Supplementary Material to "A covariance-enhanced approach to multi-tissue joint eQTL mapping with application to transcriptome-wide association studies"

Aaron J. Molstad[1], Wei Sun[2,3,4], Li Hsu[2,3]
University of Florida[1], Fred Hutchinson Cancer Research Center[2],
University of Washington[3], University of North Carolina-Chapel Hill[4]

1 Penalized ECM implementation details

Because EM algorithms are known to converge slowly, we employed a number of tricks to improve the efficiency of our implementation of Algorithm 1 and its sub-algorithms. Specifically, when solving Step 3 and Step 4 of Algorithm 1, we use *warm starting*, i.e., we initialize the algorithms for computing the $(k+1)$th iterates at the corresponding optimization variable's kth iterate. This can substantially reduce the computing time needed for each of the sub-algorithms because in general, the initializing value should be close to the minimizer.

In addition, we carefully construct a set of candidate tuning parameters so that we fit a set of models with varying degrees of sparsity without redundancy or superfluous computation. Our approach for finding reasonable tuning parameter is hierarchical: we first fix α and find a set of reasonable λ_Ω. Then, for each pair (α, λ_Ω), we construct a set of λ_β which yield non-sparse estimates of $\boldsymbol{\beta}_*$ using the Karush-Kuhn-Tucker (KKT) conditions for (2.8) of the main manuscript. Let $\bar{\alpha}$ and $\bar{\lambda}_\Omega$ denote the vectors of candidate tuning parameters for α and λ_Ω, respectively.

Specifically, we first set $\bar{\alpha} = 2^{-x}$ where $x \subseteq \{0, 1, 2, \ldots, 14, 15, 16\}$. We then run the ECM algorithm with $\lambda_\Omega = 0$ and $\lambda_\beta = \infty$. This way, we find the maximum likelihood estimator of $\boldsymbol{\Omega}_*$, say $\hat{\boldsymbol{\Sigma}}^{-1}$ without penalization under the restriction that $\boldsymbol{\beta} = 0$. Based on the KKT conditions for (2.7), it can be shown that if $\lambda_\Omega > \max_{j \neq k}[\mathbf{S}(0, \hat{\boldsymbol{\Sigma}})]_{j,k}$, the solution must be diagonal (Witten et al., 2011). Thus, we set

$$\bar{\lambda}_\Omega \subseteq [0.1 \max_{j \neq k}[\mathbf{S}(0, \hat{\Sigma})]_{j,k}, \max_{j \neq k}[\mathbf{S}(0, \hat{\Sigma})]_{j,k}],$$

where elements of $\bar{\lambda}_\Omega$ are equally spaced on the log-2 scale. Finally, for each pair $(\alpha, \lambda_\Omega) \in \bar{\alpha} \times \bar{\lambda}_\Omega$, we construct a unique set of candidate tuning parameters for λ_β, say $\bar{\lambda}_\beta(\alpha, \lambda_\Omega)$. To construct this set, we first run the ECM algorithm with $(\alpha, \lambda_\Omega, \lambda_\beta) = (1, \lambda_\Omega, \infty)$ for all $\lambda_\Omega \in \bar{\lambda}_\Omega$. Note that since $\lambda_\beta = \infty$, the tuning parameter α does not affect the solution. Thus, we have $|\bar{\lambda}_\Omega|$ estimates of $\boldsymbol{\Omega}_*$, say $\hat{\boldsymbol{\Omega}}_{\lambda_\Omega}$, for each $\lambda_\Omega \in \bar{\lambda}_\Omega$. Let $\boldsymbol{X}_j \in \mathbb{R}^n$ be the jth column of $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, and let $\text{soft}(x, \tau) \equiv \max(|x| - \tau, 0)\text{sign}(x)$ be the soft-thresholding operator. Thus, for each $(\alpha, \lambda_\Omega) \in \bar{\alpha} \times \bar{\lambda}_\Omega$, the KKT conditions for (2.8) (with $\boldsymbol{\Omega}^{(k+1)}$ replaced with $\hat{\boldsymbol{\Omega}}_{\lambda_\Omega}$) imply that if λ_β is sufficiently large so that

$$\left\| \text{soft}\left(\frac{2}{n} \hat{\boldsymbol{\Omega}}_{\lambda_\Omega} \tilde{\boldsymbol{Y}}' \boldsymbol{X}_j, w_j \alpha \lambda_\beta \right) \right\|_2 \leq (1 - \alpha)\lambda_\beta, \quad \forall j \in [1 : p], \tag{1}$$

then the solution to (2.8) is the zeros matrix, where $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{n \times q}$ has ith row $\tilde{\boldsymbol{y}}_i$ for $i \in [n]$. That is, if we

repeated the E-steps and M-steps with tuning parameters λ_β and α, the solution would not change. Hence, we set $\bar{\lambda}_\beta(\alpha, \lambda_\Omega) \subseteq [\delta\lambda_\beta^\dagger(\alpha, \lambda_\Omega), \lambda_\beta^\dagger(\alpha, \lambda_\Omega)]$ where $\lambda_\beta^\dagger(\alpha, \lambda_\Omega)$ is the smallest value of λ_β such that (1) is satisfied for the pair (α, λ_Ω). If $n > p$, we set $\delta = 0.01$, otherwise, we set $\delta = 0.1$.

To fully make use of the warm starting approach, we compute the entire solution paths in a hierarchical manner as well. For each $(\alpha, \lambda_\Omega) \in \bar{\alpha} \times \bar{\lambda}_\Omega$, we compute the entire solution path for a decreasing sequence of $\dot{\lambda}_\beta$. Letting $\lambda_\beta^1 > \lambda_\beta^2 > \cdots > \lambda_\beta^K$, each time the ECM is initiated for some tuning parameter λ_β^j, the algorithm is initialized at the solution for λ_β^{j-1}.

Along the solution path, if the validation error exceeds some conservative threshold (relative to the smallest error observed along the solution path for large values of λ_β), we do not compute the estimator for all smaller values of λ_β for that particular pair (α, λ_Ω). In the simulation studies, if the validation R^2 for a particular λ_β is less than the maximum minus two standard errors (along the solution path for the pair (α, λ_Ω)), we do not compute the estimator for smaller values of λ_β. Thus, although we may consider a large number of tuning parameter combinations, we actually compute the estimator for a much smaller number in practice. In the tables below, we provide timing results (in minutes) for computing the entire ECM path (including determining tuning parameters and computing performance metrics) across all simulation settings considered in the main manuscript. Notably, these simulations were all performed using a single CPU per replication. If computing time were an issue and one had access to a distributed computing environment, cross-validation could be performed in an embarrassingly parallel fashion with each CPU computing the solution path for one set of $\bar{\lambda}_\beta(\alpha, \lambda_\Omega)$.

2 GTEx data

2.1 Pre-processing

We obtained RNA-seq data and whole genome-sequencing data from the GTEx Portal (gtexportal.org/home/, v7). We first filtered gene expression separately for each tissue based on read depth, keeping only those genes whose 75th percentile read depth is at least 20. We then kept the intersection of all genes with sufficient read depth across all 29 tissues. Then, we transformed expression using $\log_2\{(e_{i,j,k}+1)/q_{0.75}(e_{i,\cdot,k})\}$ where $e_{i,j,k}$ is the jth gene's read depth for the kth tissue for the ith subject, and $q_{.75}(e_{i,\cdot,k})$ is the 75th percentile of that subject's read depth within the kth tissue across all genes. Finally, we adjusted for age, gender, three genotype principal components, and PEER factors in the same way as described in Hu et al. (2019) (i.e., using more PEER factors for tissues with larger sample sizes). Specifically, we regressed the normalized expression onto these covariates, and used the residuals as our normalized gene expression data for eQTL mapping.

2.2 Assessment of normality

Our proposed method assumes that the processed data follow a multivariate normal distribution conditional on the SNP genotypes. The raw RNA-seq data are counts, which, as described in Section 2.1, are log-transformed and covariate adjusted. Assuming the normality of log-transformed counts can be reasonable when the counts are relatively large. For example, Costa-Silva et al. (2017) found little difference in differential expression testing when comparing using Poisson, negative binomial, and transformations to normality.

To illustrate that this assumption is reasonable, we display density plots of the processed RNA-seq data in Figure 1. In each of these genes, which we chose based on a list of most referenced genes in the literature (Dolgin, 2017), we see that the processed gene expression data do appear approximately normal. To verify that these genes are reasonably representative, we display the histograms of the median log-transformed RNA-seq

ρ	min	Q_1	Q_2	mean	Q_3	max
0.0	28.49	37.85	47.48	65.85	77.80	306.43
0.1	30.28	39.80	52.61	69.52	79.91	490.57
0.3	33.76	44.48	55.70	70.87	81.77	330.66
0.5	35.10	48.74	60.81	71.08	82.44	216.46
0.7	38.13	55.59	80.30	91.36	108.40	377.53

Table 1: Summary statistics for complete computing times (minutes) over 500 replications displayed in Figure 1(a) of the main manuscript.

eQTL	min	Q_1	Q_2	mean	Q_3	max
0.25	29.52	65.66	100.15	127.93	158.25	664.02
0.50	30.36	49.65	74.13	94.59	116.33	552.26
0.75	30.45	47.69	64.29	83.15	98.65	509.09
0.90	32.67	52.33	82.94	106.38	129.13	494.68
1	32.45	50.82	65.22	83.59	95.93	436.27

Table 2: Summary statistics for complete computing times (minutes) over 500 replications displayed in Figure 1(b) of the main manuscript. Note that eQTL (in the leftmost column) denotes the proportion of eQTLs which are shared.

R^2	min	Q_1	Q_2	mean	Q_3	max
0.01	30.70	38.73	43.67	56.78	62.66	220.48
0.05	32.05	45.02	64.67	83.17	98.75	551.32
0.10	30.45	47.69	64.29	83.15	98.65	509.09
0.20	44.28	112.19	150.17	181.42	229.54	617.38
0.40	55.13	85.19	120.12	133.69	173.44	502.20

Table 3: Summary statistics for complete computing times (minutes) over 500 replications displayed in Figure 1(c) of the main manuscript.

counts for four different tissues in Figure 2. Each of the colored vertical lines represent one of the four genes displayed in Figure 1. It is notable that these are either near the median in terms of log-tranformed counts, or spread across the quartiles. For example, in Whole Blood, these genes each fall in a different quartile, yet all of their expression appear approximately normal in Figure 1.

3 Note on the S-MultiXcan findings

It is important to note that significant associations discovered in this paper do not imply causality (Wainberg et al., 2019). However, as the analysis is concerned with testing of predicted gene expression, it has been shown that such associations are enriched in causal genes and these results should be useful for investigating the disease mechanisms (Barbeira et al., 2018). Further, when there is evidence for association with predicted expression of a gene, we may further perform co-localization analysis using methods proposed previously (Zhu et al., 2016; Giambartolomei et al., 2014; Hormozdiari et al., 2016; Wen et al., 2017) to examine whether any specific genetic variant is pleiotropic to both the gene expression and disease risk, which may provide evidence of possible causal relationship of the gene and disease.

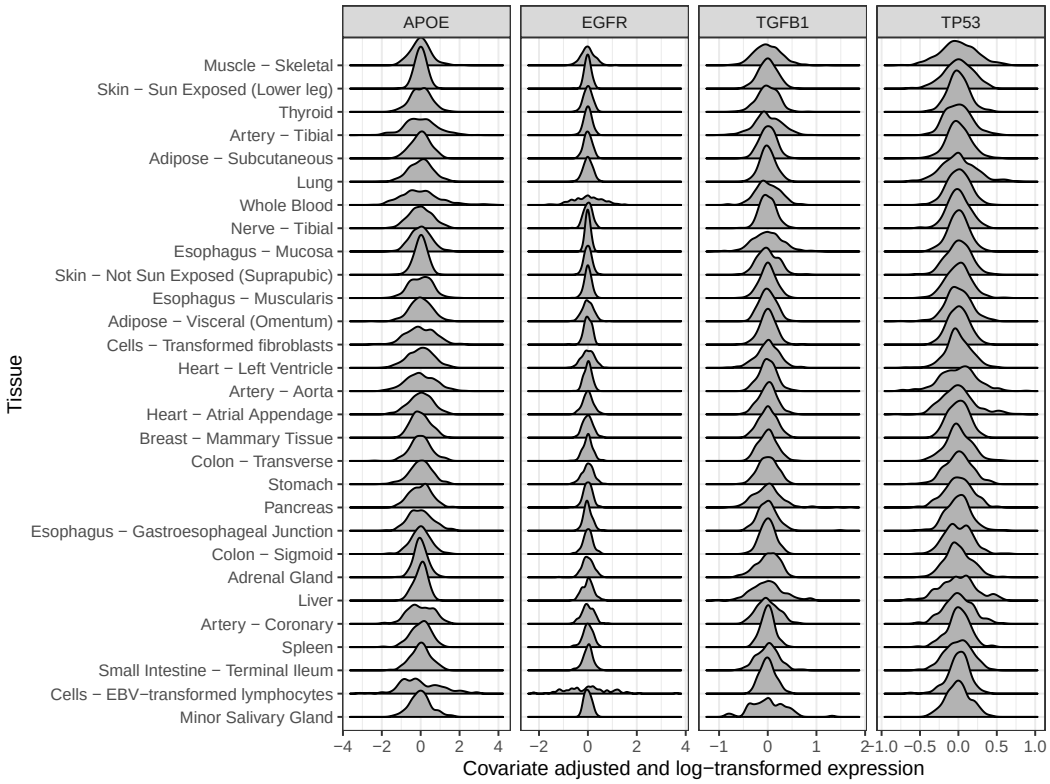


Figure 1: Density plots for the log-transformed and covariate adjusted gene expression data four genes. The log-transfrmation and covariate adjusted are described in Section 2.1.

4 Assessment of PrediXcan power and false discovery rate as a function of imputation accuracy

A reviewer wondered how prediction accuracy (i.e., imputation quality) of gene expression would affect on the downstream TWAS analysis. To illustrate that even modest improvements in imputation R^2 could lead to improved power in a PrediXcan analysis, we ran a series of additional simulation studies.

First, we start with the SNP genotype matrix we used in our simulation studies: $\boldsymbol{X} \in \mathbb{R}^{610\times 1178}$. Next we generate $\boldsymbol{\beta}$ in the same manner as in the simulation studies with 15 of 20 eQTLs shared across tissues. Given $\boldsymbol{X}$ and $\boldsymbol{\beta}$, we generate the gene expression matrix as a realization of

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}_Y, \quad \text{vec}(\boldsymbol{\epsilon}_Y) \sim \text{N}_{nq}(0, \boldsymbol{\Omega}_*^{-1}(R_Y^2) \otimes I_n)$$

where $\boldsymbol{\Omega}_*^{-1}(R_Y^2) \in \mathbb{S}_+^{29}$ is chosen so that the R^2 of the regression of $\boldsymbol{Y}$ on $\boldsymbol{X}$ is fixed at R_Y^2 for each component of the response.

Finally, we generate the trait against which we will test imputed gene expression as a realization of

$$\boldsymbol{T} = \boldsymbol{Y}\boldsymbol{\eta} + \boldsymbol{\epsilon}_T, \quad \boldsymbol{\epsilon}_T \sim \text{N}_n(0, \sigma_T^2(R_T^2) I_n),$$

where in setting (i) $\boldsymbol{\eta} \sim \text{N}_{29}(0, I_{29})$ and $\sigma_T^2(R_T^2) \in \mathbb{R}$ is set so that the R^2 for this regression is fixed at R_T^2. In setting (ii), $\boldsymbol{\eta} = 0$ (i.e., null) and $\sigma_T^2(R_T^2) \in [0.01, 5]$. In both settings, we then perform the PrediXcan analysis for 10^4 independently generated datasets. Specifically, for each dataset, we use an F-test to assess whether the

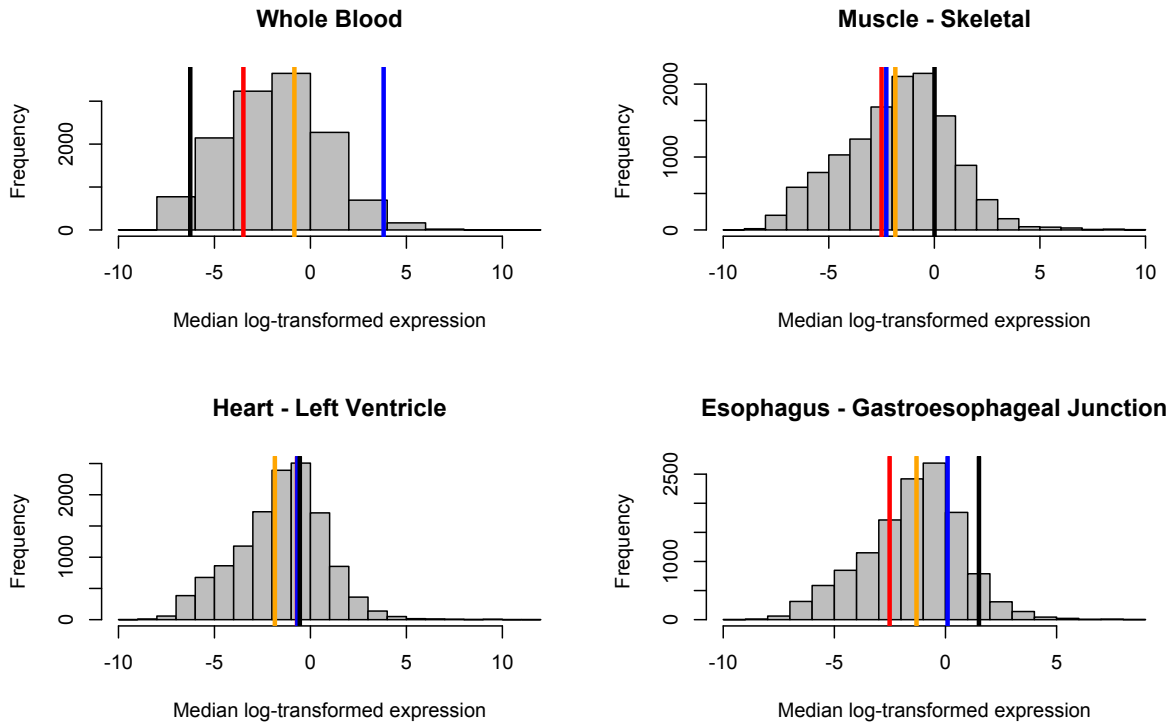


Figure 2: Histograms of median log-transformed expression across all genes for four tissues. The vertical lines denote the genes displayed in Figure 1: APOE (red), TGFBI (blue), EGFR (black), and TP53 (orange).

regression coefficients from the model of $\boldsymbol{T}$ (the trait) on $\boldsymbol{X\beta}$ (the imputed gene expression) are nonzero at the 0.05 level. Results are displayed in Figure 3.

The results displayed in this plot justify our claim that with improved prediction accuracy comes improved power to detect associations between $\boldsymbol{T}$ and $\boldsymbol{Y}$. Notably, in the leftmost panel, for a given R^2_T, when R^2_Y increases by even 0.01, the power can increase substantially. In the rightmost panel, we see that when no association exists between $\boldsymbol{T}$ and $\boldsymbol{Y}$, the value of R^2_Y appears to haveI no effect on the probability of a false discovery.

5 Computational details

5.1 Accelerated proximal gradient descent algorithm for (2.8)

To solve (2.8), we use an accelerated proximal gradient descent algorithm (Beck and Teboulle, 2009). The optimization problem can be expressed as

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^{p\times q}}{\arg\min}\left[n^{-1}\mathrm{tr}\{(\tilde{\boldsymbol{Y}}-\boldsymbol{X\beta})\boldsymbol{\Omega}(\tilde{\boldsymbol{Y}}-\boldsymbol{X\beta})'\}+g_{\lambda_\beta,\alpha}(\boldsymbol{\beta})\right], \tag{2}$$

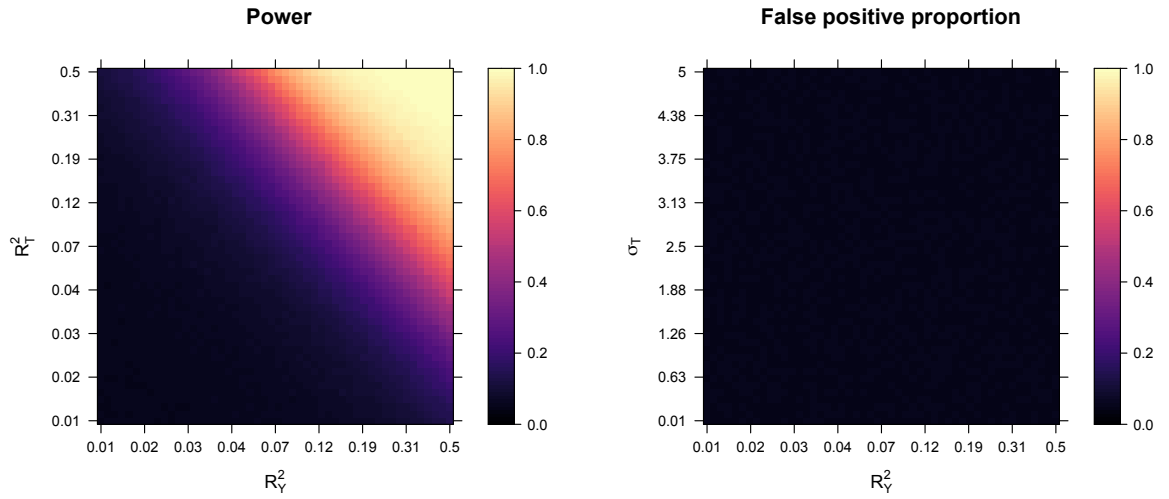


Figure 3: (Left) A heatmap displaying the power as R_T^2 and R_Y^2 vary under scenario (i). (Right) A heatmap showing the proportion of false positives as σ_T and R_Y^2 vary under scenario (ii). Note that in the right plane, the 25th percentile is 0.04850, the median 0.04990, and the 7th percentile is 0.05140.

where $\boldsymbol{\Omega}$, λ_β, and α are fixed. Let $\boldsymbol{\beta}^{(r)}$ denote the (r)th iterate of the algorithm we use to solve (2). As described in the main manuscript, our algorithm obtains the $(r+1)$th iterate of $\boldsymbol{\beta}$ using

$$\boldsymbol{\beta}^{(r+1)} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p\times q}}{\arg\min}\left\{\frac{1}{2}\|\boldsymbol{\beta}-\boldsymbol{\beta}^{(r)}+\gamma\nabla\ell_0(\boldsymbol{\beta}^{(r)})\|_F^2+\gamma g_{\lambda_\beta,\alpha}(\boldsymbol{\beta})\right\}.$$

Let $\varphi_1(A)$ denote the largest eigenvalue of a square matrix A, and let $A\otimes B$ denote the Kronecker product of matrices A and B. Thus, because the Hessian of $\mathrm{tr}\{(\tilde{\boldsymbol{Y}}-\boldsymbol{X}\boldsymbol{\beta})\boldsymbol{\Omega}(\tilde{\boldsymbol{Y}}-\boldsymbol{X}\boldsymbol{\beta})'\}$ with respect to $\mathrm{vec}(\boldsymbol{\beta})$ is $2(\boldsymbol{X}'\boldsymbol{X}\otimes\boldsymbol{\Omega})$, as long as $\gamma^{-1}n\geq 2\varphi_1(\boldsymbol{X}'\boldsymbol{X})\varphi_1(\boldsymbol{\Omega})$, the inequality in (2.9) is guaranteed to hold, and thus, the iterates of our algorithm have the *descent property* (i.e., monotonically decrease the objective function).

There are two common modifications of the off-the-shelf proximal gradient descent algorithm which can improve convergence speed. First, fixing γ at some quantity satisfying the inequality in (2.9) is often too conservative. The alternative, which we use in our implementation, is to select γ at each iteration using a backtracking line search. That is, if the new iterate satisfies an Armijo-type condition, we accept the iterate. If not, we reduce γ by some fixed factor, say 0.5, and recompute the iterate. Second, one can further speedup the algorithm by employing an acceleration technique based on momentum: see Beck and Teboulle (2009) for justification of this approach. Intuitively, this approach takes into account the direction in which the iterates changed from the $(r-1)$th to (r)th iteration. The accelerated version of this algorithm, when applied to L_1-penalized problems, is known as the fast iterative shrinkage-thresholding algorithm (FISTA, (Beck and Teboulle, 2009)). A version of the algorithm we implement is outlined in Algorithm 2

Algorithm 2: Accelerated proximal gradient algorithm for solving (2).

1. Initialize $\boldsymbol{\beta}^{(1)}=\boldsymbol{\beta}^{(0)}$, $\tau=\frac{1}{4}$, and $r=1$.
2. Update $\boldsymbol{\theta}^{(r+1)}\leftarrow\boldsymbol{\beta}^{(r)}+\tau(\boldsymbol{\beta}^{(r)}-\boldsymbol{\beta}^{(r-1)})$
3. Compute $\boldsymbol{\Delta}=\boldsymbol{\theta}^{(r+1)}-\gamma\frac{2}{n}(\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\theta}^{(r+1)}-\boldsymbol{X}'\tilde{\boldsymbol{Y}})\boldsymbol{\Omega}$ with γ chosen by backtracking line search.

4. Compute $\bar{\boldsymbol{\Delta}}_{j,k} = \max\left(|\boldsymbol{\Delta}_{j,k}| - \gamma\lambda_\beta\alpha\sqrt{\frac{n_{\max}}{n_k}}, 0\right)\text{sign}(\boldsymbol{\Delta}_{j,k})$ for all $(j,k) \in [p] \times [q]$

5. For each $j \in [p]$

   (a) If $\|\bar{\boldsymbol{\Delta}}_{j,\cdot}\|_2 \neq 0$

   – Update $\boldsymbol{\beta}_{j,\cdot}^{(r+1)} \leftarrow \max\left(1 - \frac{\gamma\lambda_\beta(1-\alpha)}{\|\bar{\boldsymbol{\Delta}}_{j,\cdot}\|_2}, 0\right)\bar{\boldsymbol{\Delta}}_{j,\cdot}$

   (b) Else

   – Update $\boldsymbol{\beta}_{j,\cdot}^{(r+1)} \leftarrow 0$

6. If objective function value has not converged, update $\tau \leftarrow \frac{r+1}{r+4}$, $\boldsymbol{\beta}^{(r)} \leftarrow \boldsymbol{\beta}^{(r+1)}$, and $\boldsymbol{\beta}^{(r-1)} \leftarrow \boldsymbol{\beta}^{(r)}$. Then, set $r = r + 1$ and return to Step 2.

5.2 Accelerated proximal gradient descent algorithm for exact version of Hu et al. (2019)

Notice, (3.1) can be written as a penalized weighted residual sum of squares estimator. Let $\mathcal{O}_k$ denote a subset of $[n]$ such that $i \in \mathcal{O}_k$ if $y_{i,k}$ is observed for $i \in [n]$. Let $\bar{\boldsymbol{y}}_i \in \mathbb{R}^q$ be a vector whose kth entry equals $y_{i,k}$ if $i \in \mathcal{O}_k$ and is 0 otherwise (i.e., we replace all missing entries with zero). Then, we can write the exact version of Hu et al. (2019) as

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^{p\times q}}{\arg\min}\left\{\frac{1}{2}\sum_{i=1}^{n}\text{tr}\left[(\bar{\boldsymbol{y}}_i - \boldsymbol{\beta}'\boldsymbol{x}_i)\boldsymbol{D}_{(i)}(\bar{\boldsymbol{y}}_i - \boldsymbol{\beta}'\boldsymbol{x}_i)'\right] + g_{\lambda_\beta,\alpha}(\boldsymbol{\beta})\right\}, \tag{3}$$

where $\boldsymbol{D}_{(i)} \in \mathbb{R}^{q\times q}$ is a diagonal matrix whose kth diagonal element can be expressed:

$$[\boldsymbol{D}_{(i)}]_{k,k} = \begin{cases} |\mathcal{O}_k|^{-1} & : i \in \mathcal{O}_k \\ 0 & : \text{otherwise} \end{cases}, \quad k \in [q]$$

Here, we have constructed the $\boldsymbol{D}_{(i)}$ in such a way that missing entries of the response do not affect the value of the objective function in (3). To solve (3), we will use a proximal gradient descent algorithm as in the previous subsection.

The gradient of the unpenalized loss function from (3) (i.e., the objective function from (3) with $\lambda_\beta = 0$) evaluated at $\boldsymbol{\beta}$ is

$$\sum_{i=1}^{n}\left(\boldsymbol{x}_i\boldsymbol{x}_i'\boldsymbol{\beta}\boldsymbol{D}_{(i)} - \boldsymbol{x}_i\bar{\boldsymbol{y}}_i'\boldsymbol{D}_{(i)}\right).$$

If the unpenalized objective function from (3) has Lipschitz continuous gradient, then we can solve (3) by simply replacing the gradient appearing in Step 3 of Algorithm 2 with the gradient above. Thus, we need only verify that the unpenalized loss function has Lipschitz continuous gradient.

To find a Lipschitz constant, notice that the Hessian of the unpenalized loss function from (3) with respect to $\text{vec}(\boldsymbol{\beta})$ is

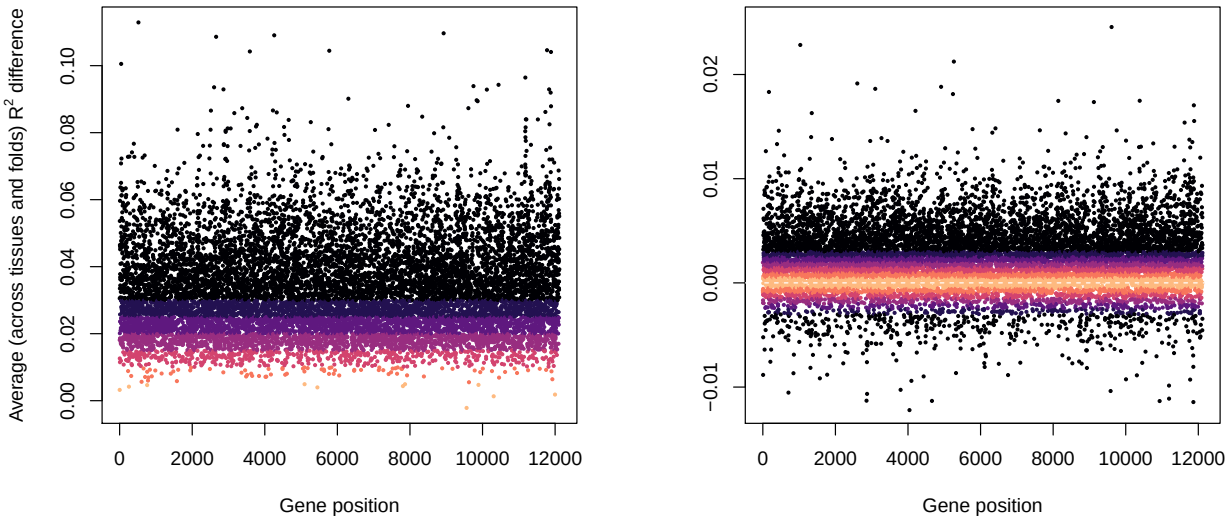
$$\sum_{i=1}^{n}\left(\boldsymbol{D}_{(i)} \otimes \boldsymbol{x}_i\boldsymbol{x}_i'\right).$$

Thus, we need only find an upper bound on the largest eigenvalue of

$$\sum_{i=1}^{n}(\boldsymbol{D}_{(i)} \otimes \boldsymbol{x}_i\boldsymbol{x}_i') = \begin{pmatrix} n_1^{-1}\sum_{i\in\mathcal{O}_1}\boldsymbol{x}_i\boldsymbol{x}_i' & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & n_q^{-1}\sum_{i\in\mathcal{O}_q}\boldsymbol{x}_i\boldsymbol{x}_i' \end{pmatrix}.$$

Because the eigenvalues of a block diagonal symmetric matrix are the eigenvalues of each diagonal block matrix, we know that if $L = \max_k \varphi_1(n_k^{-1}\sum_{i\in\mathcal{O}_k}\boldsymbol{x}_i\boldsymbol{x}_i')$, the gradient of unpenalized objective from (3) is Lipschitz continuous with constant L.

6 Additional figures



(a) `Cov-MT` minus `EN` (b) `Cov-MT` minus `MT`

Figure 4: Differences in left-out fold R^2 averaged across five folds and 29 tissues for each gene. In (a), we display the difference using our method minus using `EN`. In (b), we display the difference using our method minus using `MT`. In both, color gradients are used to improve visualization – darker colors are farther from zero. In (b), two points were omitted for ease of display: they had vertical axis values $-.0228$ and $-.0244$ at horizontal axis positions 53 and 3717.

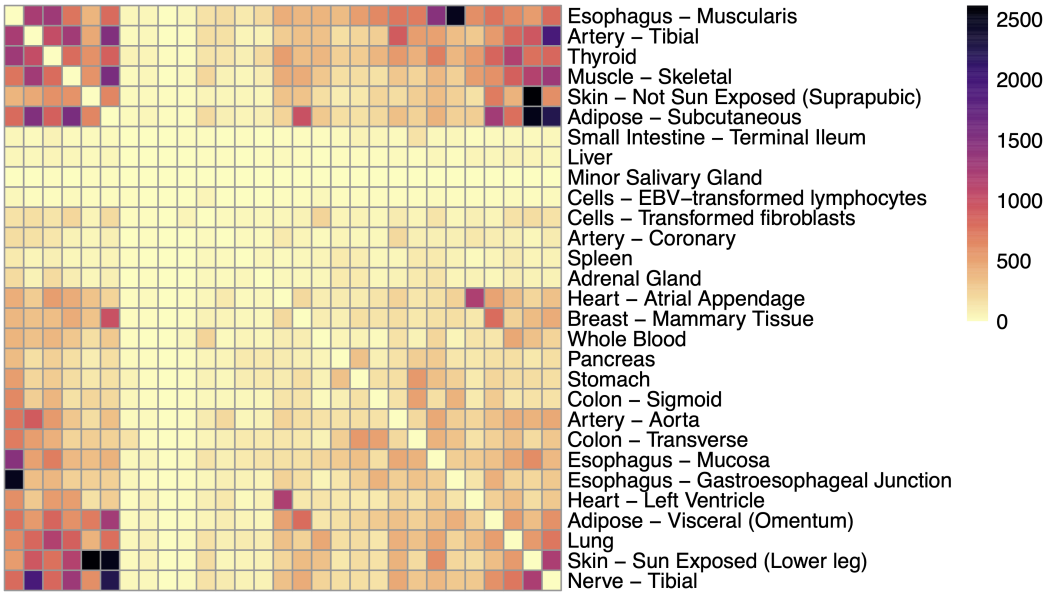


Figure 5: A heatmap with frequencies of nonzero entries in the estimates of $\boldsymbol{\Omega}_*$. The diagonal has been removed. Tissues are arranged to better visualize pairs which frequently had estimated nonzero conditional correlations. Note that there were 12116 genes included in our analysis.

References

Barbeira, A. N., Dickinson, S. P., Bonazzola, R., Zheng, J., Wheeler, H. E., Torres, J. M., Torstenson, E. S., Shah, K. P., Garcia, T., and Edwards, T. L. (2018). Exploring the phenotypic consequences of tissue specific gene expression variation inferred from GWAS summary statistics. *Nature Communications*, 9(1):1825.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Costa-Silva, J., Domingues, D., and Lopes, F. M. (2017). Rna-seq differential expression analysis: An extended review and a software tool. *PloS one*, 12(12):e0190152.

Dolgin, E. (2017). The most popular genes in the human genome. *Nature*, 551(7681):427–432.

Giambartolomei, C., Vukcevic, D., Schadt, E. E., Franke, L., Hingorani, A. D., Wallace, C., and Plagnol, V. (2014). Bayesian test for colocalisation between pairs of genetic association studies using summary statistics. *PLoS Genetics*, 10(5):e1004383.

Hormozdiari, F., Van De Bunt, M., Segre, A. V., Li, X., Joo, J. W. J., Bilow, M., Sul, J. H., Sankararaman, S., Pasaniuc, B., and Eskin, E. (2016). Colocalization of GWAS and eQTL signals detects target genes. *The American Journal of Human Genetics*, 99(6):1245–1260.

Hu, Y., Li, M., Lu, Q., Weng, H., Wang, J., Zekavat, S. M., Yu, Z., Li, B., Gu, J., Muchnik, S., et al. (2019). A statistical framework for cross-tissue transcriptome-wide association analysis. *Nature Genetics*, 51(3):568–576.

Wainberg, M., Sinnott-Armstrong, N., Mancuso, N., Barbeira, A. N., Knowles, D. A., Golan, D., Ermel, R.,

Ruusalepp, A., Quertermous, T., Hao, K., et al. (2019). Opportunities and challenges for transcriptome-wide association studies. *Nature Genetics*, 51(4):592.

Wen, X., Pique-Regi, R., and Luca, F. (2017). Integrating molecular QTL data into genome-wide genetic association analysis: Probabilistic assessment of enrichment and colocalization. *PLoS Genetics*, 13(3):e1006646.

Witten, D. M., Friedman, J. H., and Simon, N. (2011). New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900.

Zhu, Z., Zhang, F., Hu, H., Bakshi, A., Robinson, M. R., Powell, J. E., Montgomery, G. W., Goddard, M. E., Wray, N. R., Visscher, P. M., et al. (2016). Integration of summary data from GWAS and eQTL studies predicts complex trait gene targets. *Nature Genetics*, 48(5):481.