Improved Methods for Empirical Bayes Multivariate Multiple Testing and Effect Size Estimation

Yunqi Yang, Peter Carbonetto, David Gerard, Matthew Stephens

Jiahui REN

City University of Hong Kong

Journal Club Presentation 2025.11

Introduction

Motivation

- Suppose we are interested in estimating the sharing of genetic effects across different conditions (e.g., assessing the effects of many expression quantitative trait loci (eQTLs) across many tissues).
- Simplest way: analyzing each condition separately. But it would fail to exploit the sharing or similarity of effects across conditions.
- Motivates consideration of multivariate approaches to multiple testing and effect size estimation.

Background: Empirical Bayes Framework-mash

Goal: Eestimates effects of many units (n) in many conditions (R)
Model Settings:

- Let $\boldsymbol{x}_j \in \mathbb{R}^R$ be the observed vectors
- Let $\boldsymbol{\theta}_j \in \mathbb{R}^R$ be the true vectors
- Likelihood: $\boldsymbol{x}_j \mid \boldsymbol{\theta}_j, \boldsymbol{V}_j \sim N_R(\boldsymbol{\theta}_j, \boldsymbol{V}_j), j = 1, \ldots, n$
- Prior: $\boldsymbol{\theta}_j \sim \sum_{k=1}^{K} \sum_{l=1}^{L} \pi_{k,l} N(0, \omega_l \boldsymbol{U}_k)$, where $\boldsymbol{\pi} \in S_K \subseteq \mathbb{R}^K$(K-dimensional simplex) is the set of mixture proportions , ω_l is a scaling coefficient corresponds to a different effect size, $\mathcal{U} := \{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K\}$ denotes the full collection of covariance matrices.

Background: mash framework

MASH is desgined to

1. Estimate $\boldsymbol{\pi}$ and $\mathcal{U}$ by maximizing the likelihood of the observed data
   a. Estimate $\mathcal{U}$ by maximum-likelihood on a subset of the data (using Exteme Deconvolution algorithm)
   b. Estimate $\boldsymbol{\pi}$ by maximizing the likelihood from all the data (using fast optimization algorithms)
2. Compute posterior distribution $p(\boldsymbol{\theta}_j|\boldsymbol{x}_j, \hat{\boldsymbol{\pi}}, \hat{\mathcal{U}}, \boldsymbol{V}_j)$

However, in the first step that estimates $\mathcal{U}$, several challenges were presented.

mash limitations

1. The Extreme Deconvolution (ED) algorithm can be slow to converge
2. The results ot ED are often sensitive to initalization
3. The estimated covariance matrices can be quite unstable when R is large relative to n

Therefore, the authors proposed a new method called "**Truncated Eigenvalue Decomposition**" (TED) and use some regularization schemes(penalty function) to improve the estimation of $\mathcal{U}$, and TED also converges much faster than ED.

Preliminaries: Reformulation

Scale invariance requires

$$\hat{\boldsymbol{\theta}}_j(s\boldsymbol{x}_1,\ldots,s\boldsymbol{x}_n,\ s^2\boldsymbol{V}_1,\ldots,s^2\boldsymbol{V}_n)\ =\ s\,\hat{\boldsymbol{\theta}}_j(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n,\ \boldsymbol{V}_1,\ldots,\boldsymbol{V}_n).$$

for the penalty function, it is required that $\tilde{\rho}(\boldsymbol{U})=\tilde{\rho}(s\boldsymbol{U})$ for any $s>0$, therefore consider:

$$\tilde{\rho}(\boldsymbol{U})=\min_{s>0}\rho(\boldsymbol{U}/s)$$

for any $s>0$.
The penalty function $\tilde{\rho}(\boldsymbol{U})$ is designed to penalize the "**shape**" rather than "**scale**" of the covariance matrices.

The Empirical Bayes Multivariate Normal Means Model(EBMNM)

- Observed vectors: $\boldsymbol{x}_j \in \mathbb{R}^R$ are independent, noisy, normally-distributed measurements of underlying true values $\boldsymbol{\theta}_j \in \mathbb{R}^R$

$$\boldsymbol{x}_j \mid \boldsymbol{\theta}_j \sim N_R(\boldsymbol{\theta}_j, \boldsymbol{V}_j), j = 1, \ldots, n \tag{1}$$

, where $\boldsymbol{V}_j \in P_R^+$ is assumed to be known and invertible.

- Assume the unknown means are independen and identically distributed draws from a mixtrue of zero-mean multivariate normals:

$$\boldsymbol{\theta}_j \sim \sum_{k=1}^{K} \pi_k N_R(\boldsymbol{\theta}_j; 0, \boldsymbol{U}_k) \tag{2}$$

, where $\boldsymbol{\pi} \in S_K \subseteq \mathbb{R}^K$(K-dimensional simplex) is the set of mixture proportions , $\mathcal{U} := \{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K\}$ denotes the full collection of covariance matrices. The $\boldsymbol{\omega}$ are absorbed in the $\boldsymbol{U}_k$ because the penalty function is scale invariant.

The Empirical Bayes Multivariate Normal Means Model(EBMNM)

- If $\boldsymbol{V}_j = \boldsymbol{V}$ for all $j = 1, \ldots, n$, we refer this as the "**homoscedastic**" case.
- If $\boldsymbol{V}_j \neq \boldsymbol{V}$ for some $j = 1, \ldots, n$, we refer this as the "**heteroscedastic**" case.

The Empirical Bayes Multivariate Normal Means Model(EBMNM)

The marginal distribution of $\boldsymbol{x}_j$ is:

$$p(\boldsymbol{x}_j|\boldsymbol{\pi},\mathcal{U}) = \sum_{k=1}^{K} \pi_k N_R(\boldsymbol{x}_j; 0, \boldsymbol{U}_k + \boldsymbol{V}_j) \tag{3}$$

And the log-likelihood is:

$$\ell(\boldsymbol{\pi},\mathcal{U}) = \sum_{j=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k N_R(\boldsymbol{x}_j; 0, \boldsymbol{U}_k + \boldsymbol{V}_j)\right) \tag{4}$$

The Empirical Bayes Multivariate Normal Means Model(EBMNM)

The EB approach fitting the model in two stages:

1. Estimate $\boldsymbol{\pi}$ and $\mathcal{U}$ by maximizing a penalized log-likelihood:

$$(\hat{\boldsymbol{\pi}}, \hat{\mathcal{U}}) := \underset{\boldsymbol{\pi} \in S_K, \boldsymbol{U} \in P_R^{+,k}, \boldsymbol{s} > 0}{\arg\max} \left(\ell(\boldsymbol{\pi}, \mathcal{U}) - \sum_{k=1}^{K} \tilde{\rho}(\boldsymbol{U}_k / s_k) \right) \tag{5}$$

2. Compute posterior distribution

$$p_{\text{post}}(\boldsymbol{\theta}_j) := p(\boldsymbol{\theta}_j | \boldsymbol{x}_j, \hat{\boldsymbol{\pi}}, \hat{\mathcal{U}}) \propto p(\boldsymbol{x}_j | \boldsymbol{\theta}_j) p(\boldsymbol{\theta}_j | \hat{\boldsymbol{\pi}}, \hat{\mathcal{U}}) \tag{6}$$

Penalty Function

Penalty Function

Two different penalties that have been previously used for covariance regularization:

1. The "**inverse Wishart**" (IW) penalty:

$$\begin{aligned}\rho_\lambda^{\mathsf{IW}}(\boldsymbol{U}) &:= \frac{\lambda}{2}\{\log|\boldsymbol{U}| + \mathsf{tr}(\boldsymbol{U}^{-1})\} \\ &= \frac{\lambda}{2}\sum_{i=1}^{R}(\log e_r + 1/e_r)\end{aligned} \tag{7}$$

2. The "**nuclear norm**" (NN) penalty:

$$\begin{aligned}\rho_\lambda^{\mathsf{NN}}(\boldsymbol{U}) &:= \lambda\{0.5||\boldsymbol{U}||_* + 0.5||\boldsymbol{U}^{-1}||_*\} \\ &= \frac{\lambda}{2}\sum_{i=1}^{R}(0.5e_i + 0.5/e_i)\end{aligned} \tag{8}$$

where e_i are the eigenvalues of $\boldsymbol{U}$ and $\boldsymbol{U}^{-1}$, and $||.||_*$ is the nuclear norm, $\lambda > 0$ controls the strength of the penalty.

Constraints

Constraints

As an alternative to penalized estimation of $\boldsymbol{U}$, the authors also consider estimating $\boldsymbol{U}$ under different constraints:

1. A **scaling** constraint: $\boldsymbol{U_k} = c_k \boldsymbol{U}_{0k}$, for some chosen $U_{0k} \in P_R^+$
2. A **rank-1** constraint: $\boldsymbol{U_k} = \boldsymbol{u}_k \boldsymbol{u}_k^T$, for some $\boldsymbol{u}_k \in \mathbb{R}^R$

Introduction to TED, FA and ED

A simpler case for K=1 and no penalty

For easily to understanding the three methods, we consider the simpler case for K=1 and no penalty.

- When $K = 1$, the prior is: $\boldsymbol{\theta}_j \sim N_R(\boldsymbol{\theta}_j; 0, \boldsymbol{U})$, the model is:

$$\boldsymbol{x}_j | \boldsymbol{U} \sim N_R(0, \boldsymbol{U} + \boldsymbol{V}_j), j = 1, \dots, n \tag{9}$$

- Goal: compute the maximum likelihood estimate of $\boldsymbol{U}$:

$$\hat{\boldsymbol{U}} := \underset{\boldsymbol{U} \in P_R^+}{\arg\max} \sum_{j=1}^{n} \log N_R(0, \boldsymbol{U} + \boldsymbol{V}_j) \tag{10}$$

Truncated Eigenvalue Decomposition (TED)

- Special case: homoscedastic noise $\boldsymbol{V}_j = \boldsymbol{I}_R$.
- Sample covariance:

$$\boldsymbol{S} := \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j \boldsymbol{x}_j^{\mathrm{T}}.$$

- Naive idea $\hat{\boldsymbol{U}} = \boldsymbol{S} - \boldsymbol{I}_R$ may not be PSD.
- TED sets negative eigenvalues of $(\boldsymbol{S} - \boldsymbol{I}_R)$ to zero:

$$\hat{\boldsymbol{U}} \ = \ (\boldsymbol{S} - \boldsymbol{I}_R)_+.$$

- General case $\boldsymbol{V}_j = \boldsymbol{V}$ (constant across j): whiten

$$\boldsymbol{V} = \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}, \quad \tilde{\boldsymbol{x}}_j = \boldsymbol{R}^{-1}\boldsymbol{x}_j, \quad \boldsymbol{R}^{-1}\boldsymbol{x_j}|\boldsymbol{\theta}_j \sim N_R(\boldsymbol{R}^{-1}\boldsymbol{\theta}_j, \boldsymbol{I} + \boldsymbol{R}^{-1}\boldsymbol{U}\boldsymbol{R})$$

Apply TED to $\tilde{\boldsymbol{x}}_j$ to get $\widehat{\boldsymbol{U}}'$ and then back-transform

$$\hat{\boldsymbol{U}} = \boldsymbol{R}\widehat{\boldsymbol{U}}'\boldsymbol{R}^{\mathrm{T}}.$$

Extreme Deconvolution (ED)

- EM algorithm for the $K = 1$ model using data augmentation:

$$\boldsymbol{\theta}_j \sim N_R(\mathbf{0}, \boldsymbol{U}), \qquad \boldsymbol{x}_j \mid \boldsymbol{\theta}_j \sim N_R(\boldsymbol{\theta}_j, \boldsymbol{V}_j).$$

- E-step: posterior mean and covariance of $\boldsymbol{\theta}_j$ given current $\boldsymbol{U}$:

$$\boldsymbol{b}_j := \boldsymbol{U}(\boldsymbol{U} + \boldsymbol{V}_j)^{-1}\boldsymbol{x}_j, \qquad \boldsymbol{B}_j := \boldsymbol{U} - \boldsymbol{U}(\boldsymbol{U} + \boldsymbol{V}_j)^{-1}\boldsymbol{U}.$$

- M-step: update covariance

$$\boldsymbol{U}^{\text{new}} = \frac{1}{n}\sum_{j=1}^{n}\Big(\boldsymbol{B}_j + \boldsymbol{b}_j\boldsymbol{b}_j^{\mathrm{T}}\Big).$$

- Guarantees monotone ascent of the likelihood and convergence to a stationary point.

Factor Analysis (FA)

- Rank-1 constraint on covariance: $\boldsymbol{U} = \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ with $\boldsymbol{u} \in \mathbb{R}^R$.
- Data augmentation:

$$a_j \sim N(0,1), \qquad \boldsymbol{x}_j \mid \boldsymbol{u}, \boldsymbol{V}_j, a_j \sim N_R(a_j\,\boldsymbol{u},\ \boldsymbol{V}_j).$$

- EM updates:

$$\mu_j := \sigma_j^2\,\boldsymbol{u}^{\mathrm{T}}\boldsymbol{V}_j^{-1}\boldsymbol{x}_j, \qquad \sigma_j^2 := \frac{1}{1 + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{V}_j^{-1}\boldsymbol{u}},$$

$$\boldsymbol{u}^{\text{new}} = \left(\sum_{j=1}^{n}(\mu_j^2 + \sigma_j^2)\boldsymbol{V}_j^{-1}\right)^{-1}\left(\sum_{j=1}^{n}\mu_j\,\boldsymbol{V}_j^{-1}\boldsymbol{x}_j\right).$$

Extend to General Case with K>1 and penalty

Derivation of the EM algorithm for fitting the EBMNM model

- First, they introduce a latent variable $\boldsymbol{z}_j$ for each $j = 1, \ldots, n$. Each $\boldsymbol{z}_j$ is a binary vector of length K indicating the component k from which $\boldsymbol{x}_j$ arose.
- Following Neal and Hinton (1998), the expected value of z_{jk} is:

$$\omega_{jk} := \mathbb{E}[z_{jk}] = \frac{\pi_k N_R(0, \boldsymbol{U}_k + \boldsymbol{V}_j)}{\sum_{l=1}^{K} \pi_l N_R(0, \boldsymbol{U}_l + \boldsymbol{V}_j)} \tag{11}$$

- The M-step for $\boldsymbol{\pi}$:

$$\hat{\pi}_k = \frac{1}{n} \sum_{j=1}^{n} \omega_{jk} \tag{12}$$

TED without penalty

- Special case $\boldsymbol{V}_j = \boldsymbol{I}_R$; define weighted covariance

$$\hat{\boldsymbol{S}} := \sum_{j=1}^{n} \tilde{w}_j \, \boldsymbol{x}_j \boldsymbol{x}_j^{\mathrm{T}}, \quad \tilde{w}_j := \frac{w_j}{\bar{w}}, \ \ \bar{w} = \sum_{i=1}^{n} w_i.$$

- Objective (dropping U-independent terms):

$$\phi(\boldsymbol{U}; \boldsymbol{w}) \ = \ -\frac{\bar{w}}{2} \Big\{ \log |\boldsymbol{U} + \boldsymbol{I}| + \operatorname{tr} \big[(\boldsymbol{U} + \boldsymbol{I})^{-1} \hat{\boldsymbol{S}} \big] \Big\}.$$

- Closed-form solution:

$$\hat{\boldsymbol{U}} \ = \ (\hat{\boldsymbol{S}} - \boldsymbol{I})_{+}.$$

  If $\hat{\boldsymbol{S}} = \boldsymbol{L} \operatorname{diag}(d_1, \ldots, d_R) \boldsymbol{L}^{\mathrm{T}}$ then

$$\hat{\boldsymbol{U}} \ = \ \boldsymbol{L} \operatorname{diag} \big(\max\{d_r - 1, \, 0\} \big)_{r=1}^{R} \boldsymbol{L}^{\mathrm{T}}.$$

- If $\boldsymbol{V}_j = \boldsymbol{V} \neq \boldsymbol{I}$ (constant), whiten $\tilde{\boldsymbol{x}}_j = \boldsymbol{R}^{-1} \boldsymbol{x}_j$ with $\boldsymbol{V} = \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}$, apply the above on $\tilde{\boldsymbol{x}}_j$, then back-transform.

TED with penalty

- For IW/NN penalties there is no closed form, use the numerical method:
- With separable penalty $\rho(\boldsymbol{U}/s) = \sum_{r=1}^{R} \rho_r(e_r/s)$, write $\hat{\boldsymbol{S}} = \boldsymbol{L}\,\mathrm{diag}(d_1, \ldots, d_R)\boldsymbol{L}^{\mathrm{T}}$.
- Optimize eigenvalues independently:

$$\hat{e}_r \;=\; \operatorname*{arg\,max}_{e_r \geq 0} \; -\frac{\bar{w}}{2}\left[\log(e_r+1) + \frac{d_r}{e_r+1}\right] - \rho_r(e_r/s).$$

- Update

$$\hat{\boldsymbol{U}} \;=\; \boldsymbol{L}\;\mathrm{diag}(\hat{e}_1, \ldots, \hat{e}_R)\,\boldsymbol{L}^{\mathrm{T}}.$$

ED without penalty

- Weighted complete-data objective:

$$\phi^{\mathsf{ED}}(\boldsymbol{U}, \Theta; \boldsymbol{w}) \;=\; \sum_{j=1}^{n} w_j \log p(\boldsymbol{x}_j, \boldsymbol{\theta}_j \mid \boldsymbol{U}, \boldsymbol{V}_j).$$

- E-step (posterior of $\boldsymbol{\theta}_j$ given current $\boldsymbol{U}$):

$$\boldsymbol{b}_j = \boldsymbol{U}(\boldsymbol{U} + \boldsymbol{V}_j)^{-1}\boldsymbol{x}_j, \qquad \boldsymbol{B}_j = \boldsymbol{U} - \boldsymbol{U}(\boldsymbol{U} + \boldsymbol{V}_j)^{-1}\boldsymbol{U}.$$

- M-step (closed form, no penalty). With normalized weights $\tilde{w}_j := \frac{w_j}{\bar{w}}$, $\bar{w} := \sum_{i=1}^{n} w_i$,

$$\boldsymbol{U}^{\mathsf{new}} \;=\; \sum_{j=1}^{n} \tilde{w}_j \Big(\boldsymbol{B}_j + \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}}\Big).$$

For unweighted data ($w_j = 1$), this reduces to $\boldsymbol{U}^{\mathsf{new}} = \frac{1}{n}\sum_j (\boldsymbol{B}_j + \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}})$.

ED with penalty

- Add a shape penalty on $\boldsymbol{U}$; with the IW penalty and scale parameter s, the M-step maximizes

$$\mathbb{E}\left[\phi^{\mathsf{ED}}(\boldsymbol{U}, \Theta; \boldsymbol{w})\right] \;-\; \rho_\lambda^{\mathsf{IW}}(\boldsymbol{U}/s).$$

- Closed-form update with IW penalty:

$$\boldsymbol{U}^{\mathsf{new}} \;=\; \frac{\sum_{j=1}^{n} w_j \left(\boldsymbol{B}_j + \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}}\right) + \lambda s\, \boldsymbol{I}_R}{\sum_{j=1}^{n} w_j + \lambda}.$$

- Under the NN penalty, the update is not closed-form.

FA

- Rank-1 parameterization: $\boldsymbol{U} = \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$, $\boldsymbol{u} \in \mathbb{R}^R$.
- Model (weighted case): $\boldsymbol{x}_j \mid \boldsymbol{u}, \boldsymbol{V}_j \sim N_R\big(\mathbf{0},\, \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} + \boldsymbol{V}_j\big)$.
- Augmentation: $a_j \sim N(0,1)$, and $\boldsymbol{x}_j \mid a_j, \boldsymbol{u}, \boldsymbol{V}_j \sim N_R(a_j\boldsymbol{u}, \boldsymbol{V}_j)$.
- E-step (posterior of $a_j \mid \boldsymbol{x}_j$):

$$\mu_j \;=\; \sigma_j^2\, \boldsymbol{u}^{\mathrm{T}} \boldsymbol{V}_j^{-1} \boldsymbol{x}_j, \qquad \sigma_j^2 \;=\; \frac{1}{1 + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{V}_j^{-1} \boldsymbol{u}}.$$

- M-step (closed form with weights w_j):

$$\boldsymbol{u}^{\text{new}} \;=\; \left(\sum_{j=1}^{n} w_j (\mu_j^2 + \sigma_j^2) \boldsymbol{V}_j^{-1} \right)^{-1} \left(\sum_{j=1}^{n} w_j\, \mu_j\, \boldsymbol{V}_j^{-1} \boldsymbol{x}_j \right).$$

Updating the Scaling Parameter

- Update the scale s by maximizing the s-dependent part of the objective:

$$s^{\text{new}} = \arg\max_{s>0} -\rho(\boldsymbol{U}/s).$$

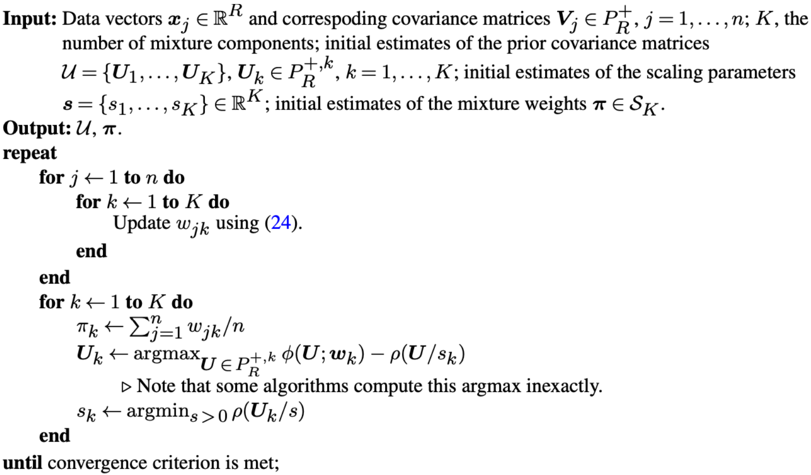
- Closed-form solutions:
  - IW penalty:

$$s^{\text{new}} = \frac{R}{\operatorname{tr}(\boldsymbol{U}^{-1})}.$$

  - NN penalty:

$$s^{\text{new}} = \sqrt{\frac{\operatorname{tr}(\boldsymbol{U})}{\operatorname{tr}(\boldsymbol{U}^{-1})}}.$$

Overview of the Algorithm



```
Input: Data vectors $\boldsymbol{x}_j \in \mathbb{R}^R$ and correspoding covariance matrices $\boldsymbol{V}_j \in P_R^+$, $j = 1, \ldots, n$; $K$, the
       number of mixture components; initial estimates of the prior covariance matrices
       $\mathcal{U} = \{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K\}$, $\boldsymbol{U}_k \in P_R^{+,k}$, $k = 1, \ldots, K$; initial estimates of the scaling parameters
       $\boldsymbol{s} = \{s_1, \ldots, s_K\} \in \mathbb{R}^K$; initial estimates of the mixture weights $\boldsymbol{\pi} \in \mathcal{S}_K$.
Output: $\mathcal{U}$, $\boldsymbol{\pi}$.
repeat
    for $j \leftarrow 1$ to $n$ do
        for $k \leftarrow 1$ to $K$ do
            Update $w_{jk}$ using (24).
        end
    end
    for $k \leftarrow 1$ to $K$ do
        $\pi_k \leftarrow \sum_{j=1}^n w_{jk}/n$
        $\boldsymbol{U}_k \leftarrow \mathrm{argmax}_{\boldsymbol{U} \in P_R^{+,k}} \phi(\boldsymbol{U}; \boldsymbol{w}_k) - \rho(\boldsymbol{U}/s_k)$
            ▷ Note that some algorithms compute this argmax inexactly.
        $s_k \leftarrow \mathrm{argmin}_{s>0} \rho(\boldsymbol{U}_k/s)$
    end
until convergence criterion is met;
```

Simulations

Simulation Settings

1. Goal:
   - Compare algorithms for updating $\boldsymbol{U}_k$ (TED, ED, FA).
   - Assess benefits of penalties (IW/NN) and constraints (e.g., rank-1).
   - Study sensitivity to the number of mixture components K.
2. Data generation:
   - Generate "true" means $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n \in \mathbb{R}^R$ from the mixture prior with K components.
   - Observed data: $\boldsymbol{x}_j \sim N_R(\boldsymbol{\theta}_j, \boldsymbol{V}_j)$ independently for $j = 1, \dots, n$.
   - Create separate test sets $(\boldsymbol{x}_j^{\text{test}}, \boldsymbol{\theta}_j^{\text{test}})$ to evaluate generalization.
   - In the main comparisons, use homoskedastic noise $\boldsymbol{V}_j = \boldsymbol{I}_R$ for all j.
   - Set $K = 10$ with uniform mixture weights $\pi_1 = \cdots = \pi_{10} = 1/10$.
3. Two scenarios:
   - Scenario 1: Hybrid covariances.
   - Scenario 2: Rank-1 covariances.

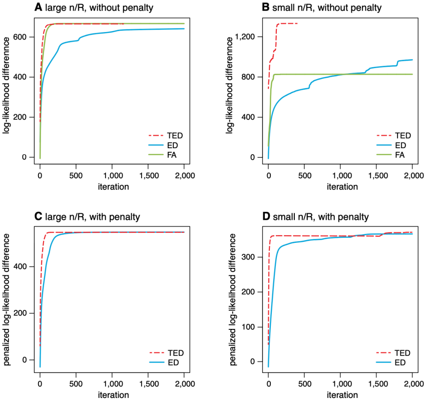
Scenarios

1. Scenario 1: Hybrid covariances:
   - Construct $K = 10$ prior covariances by combining canonical and random draws.
   - 3 canonical matrices:
     - $\boldsymbol{U}_1 = 5\,\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}}$ (“singleton” in first coordinate)
     - $\boldsymbol{U}_2 = 5\,\mathbf{1}\mathbf{1}^{\mathrm{T}}$ (equal means across dimensions)
     - $\boldsymbol{U}_3 = 5\,\boldsymbol{I}_R$ (independent effects)
   - Remaining 7 matrices sampled from an inverse-Wishart.
2. Scenario 2: Rank-1 covariances:
   - Use 5 coordinate “spike” covariances: $\boldsymbol{U}_k = 5\,\boldsymbol{e}_k\boldsymbol{e}_k^{\mathrm{T}}$ for $k = 1, \ldots, 5$.
   - Remaining 5 are random rank-1: $\boldsymbol{U}_k = \boldsymbol{u}_k\boldsymbol{u}_k^{\mathrm{T}}$ with $\boldsymbol{u}_k \sim N_R(\mathbf{0}, \boldsymbol{I}_R)$.
   - Maintains $\boldsymbol{V}_j = \boldsymbol{I}_R$ to focus on method differences.

Dataset sizes and evaluation

- Two regimes:
  - Large-n/low-R: $n = 10{,}000$, $R = 5$.
  - Small-n/high-R: $n = 1{,}000$, $R = 50$.
- Fit on training data; evaluate on both training and held-out test sets.
- Compare accuracy of inferences (e.g., effect estimation), and computational efficiency.

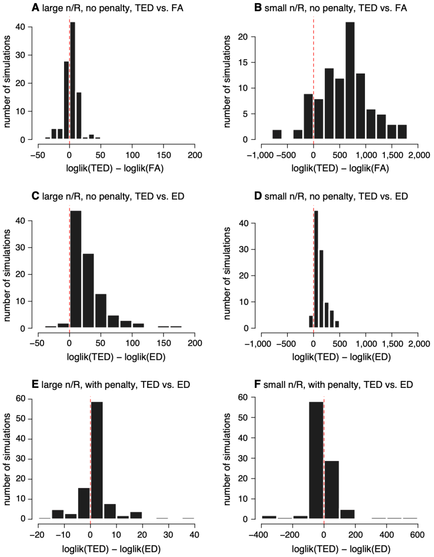
Comparison of convergence



Convergence

- **Warm-start:** Prefit 20 iterations of ED
- **Without penalty**: TED and FA rise to near-optimal values within a few iterations; ED is much slower.
- **With IW penalty**: TED still fastest; gap to ED narrows but remains.
- **Small** n/R: methods can converge to different local optima.

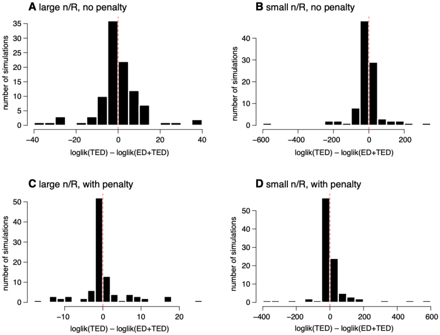
Aggregate results over 100 simulations



- **Without penalty(A-D)**: TED achieves better solutions than FA and ED.
- **With IW penalty**: TED and ED are similar

Question: The improved performance of TED is due to faster convergence or better solutions?

Can TED rescue ED?

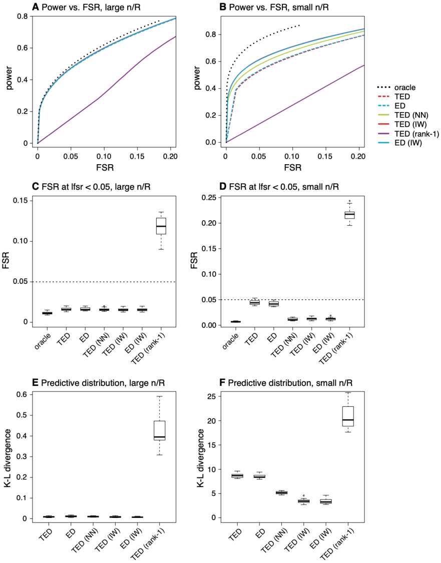


- Protocol: running TED initialized to the ED solution ("ED+TED")
  - if ED is simply slow to converge, then ED+TED will be similar to TED.
  - If ED converges to a poorer local optimum, then TED will not rescue it, and ED+TED will be similar to ED.
- Even 100,000 ED iterations often fall short of 1,000 TED updates (avg gap 40.6 loglik units).

Comparison of the penalties and constraints

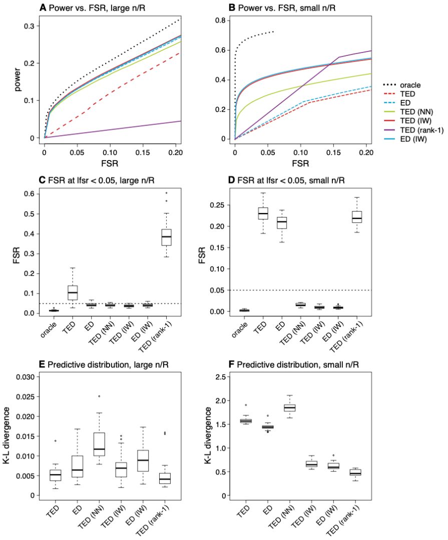
- Metrics used:
  - Power vs false sign rate (FSR): better curves have higher power at a given FSR.
  - Empirical FSR among tests with $\widehat{\text{lfsr}} < 0.05$; well-calibrated methods have small FSR (ideally below 0.05).
  - Accuracy of predictive distribution on test sets (approximate Kullback–Leibler divergence):

$$\frac{1}{n_{\text{test}}}\sum_{j=1}^{n_{\text{test}}}\log\left\{\frac{p(\boldsymbol{x}_j^{\text{test}} \mid \boldsymbol{U}^{\text{true}}, \boldsymbol{\pi}^{\text{true}}, \boldsymbol{V}_j^{\text{test}})}{p(\boldsymbol{x}_j^{\text{test}} \mid \widehat{\boldsymbol{U}}, \widehat{\boldsymbol{\pi}})}\right\}.$$



Hybrid scenario

- **small** n/R: both IW and NN improve power vs. FSR and the accuracy of predictive distribution.
- TED and ED are similar in this scenario.
- **rank-1 constraint**: performance is very poor.

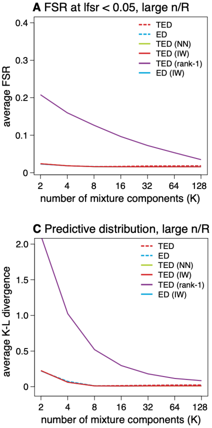


Rank-1 scenario

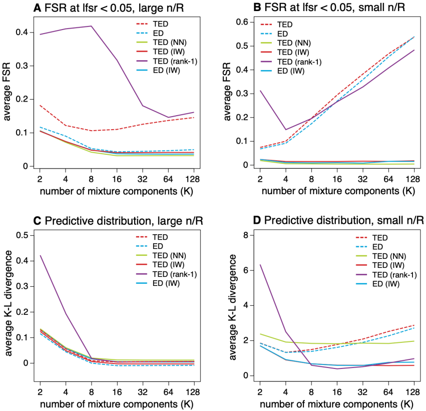
- **predictive distribution**: The predictive distribution is improved.
- rank-1 constraint performs very poor in other metrics because the $lfsr$ do not differ across conditions.

Robustness to mis-specifying the number of mixture components

- In previous simulations, models used the true number of components K.
- In practice K is unknown. Overstating K could cause overfitting.
- Setup: reuse the same 80 data sets (true $K = 10$); fit with $K \in \{2, \ldots, 128\}$.
- Evaluate:
    - Predictive accuracy on test data (smaller K–L divergence is better).
    - Average FSR among tests with $\widehat{\mathrm{lfsr}} < 0.05$ (smaller is better).



Hybrid scenario



Rank-1 scenario

Results

1. Results: large n/low R:
   - Most methods are robust to increasing K up to 128.
   - Fits with rank-1 constraints degrade when K is very large.
   - Penalized and unpenalized methods behave similarly in this regime.
2. Results: small n/high R:
   - Unpenalized methods worsen as K grows (higher FSR, worse predictive K—L): evidence of overfitting.
   - Penalized methods (IW/NN) remain stable; performance does not substantially decline with larger K.
3. Penalties help:
   - Penalization effectively selects/merges components:
     - Fewer “important”components where $\hat{\pi}_k > 0.01$.
     - Some $\boldsymbol{U}_k$ are estimated to be very similar.
   - Shrinkage toward the identity matrix keeps redundant components near negligible weight.

Real Data Analysis

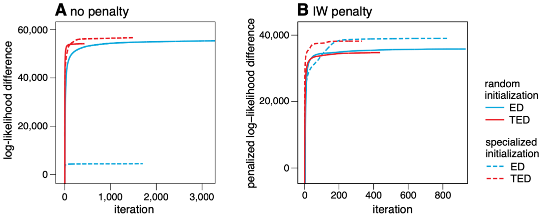
Analysis of genetic effects on gene expression in 49 human tissues

- Goal: apply the EBMNM model to multi-tissue cis-eQTL analysis (GTEx).
- Prior work: two-stage EBMNM analysis in (Urbut et al., 2019).
- Here we focus on estimating prior covariances (stage 1).
- Compare matrix-update algorithms (TED vs ED), initializations (specialized vs random), and penalties (none vs IW).
- Analyzed z-scores from association tests between gene expression and genotypes across tissues.
- After filtering: $n = 15{,}636$ genes, $R = 49$ tissues.
- Set measurement covariances to a common correlation matrix: $V_j = C$.
- Select, per gene, the variant with the largest magnitude z-score across tissues (as in (Urbut et al., 2019)).

Model fitting setup

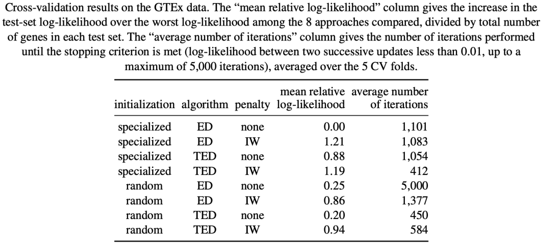
- Mixture size: $K = 40$ components (chosen to match number produced by specialized init).
- Initialization:
  - Specialized init from (Urbut et al., 2019) (rank-1—heavy, subspace-preserving under ED).
  - Simple random init as a baseline.
- Algorithms: TED and ED; penalties: none or IW (with $\lambda = R$).
- Stopping: difference in (penalized) log-likelihood $<$ 0.01 or 5,000 updates.

Convergence and fit quality



- Without penalty: specialized init + TED achieves the best fits quickly; ED with random init is slower and worse.
- With IW penalty: gaps narrow; TED remains competitive or better, and converges in fewer iterations.
- Specialized init improves fit quality vs random init but adds substantial computation time.

Cross-validation results

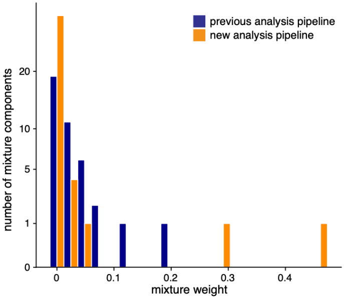


Cross-validation results on the GTEx data. The “mean relative log-likelihood” column gives the increase in the test-set log-likelihood over the worst log-likelihood among the 8 approaches compared, divided by total number of genes in each test set. The “average number of iterations” column gives the number of iterations performed until the stopping criterion is met (log-likelihood between two successive updates less than 0.01, up to a maximum of 5,000 iterations), averaged over the 5 CV folds.

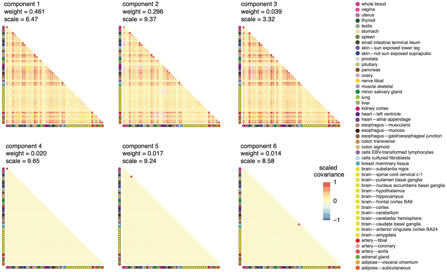
initialization	algorithm	penalty	mean relative log-likelihood	average number of iterations
specialized	ED	none	0.00	1,101
specialized	ED	IW	1.21	1,083
specialized	TED	none	0.88	1,054
specialized	TED	IW	1.19	412
random	ED	none	0.25	5,000
random	ED	IW	0.86	1,377
random	TED	none	0.20	450
random	TED	IW	0.94	584

- 5-fold CV: train on 80% genes, evaluate test-set log-likelihood on 20%.
- Penalty (IW) consistently improves test-set performance.
- Specialized init further improves test-set log-likelihood vs random init, but with notable overhead.
- Among 8 pipelines (TED/ED × init × penalty), ED with no penalty and specialized init performs worst on test sets.

Estimated priors and sharing patterns

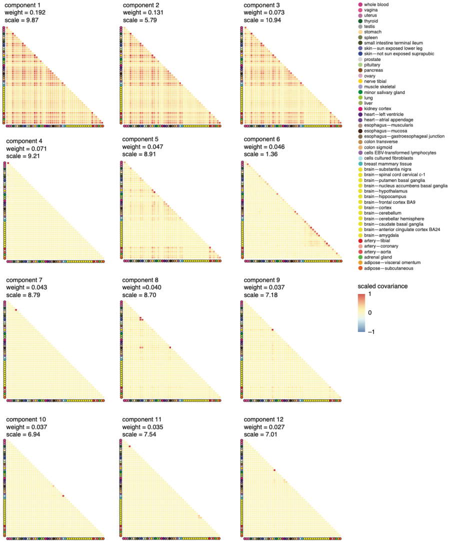


- New penalized pipeline produces more evenly distributed mixture weights (more “important”components).



Old

- Top covariance components capture broad tissue-sharing and tissue-specific patterns (e.g., brain clusters).
- New pipeline learned a greater variety of tissue-specific patterns



New

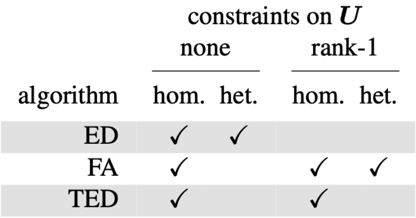
R package

R package

```r
devtools::install("/Users/jiahui/Downloads/
                   arXiv-2406.08784v1/anc/
                   supplementary_code/udr_0.3-153",
                   dependencies = TRUE, upgrade = "never")
library(udr)

set.seed(1)
V <- rbind(c(0.8, 0.2),
           c(0.2, 1.5))
U <- list(none   = rbind(c(0,0), c(0,0)),
          shared = rbind(c(1.0,0.9), c(0.9,1.0)))
w <- c(0.8, 0.2)

X <- simulate_ud_data(n = 2000, w = w, U = U, V = V)

fit <- ud_init(X, V = V)  # default prior: 2 scaled, 4 rank-1, 8 unconstrained
fit <- ud_fit(fit, control = list(version = "Rcpp", maxiter = 50))

logLik(fit)
summary(fit)
plot(fit$progress$iter,
     max(fit$progress$loglik) - fit$progress$loglik + 0.1,
     type = "l", col = "dodgerblue", lwd = 2, log = "y",
     xlab = "iteration", ylab = "dist to best loglik")
```

Conclusion

Conclusion



	constraints on $\boldsymbol{U}$			
	none		rank-1	
algorithm	hom.	het.	hom.	het.
ED	✓	✓		
FA	✓		✓	✓
TED	✓		✓	

Conclusion

- TED provides fast, stable covariance updates; ED can be slow and initialization-sensitive.
- Shape penalties (IW/NN) improve calibration and test performance, especially when n is small and R is large.
- With penalties, using a relatively large K is robust; without penalties, large K can overfit.
- Real data (GTEx): penalized methods and TED yield better or comparable fits with fewer iterations and more interpretable priors.

References

[1]Yunqi Yang, Peter Carbonetto, David Gerard, and Matthew Stephens. Improved methods for empirical Bayes multivariate multiple testing and effect size estimation. arXiv:2406.08784, 2024. Available at `https://arxiv.org/abs/2406.08784`.

[2]S. M. Urbut, G. Wang, P. Carbonetto, and M. Stephens. Flexible statistical methods for estimating and testing effects in genomic studies with multiple conditions. *Nature Genetics*, 51:187–195, 2019. `https://doi.org/10.1038/s41588-018-0268-8`.