithm:

$$\begin{aligned} &\min_{W} \frac{1}{p}\sum_{k=0}^{p}\frac{1}{2n_k}\left\|\left(\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k)}\boldsymbol{W}^{(k)}\right)\hat{\Omega}_0^{-\frac{1}{2}}\right\|_F^2 + \lambda\|\boldsymbol{W}\|_1 \\ &\text{s.t.} \quad h(\boldsymbol{W}) := \text{tr}[\exp(\boldsymbol{W}\odot\boldsymbol{W})] - p = 0. \end{aligned} \tag{6}$$

Problem: Estimating DAGs with Poisson count data

- Reall the PLN model:

$$\begin{aligned} \boldsymbol{X}_i^{(k)} &\sim \text{Normal}, \\ y_{ij}^{(k)} &\sim \text{Pois}(\exp(X_{ij}^{(k)})), \end{aligned} \tag{4}$$

- In the LSEMs:

$$\boldsymbol{X}^{(k)} \leftarrow \boldsymbol{X}^{(k)} \boldsymbol{W}^{(k)} + \boldsymbol{\epsilon}^{(k)}, \qquad k = 0, 1, \cdots, p, \tag{5}$$

- **dotears** (Xue et al., 2025) algorithm:

$$\begin{aligned} &\min_{W} \frac{1}{p} \sum_{k=0}^{p} \frac{1}{2n_k} \left\| \left(\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k)} \boldsymbol{W}^{(k)} \right) \hat{\Omega}_0^{-\frac{1}{2}} \right\|_F^2 + \lambda \|\boldsymbol{W}\|_1 \\ &\text{s.t.} \quad h(\boldsymbol{W}) := \text{tr}[\exp(\boldsymbol{W} \odot \boldsymbol{W})] - p = 0. \end{aligned} \tag{6}$$

Problem: Estimating DAGs with Poisson count data

- Reall the PLN model:

$$\begin{aligned} \boldsymbol{X}_i^{(k)} &\sim \text{Normal}, \\ y_{ij}^{(k)} &\sim \text{Pois}(\exp(X_{ij}^{(k)})), \end{aligned} \tag{4}$$

- In the LSEMs:

$$\boldsymbol{X}^{(k)} \leftarrow \boldsymbol{X}^{(k)}\boldsymbol{W}^{(k)} + \boldsymbol{\epsilon}^{(k)}, \qquad k = 0, 1, \cdots, p, \tag{5}$$

- **dotears** (Xue et al., 2025) algorithm:

$$\begin{aligned} &\min_{W} \frac{1}{p} \sum_{k=0}^{p} \frac{1}{2n_k} \left\| \left(\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k)}\boldsymbol{W}^{(k)} \right) \hat{\Omega}_0^{-\frac{1}{2}} \right\|_F^2 + \lambda \|\boldsymbol{W}\|_1 \\ &\text{s.t.} \quad h(\boldsymbol{W}) := \text{tr}[\exp(\boldsymbol{W} \odot \boldsymbol{W})] - p = 0. \end{aligned} \tag{6}$$

References I

Barry, T., K. Mason, K. Roeder, and E. Katsevich (2024). Robust differential expression testing for single-cell crispr screens at low multiplicity of infection. *Genome biology 25*(1), 124.

Jackson, C. A., D. M. Castro, G.-A. Saldi, R. Bonneau, and D. Gresham (2020). Gene regulatory network reconstruction using single-cell rna sequencing of barcoded genotypes in diverse environments. *eLife 9*, e51254.

Lin, Y., Y. Cao, E. Willie, E. Patrick, and J. Yang (2023, 07). Atlas-scale single-cell multi-sample multi-condition data integration using scmerge2. *Nature Communications 14*.

Nandy, P., A. Hauser, and M. H. Maathuis (2018). High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics 46*(6A), 3151–3183.

References II

Replogle, J. M., R. A. Saunders, A. N. Pogson, J. A. Hussmann, A. Lenail, A. Guna, L. Mascibroda, E. J. Wagner, K. Adelman, G. Lithwick-Yanai, N. Iremadze, F. Oberstrass, D. Lipson, J. L. Bonnar, M. Jost, T. M. Norman, and J. S. Weissman (2022). Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell 185*(14), 2559–2575.e28.

Smillie, C. S., M. Biton, J. Ordovas-Montanes, K. M. Sullivan, G. Burgin, D. B. Graham, R. H. Herbst, N. Rogel, M. Slyper, J. Waldman, M. Sud, E. Andrews, G. Velonias, A. L. Haber, K. Jagadeesh, S. Vickovic, J. Yao, C. Stevens, D. Dionne, L. T. Nguyen, A.-C. Villani, M. Hofree, E. A. Creasey, H. Huang, O. Rozenblatt-Rosen, J. J. Garber, H. Khalili, A. N. Desch, M. J. Daly, A. N. Ananthakrishnan, A. K. Shalek, R. J. Xavier, and A. Regev (2019). Intra- and inter-cellular rewiring of the human colon during ulcerative colitis. *Cell 178*, 714–730.e22.

References III

Weinstein, J. N., E. A. Collisson, G. B. Mills, K. R. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics 45*(10), 1113–1120.

Xue, A., J. Rao, S. Sankararaman, and H. Pimentel (2025). dotears: Scalable and consistent directed acyclic graph estimation using observational and interventional data. *iScience 28*(2), 111673.

Thank you!