

A flexible empirical Bayes approach to multivariate multiple regression, and its improved accuracy in predicting multi-tissue gene expression from genotypes—S1 Text

Fabio Morgante^{1,2,3*}, Peter Carbonetto^{4,5}, Gao Wang^{4,6,7}, Yuxin Zou^{8,9}, Abhishek Sarkar^{4,✉}, Matthew Stephens^{4,8*}

1 Center for Human Genetics, Clemson University, Greenwood, South Carolina, United States of America

2 Department of Genetics and Biochemistry, Clemson University, Clemson, South Carolina, United States of America

3 Section of Genetic Medicine, Department of Medicine, University of Chicago, Chicago, Illinois, United States of America

4 Department of Human Genetics, University of Chicago, Chicago, Illinois, United States of America

5 Research Computing Center, University of Chicago, Chicago, Illinois, United States of America

6 Department of Neurology, Columbia University, New York, New York, United States of America

7 Gertrude. H. Sergievsky Center, Columbia University, New York, New York, United States of America

8 Department of Statistics, University of Chicago, Chicago, Illinois, United States of America

9 Regeneron Genetics Center, Regeneron Pharmaceuticals Inc., Tarrytown, New York, United States of America

✉ Current address: Vesalius Therapeutics, Inc., Cambridge, Massachusetts, United States of America

A. Conventions used in mathematical expressions

For the mathematical expressions below, we denote matrices as bold, uppercase letters (e.g., \mathbf{A}), column vectors as bold, lowercase letters (\mathbf{a}), and scalars are written in plain, lowercase letters (a). We use $\mathbb{R}^{m \times n}$ to denote the set of real $m \times n$ matrices, we use \mathbf{I}_n to denote the $n \times n$ identity matrix, $\mathbf{1}_n$ is a column-vector of ones of length n , and $\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$ denotes the trace of $n \times n$ matrix \mathbf{A} .

For describing Bayesian calculations, our convention is to use the “0” subscript to indicate priors and the “1” subscript to indicate posteriors.

B. Preparation of GTEx data

Here we describe the steps that were taken to prepare the GTEx gene expression and genotype data for *mr.mash*. These same steps were taken for both the application and the simulations, but note that only the genotype data were used in the simulations.

The GTEx v8 data were used for all the analyses. Files containing expression data (RNASeq QC version 1.1.9 TPM), covariate data and genotypes (VCF format) were downloaded from dbGaP, study accession phs000424.v8.p2. The raw expression and genotype data were preprocessed using data processing scripts available online. The data processing scripts used are available at: https://github.com/gaow/mvarbvs/blob/master/analysis/gtex-v7/20170929_Expression_Covariate.ipynb, https://github.com/gaow/mvarbvs/blob/master/analysis/gtex-v7/20170929_Genotype.ipynb, and https://github.com/gaow/mvarbvs/blob/master/workflow/GTEx_V8_preprocessing.ipynb. All these scripts are intended to be run using the SoS (Script of Scripts) framework available at <https://github.com/vatlab/sos>. These scripts were all run with the default parameters, e.g., `sos run GTEx_V8_preprocessing.ipynb`. These scripts were used to extract and reformat the genotype and expression data from the GTEx v8 data files for analysis with *mr.mash* and the other methods. Data processing steps included reformatting the data, removing the effects of selected covariates from the gene expression levels, and selecting biallelic genetic variants 1 Mb upstream and downstream of the transcription start site (TSS) for each

gene. The result of running this pipeline was an R object (`.rds`) file for all genes (including pseudo-genes) on chromosomes 1–22. In total, 39,840 `.rds` files were generated. Each file contained an $n \times p$ genotype matrix \mathbf{X} and an $n \times r$ matrix of gene expression residuals \mathbf{Y} , possibly containing missing values. For a given gene, the number of samples (n) was at most 848, and the number of tissues (r) was at most 49. (Data preparation done as part of the GTEx project included filtering out tissues for each gene (see <https://gtexportal.org/home/methods> for details), so r was often less than 49 for a given gene.) We also removed samples in which expression measurements were missing in all tissues.

After running this data processing pipeline, the following additional filtering steps were applied, separately for each gene, to the genotype data \mathbf{X} before running *mr.mash* or one of the other multivariate regression methods. Genetic variants satisfying these conditions were retained: genetic variants with minor allele frequency (MAF) greater than 0.05 (these are MAFs calculated from the GTEx genotypes); missing genotype rate less than 0.05; and genotype variance greater than 0.05. For any genetic variants that had one or more missing genotypes, the missing genotypes were filled in using the mean genotype for that variant.

After taking these steps, on average there were 4,465 genetic variants for a given gene, but this number varied widely from gene to gene, from 41 to 21,247. The number of tissues, r , varied from 1 to 49, with a median of 43. The number of samples, n , varied from 73 to 838, with a median of 836. The rate of missing entries in \mathbf{Y} ranged from 0% to 77%, with a median of 63%.

Finally, we removed the “kidney cortex” tissue from \mathbf{Y} because it had a maximum sample size less than 100.

C. Simulations with GTEx genotypes

In brief, for each simulation, we generated multi-tissue expression data sets for 2,000 genes randomly selected from chromosomes 1–22. We only considered genes in which all 838 genotype samples were retained by the data processing pipeline above.

For each gene, we simulated the $n \times r$ gene expression matrix \mathbf{Y} for $n = 838$ samples and $r = 10$ tissues using the multivariate multiple regression model. The coefficients \mathbf{B} and the residual covariance matrix \mathbf{V} used to simulate the phenotypes were chosen so as to capture a variety of genetic architectures: different numbers of genetic variants affecting the phenotypes; different patterns of effect sharing across tissues; and different levels of genomic heritability (including no heritability—that is, simulations in which one or more expression levels were unaffected by genotype). Once we simulated the expression levels, we split the samples into two subsets, one containing 80% of the samples (a training set with $n = 670$) and another containing 20% of the samples (a test set with $n = 168$).

The training data for the 2,000 genes were used to estimate data-driven covariance matrices from the *mr.mash* prior (see Sec. E.4.3). Then we selected 1 out of the 2,000 genes, and fit regression models (*Elastic Net*, *mr.mash*, *etc*) to the genotype and phenotype data from the training set for that gene. Finally, we evaluated the accuracy of each model’s predictions in the test samples for that same gene.

C.1. Simulation scenarios

The five simulation scenarios are described briefly in the main text, and here we describe them in greater detail.

- A. In the “Equal Effects” scenario, a causal variant affected all tissues, and the effects were exactly the same in all tissues. This was the setting in which we had most to gain by analyzing all tissues simultaneously compared to analyzing the tissues one at a time.
- B. In the “Independent Effects” scenario, a causal variant affected all tissues, and the effects were independent across all tissues (more precisely, the effects were independent conditioned on the genetic variant being a causal variant). The Independent Effects scenario represented the situation in which there was less to gain from a joint analysis of all tissues because the effect sizes were not shared across tissues. Nonetheless, we expected some gain in a multivariate analysis compared to a tissue-by-tissue analysis because knowing that a genetic variant has an effect in one tissue was helpful for detecting an effect in other tissues.

- C. In the “Mostly Null” scenario, a causal variant only affected the first tissue, and therefore the remaining tissues were unaffected by genotype. This represented a scenario in which the effects for the given gene were tissue-specific. We did not expect any gain in performing a joint analysis of all tissues in this setting because there was no sharing of effects across tissues. In fact, there was the potential of multivariate analysis methods to harm performance by incorrectly shrinking effect estimates for the first tissue toward effect estimates in the remaining “null” tissues.
- D. In the “Equal Effects + Null” scenario, the effects were equal across tissue 1 through 3, and there were no effects in tissues 4 through 10.
- E. In the “Shared Effects in Subgroups” scenario, effects were drawn from a mixture of effect-sharing patterns: 50% of the time, the effects were shared (unequally) across tissues 1 through 3 and explained 20% of the variance of each tissue; in the other 50% of the time, the effects are shared (unequally) in tissues 4 through 10 and explained only 5% of the variance of each tissue. This scenario was intended to correspond roughly to the patterns of effect sharing estimated from previous multivariate analyses of GTEx Project data (see for example Fig. 3a in Uebachs et al. 34).

In the first four scenarios, the genetic effects explain 20% of the variance of each tissue. (Except in the Mostly Null scenario, where the genetic effects explained none of the variance in all tissues other than the first, and in the Equal Effects + Null scenario, where the genetic effects explained none of the variance in tissues 4 through 10.)

C.2. Simulation procedure

For all simulation scenarios, we used the following procedure to simulate the multi-tissue expression. For all simulations, $n = 838$ and $r = 10$.

1. Choose a gene at random.
2. Center the columns of the $n \times p$ genotype matrix \mathbf{X} so that each column has a mean of zero.
3. Choose the number of causal variants, p_{causal} , uniformly at random among integers 1–10.
4. Choose the p_{causal} causal variants uniformly at random from $\{1, \dots, p\}$.
5. Set $\mathbf{b}_j = (0, \dots, 0)^T \in \mathbb{R}^r$ for each non-causal variant j .
6. For each causal variant j , simulate $\mathbf{b}_j \in \mathbb{R}^r$: for the Independent Effects scenario, $\mathbf{b}_j \sim N_r(\mathbf{0}, \mathbf{I}_r)$; for the Equal Effects scenario, $\mathbf{b}_j \sim N_r(\mathbf{0}, \mathbf{1}_r \mathbf{1}_r^T)$, where $\mathbf{1}_r = (1, \dots, 1)^T \in \mathbb{R}^r$; for the Mostly Null scenario, $\mathbf{b}_j \sim N_r(\mathbf{0}, \mathbf{S})$, where \mathbf{S} is an $r \times r$ matrix of all zeros except for a single one, $s_{11} = 1$; for the Equal Effects + Null scenario, $\mathbf{b}_j \sim N_r(\mathbf{0}, \mathbf{S})$, where \mathbf{S} is an $r \times r$ matrix of all zeros except for all ones in the 3×3 top-right subblock, $s_{ij} = 1$, $i \leq 3, j \leq 3$; for the Shared Effects in Subgroups scenario, the effects \mathbf{b}_j were simulated using a mixture of zero-centered multivariate normals. The mixture weights and covariance matrices for this scenario are given in the Zenodo repository [28].
7. Form the residual covariance matrix across tissues, \mathbf{V} . This is a diagonal $r \times r$ matrix in which the diagonal entries v_{ii} are chosen to achieve the target PVE (proportion of variance explained) for each tissue. For tissues i in which genetic variants did not explain variation in expression, we set $v_{ii} = 1$.
8. Simulate the $n \times r$ matrix of gene expression levels, $\mathbf{Y} \sim MN(\mathbf{B}_0 + \mathbf{X}\mathbf{B}, \mathbf{I}_n, \mathbf{V})$, where \mathbf{B}_0 is an $n \times r$ matrix of ones, \mathbf{B} is the $p \times r$ coefficients matrix with rows \mathbf{b}_j , and \mathbf{I}_n is the $n \times n$ identity matrix.
9. Randomly split the data—rows of \mathbf{X} and rows of \mathbf{Y} —into a training set (80% of the data, or 670 individuals) and a test set (20% of the data, or 168 individuals). This training-test split was different for each scenario, but the same for all simulations in a scenario.

Algorithm 1 Sketch of the *mr.mash* variational empirical Bayes algorithm allowing for missing data. The matrix \mathbf{Y} may contain missing values; \mathbf{Y}_{obs} denotes the set of observed values. The matrix \mathbf{X} may be “standardized”—columns scaled to unit standard deviation—which typically reduces computation because more calculations can be reused. In the algorithm, $\tilde{\mathbf{x}}_j$ denotes the j th column of $\tilde{\mathbf{X}}$, and \mathbf{b}_j is the j th row of \mathbf{B} (stored as a column vector). “BMSR-mix” is explained in the text. Note this is only an outline of the algorithm, and actual implementation differs to avoid redundant calculations and reduce memory usage.

Require: Data inputs \mathbf{X} ($n \times p$ matrix) and \mathbf{Y} ($n \times r$ matrix).

Require: Prior covariances $\mathcal{S}_0 = \{\mathbf{S}_{0,1}, \dots, \mathbf{S}_{0,K}\}$.

Require: Initial estimates $\mathbf{b}_0, \mathbf{B}, \mathbf{w}_0, \mathbf{V}$.

repeat

Center the columns of \mathbf{X} and store the result in $\tilde{\mathbf{X}}$.

If \mathbf{Y} contains missing values, impute the missing values using $\mathbf{Y}_{\text{obs}}, \mathbf{b}_0, \mathbf{B}, \mathbf{V}$.

Store the observed and imputed values in \mathbf{Y}_{full} .

Center the columns of \mathbf{Y}_{full} and the store result in $\tilde{\mathbf{Y}}$.

Compute the expected residuals, $\mathbf{R} \leftarrow \tilde{\mathbf{Y}} - \tilde{\mathbf{X}}\mathbf{B}$.

for j in $1, \dots, p$ **do**

Remove variable j from the expected residuals, $\mathbf{R}_j \leftarrow \mathbf{R} + \tilde{\mathbf{x}}_j \mathbf{b}_j^\top$.

$(w_{1,1}^{(j)}, \dots, w_{1,K}^{(j)}, \mathbf{b}_{1,1}^{(j)}, \dots, \mathbf{b}_{1,K}^{(j)}, \mathbf{S}_{1,1}^{(j)}, \dots, \mathbf{S}_{1,K}^{(j)}) \leftarrow \text{BMSR-mix}(\mathbf{x}_j, \mathbf{R}_j, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0)$.

Compute the posterior mean estimates, $\mathbf{b}_j = \sum_{k=1}^K w_{1,k}^{(j)} \mathbf{b}_{1,k}^{(j)}$.

Store \mathbf{b}_j in j th column of \mathbf{B} .

Include variable j in the expected residuals, $\mathbf{R} \leftarrow \mathbf{R}_j - \tilde{\mathbf{x}}_j \mathbf{b}_j^\top$.

Update the prior, $w_{0,k} \leftarrow \sum_{j=1}^p w_{1,k}^{(j)} / p$.

Update the residual covariance matrix, $\mathbf{V} \leftarrow E_q[(\mathbf{Y}_{\text{full}} - \tilde{\mathbf{X}}\mathbf{B})^\top (\mathbf{Y}_{\text{full}} - \tilde{\mathbf{X}}\mathbf{B})] / n$.

Update the intercept, $\mathbf{b}_0 \leftarrow (\mathbf{Y}_{\text{full}} - \mathbf{X}\mathbf{B})^\top \mathbf{1}_n / n$.

until convergence criterion is met or maximum iteration is reached

return Intercept \mathbf{b}_0 , coefficients \mathbf{B} , imputed responses \mathbf{Y}_{full} , prior weights \mathbf{w}_0 , and residual covariance \mathbf{V} .

C.3. Computing environment

All simulations were run on Linux machines (CentOS Linux 8.4-2105) with Intel Xeon Gold 6248R (“Cascade Lake”) processors. We used R 4.0.3 [32] linked to OpenBLAS 0.3.12 (conda package r-base 4.0.3, build h349a78a.8 installed via conda-forge). At most 4 GB of memory was needed to run *mr.mash* or one of the other methods on a simulated data set. All methods and other computations were run on a single CPU.

D. Algorithms for fitting the *mr.mash* model

An outline of the variational empirical Bayes algorithm for *mr.mash* is given in Algorithm 1. This algorithm also handles imputation of missing data which we describe in later sections. Algorithm 2 gives a more detailed description of the *mr.mash* algorithm with no missing data. The algorithm for \mathbf{Y} with missing values is given in Algorithm 3.

A brief derivation of Algorithm 2 is given in the next two subsections, and more detailed derivations are given in Sections F and G. The algorithm for *mr.mash* with missing data is derived in Sec. H.

The steps of Algorithm 2 that involve the most effort are the updates to the posteriors $q_j(\mathbf{b}_j)$ in the inner loop over j , and the update to \mathbf{V} . The update to \mathbf{V} involves computing the ERSS (13), which involves $O((n+p)r^2)$ flops. Each update to $q_j(\mathbf{b}_j)$ involves computing $\tilde{\mathbf{R}}_j$ and the BMSR-mix posterior, requiring $O(nr + kr^3)$ flops. Therefore, the computational complexity of a single update of all the parameters (a single iteration of the outer loop) is $O(p(nr + kr^3) + nr^2)$. The effort in the inner-loop updates is greatly reduced when \mathbf{X} is standardized because the most expensive computations can be reused (Sec. I). In this special case, the complexity of a single outer-loop iteration reduces to $O(p(nr + kr^2) + nr^2 + kr^3)$. The complexity of the missing data imputation step in Algorithm 3 in

Algorithm 2 *mr.mash* with no missing data

Require: $n \times p$ data matrix \mathbf{X} and $n \times r$ data matrix \mathbf{Y} . Columns of \mathbf{X} and \mathbf{Y} should be “centered” so that each column has a mean of zero. Optionally, \mathbf{X} may also be “standardized”—that is, scaled so that each column has unit standard deviation—which simplifies downstream posterior computations (Sec. I).

Require: Set of K covariance matrices, \mathcal{S}_0 .

Require: Initial estimates of the posterior mean coefficients, stored as a $p \times r$ matrix, $\bar{\mathbf{B}}$.

Require: Initial estimates of the prior mixture weights, $\mathbf{w}_0 = (w_{0,1}, \dots, w_{0,K})$, and the $r \times r$ residual covariance matrix \mathbf{V} .

Require: Convergence threshold, $\text{tol} \geq 0$, and an upper limit on the number of iterations, t_{\max} .

- 1: $t \leftarrow 0$
- 2: $\delta \leftarrow \infty$
- 3: $\text{ELBO}^{(0)} \leftarrow F(q; \mathbf{V}, \mathbf{w}_0)$
- 4: **while** $\delta > \text{tol}$ and $t < t_{\max}$ **do**
- 5: $t \leftarrow t + 1$
- 6: Compute expected residuals, $\bar{\mathbf{R}} \leftarrow \mathbf{Y} - \mathbf{X}\bar{\mathbf{B}}$
- 7: Initialize \mathbf{W} to a $p \times K$ matrix of zeros
- 8: **for** j in $1, \dots, p$ **do**
- 9: Remove variable j from expected residuals, $\bar{\mathbf{R}}_j \leftarrow \bar{\mathbf{R}} + \mathbf{x}_j \bar{\mathbf{b}}_j^\top$
- 10: $(w_{1,1}^{(j)}, \dots, w_{1,K}^{(j)}, \mathbf{b}_{1,1}^{(j)}, \dots, \mathbf{b}_{1,K}^{(j)}, \mathbf{S}_{1,1}^{(j)}, \dots, \mathbf{S}_{1,K}^{(j)}) \leftarrow \text{BMSR-mix}(\mathbf{x}_j, \bar{\mathbf{R}}_j, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0)$ \triangleright See (58).
- 11: Compute posterior mean, $\bar{\mathbf{b}}_j = \sum_{k=1}^K w_{1,k}^{(j)} \mathbf{b}_{1,k}^{(j)}$
- 12: Compute posterior covariance, $\mathbf{S}_j = \sum_{k=1}^K w_{1,k}^{(j)} [\mathbf{b}_{1,k}^{(j)} (\mathbf{b}_{1,k}^{(j)})^\top + \mathbf{S}_{1,k}^{(j)}] - \bar{\mathbf{b}}_j \bar{\mathbf{b}}_j^\top$
- 13: Store $w_{1,1}^{(j)}, \dots, w_{1,K}^{(j)}$ in the j th row of \mathbf{W}
- 14: Include variable j in expected residuals, $\bar{\mathbf{R}} \leftarrow \bar{\mathbf{R}}_j - \mathbf{x}_j \bar{\mathbf{b}}_j^\top$
- 15: Update prior weights, $\mathbf{w}_0 \leftarrow \text{UPDATE-WEIGHTS}(\mathbf{W})$ \triangleright See (14).
- 16: Update residual covariance, $\mathbf{V} \leftarrow \text{UPDATE-RESID-COV}(\mathbf{X}, \bar{\mathbf{R}}, \mathbf{S}_1, \dots, \mathbf{S}_p)$ \triangleright See (12).
- 17: $\text{ELBO}^{(t)} \leftarrow F(q; \mathbf{V}, \mathbf{w}_0)$
- 18: $\delta \leftarrow \text{ELBO}^{(t)} - \text{ELBO}^{(t-1)}$
- return** $\bar{\mathbf{B}}, \mathbf{V}, \mathbf{w}_0, \text{ELBO}^{(t)}$

$O(nr^3)$.

D.1. Computing the posterior distribution of the regression coefficients

Our aims are to (i) compute posterior estimates of the *mr.mash* unknowns and (ii) adapt the *mr.mash* priors to the data. Both (i) and (ii) are computationally challenging in multiple linear regression models, especially in a multivariate framework such as *mr.mash*. (With some exceptions such as ridge regression.) Therefore, some numerical approximations are required. The most widely used numerical approximations are based on Monte Carlo methods, most commonly Markov chain Monte Carlo (MCMC) and variants of MCMC [9, 11, 13, 16, 18, 22, 19, 27, 29, 31, 40, 41]. While MCMC and other Monte Carlo methods have allowed for a proliferation of different models for prediction, Monte Carlo methods typically involve a high computational effort, and therefore may limit our ability to apply *mr.mash* to genetic data sets with many genetic variants and many phenotypes. Another approach is based on stochastic search methods [1, 4, 5, 6, 21, 36]. But stochastic search only focuses on identifying highest-probability configurations, and therefore these approaches are not well suited to predicting \mathbf{Y} and adapting the priors.

An alternative approach to MCMC is variational inference (VI) [3, 24]. In VI, we treat posterior inference as an optimization problem; the aim is to find a distribution, $q(\mathbf{B})$, within a prescribed family of distributions, \mathcal{Q} , that best approximates the *mr.mash* posterior, $p(\mathbf{B} \mid \mathbf{X}, \mathbf{Y}, \mathbf{V}, \mathbf{w}_0, \mathcal{S}_0)$. The closeness of the approximate distribution to the true posterior is measured by the Kullback-Leibler

Algorithm 3 *mr.mash* with missing data

Require: $n \times p$ data matrix \mathbf{X} (does not need to be centered). Optionally, \mathbf{X} may be “standardized”—that is, scaled so that each column has unit standard deviation—which simplifies downstream posterior computations (see Sec. I).

Require: Set of K covariance matrices, \mathcal{S}_0 .

Require: Initial estimates of the posterior mean coefficients, stored as a $p \times r$ matrix, $\bar{\mathbf{B}}$, and initial estimate of the intercept, $\bar{\mathbf{b}}_0$ (a vector of length r).

Require: Initial estimate of \mathbf{Y}_{full} , the $n \times r$ response matrix containing the observed values and initial estimates of missing values.

Require: Initial estimates of the prior mixture weights, $\mathbf{w}_0 = (w_{0,1}, \dots, w_{0,K})$, and the $r \times r$ residual covariance matrix \mathbf{V} .

- 1: **while** convergence criterion is not met **do**
- 2: Center columns of \mathbf{X} and store result in $\tilde{\mathbf{X}}$
- 3: Impute missing values as posterior mean estimates given $\mathbf{b}_0, \bar{\mathbf{B}}, \mathbf{V}$, and store observed and missing values in \mathbf{Y}_{full} (see eqs. 81, 82).
- 4: Center columns of \mathbf{Y}_{full} and store result in $\tilde{\mathbf{Y}}$
- 5: Run Algorithm 2 with $t_{\text{max}} \leftarrow 1$, in which \mathbf{X}, \mathbf{Y} are set to $\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}$
- 6: Update intercept, $\hat{\mathbf{b}}_0 \leftarrow \frac{1}{n}(\mathbf{Y}_{\text{full}} - \mathbf{X}\bar{\mathbf{B}})^\top \mathbf{1}_n$ (see eq. 78)

return $\hat{\mathbf{b}}_0, \bar{\mathbf{B}}, \mathbf{Y}_{\text{full}}, \mathbf{V}, \mathbf{w}_0$

(K-L) divergence, so the VI optimization problem is

$$\begin{aligned} & \text{minimize} && D_{\text{KL}}(q(\mathbf{B}) \parallel p(\mathbf{B} \mid \mathbf{X}, \mathbf{Y}, \mathbf{V}, \mathbf{w}_0)) \\ & \text{subject to} && q \in \mathcal{Q}, \end{aligned} \quad (5)$$

where $D_{\text{KL}}(q \parallel p)$ denotes the K-L divergence between distributions q and p . (In this expression and in the expressions below, we omit the dependence on \mathcal{S}_0 since it is a prespecified model parameter and does not change.)

Since the K-L divergence is itself intractable, we instead work with a different objective function that is easier to compute but yields the same solutions as (5). This alternative objective is the “Evidence Lower Bound” (ELBO) [3]:

$$F(q; \mathbf{V}, \mathbf{w}_0) := \log p(\mathbf{Y} \mid \mathbf{V}, \mathbf{w}_0) - D_{\text{KL}}(q(\mathbf{B}) \parallel p(\mathbf{B} \mid \mathbf{X}, \mathbf{Y}, \mathbf{V}, \mathbf{w}_0)) \quad (6)$$

Since the marginal likelihood $p(\mathbf{Y} \mid \mathbf{V}, \mathbf{w}_0)$ does not depend on q , minimizing the K-L divergence with respect to q is equivalent to maximizing the ELBO:

$$\underset{q \in \mathcal{Q}}{\operatorname{argmin}} D_{\text{KL}}(q(\mathbf{B}) \parallel p(\mathbf{B} \mid \mathbf{X}, \mathbf{Y}, \mathbf{V}, \mathbf{w}_0)) = \underset{q \in \mathcal{Q}}{\operatorname{argmax}} F(q; \mathbf{V}, \mathbf{w}_0) \quad (7)$$

The advantage of the ELBO, however, is that, by straightforward algebraic manipulations the ELBO can be rewritten as

$$F(q; \mathbf{V}, \mathbf{w}_0) = E_q[\log p(\mathbf{Y} \mid \mathbf{X}, \mathbf{B}, \mathbf{V}, \mathbf{w}_0)] + E_q\left[\log \left(\frac{p(\mathbf{B}; \mathbf{w}_0)}{q(\mathbf{B})}\right)\right], \quad (8)$$

which will lead to tractable computations for specific choices of \mathcal{Q} .

To make computations feasible in large data sets, we introduce the assumption that the \mathbf{b}_j ’s are independent in the approximate posterior:

$$q(\mathbf{B}) = \prod_{j=1}^p q_j(\mathbf{b}_j) \quad (9)$$

In the VI literature, this assumption is referred to as the “mean-field” approximation. In practice, this independence assumption has the effect of selecting at most one variable with high probability

when multiple variables are strongly correlated with each other [8]. This can be a problem if variable selection is the main goal [37], but it is less of a concern when phenotype prediction is the main goal, which is precisely our goal here. Under this independence assumption, we reexpress the optimization problem as

$$\hat{q}_1, \dots, \hat{q}_p := \underset{q_1, \dots, q_p}{\operatorname{argmax}} F(q_1, \dots, q_p; \mathbf{V}, \mathbf{w}_0). \quad (10)$$

Optimizing the ELBO over all q_1, \dots, q_p simultaneously is difficult. But optimizing a single factor q_j is relatively straightforward because it reduces to computing analytic posterior quantities for a Bayesian multivariate simple regression model. We formalize this statement in the following proposition.

PROPOSITION 1. Let $\bar{\mathbf{b}}_j$ be the expected value of \mathbf{b}_j with respect to the approximate posterior q_j , let $\bar{\mathbf{R}}_j = \mathbf{Y} - \sum_{j' \neq j} \mathbf{x}_{j'} \bar{\mathbf{b}}_{j'}^\top$ be the expected residuals ignoring the j th variable, and define a function

$$\text{BMSR-mix}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0) := (w_{1,1}, \dots, w_{1,K}, \mathbf{b}_{1,1}, \dots, \mathbf{b}_{1,K}, \mathbf{S}_{1,1}, \dots, \mathbf{S}_{1,K})$$

that returns the posterior distribution of \mathbf{b} under the Bayesian multivariate simple regression model with a mixture-of-normals prior (see Proposition 5 for the exact definition of BMSR-mix). Then we have that

$$\begin{aligned} \operatorname{argmax}_{q_j} F(q_1, \dots, q_j; \mathbf{V}, \mathbf{w}_0) &= \text{BMSR-mix}(\mathbf{x}_j, \bar{\mathbf{R}}_j, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0) \\ &:= (w_{1,1}^{(j)}, \dots, w_{1,K}^{(j)}, \mathbf{b}_{1,1}^{(j)}, \dots, \mathbf{b}_{1,K}^{(j)}, \mathbf{S}_{1,1}^{(j)}, \dots, \mathbf{S}_{1,K}^{(j)}). \end{aligned} \quad (11)$$

This suggests a *co-ordinate ascent algorithm* for maximizing the ELBO, in which we update the q_j 's sequentially. In the next section, we turn to the question of updating the other model parameters, \mathbf{w}_0 and \mathbf{V} .

D.2. Estimating \mathbf{V} and \mathbf{w}_0

We estimate \mathbf{w}_0 and \mathbf{V} by maximizing the ELBO over \mathbf{w}_0 and \mathbf{V} , with q fixed. This procedure can be viewed as an “EM-like” algorithm in which updating q is an approximate E-step, and updating \mathbf{w}_0 and \mathbf{V} is an M-step [30].

For the residual covariance matrix, \mathbf{V} , the update is

$$\text{UPDATE-RESID-COV}(\mathbf{X}, \bar{\mathbf{R}}, \mathbf{S}_1, \dots, \mathbf{S}_p) := \operatorname{argmax}_{\mathbf{V}} F(q; \mathbf{V}, \mathbf{w}_0) = \text{ERSS}/n, \quad (12)$$

in which “ERSS” is the expected residual sum of squares,

$$\text{ERSS} := E_q[\mathbf{R}^\top \mathbf{R}] = \bar{\mathbf{R}}^\top \bar{\mathbf{R}} + \sum_{j=1}^p \mathbf{x}_j^\top \mathbf{x}_j \mathbf{S}_j, \quad (13)$$

where $\mathbf{R} = \mathbf{Y} - \mathbf{X}\mathbf{B}$ is the $n \times r$ matrix of residuals, $\bar{\mathbf{R}} := E_q[\mathbf{R}] = \mathbf{Y} - \mathbf{X}\bar{\mathbf{B}}$, and $\mathbf{S}_j = \text{Cov}_q(\mathbf{b}_j)$. See Sec. G for a more detailed derivation of these expressions.

To derive the update for the mixture weights, \mathbf{w}_0 , we consider an augmented representation of the *mr.mash* model. Maximizing the ELBO for this augmented model, denoted as $\tilde{F}(q; \mathbf{V}, \mathbf{w}_0)$, with q fixed, yields the following update:

$$\text{UPDATE-WEIGHTS}(\mathbf{W}) := \operatorname{argmax}_{\mathbf{w}_0} \tilde{F}(q; \mathbf{V}, \mathbf{w}_0) = (\hat{w}_{0,1}, \dots, \hat{w}_{0,K}), \quad (14)$$

in which $\hat{w}_{0,k} := \sum_{j=1}^p w_{1,k}^{(j)}/p$, $w_{1,k}^{(j)}$ is defined in (11), and \mathbf{W} is a $p \times K$ matrix in which $w_{1,k}^{(j)}$ is stored in row j , column k of the matrix. See Sec. G for a more detailed derivation of this update.

E. Details of the methods compared

This section describes how the methods were run in the simulations and the GTEx case study. Note that we ran all methods standardizing \mathbf{X} and without standardizing \mathbf{Y} . The one exception was the *Elastic Net* implemented in the *glmnet* package; for *Elastic Net*, we standardized both \mathbf{X} and \mathbf{Y} .

E.1. Elastic Net

The *Elastic Net* method is a univariate multiple linear regression method that typically produces sparse estimates of the regression coefficients through a combination of ℓ_1 and ℓ_2 penalties [42]. Since the *Elastic Net* does not model correlations among effects of different phenotypes, it is only expected to be competitive with the multivariate regression methods when there is little benefit to modeling correlations, such as when genetic variant effects are rarely shared across phenotypes, or when the effects are already accurately estimated in the individual regressions.

We used the fast algorithms implemented in version 4.1.1 of the R package `glmnet` [14] to fit *Elastic Net* models separately to each of the r responses. The penalty strength parameter λ was chosen via cross-validation, separately for each of the r responses. The mixing parameter α , which controls the tradeoff between the Lasso penalty (ℓ_1 -norm) and the ridge penalty (ℓ_2 -norm), was set to 0.5, which was the same setting used in PrediXcan [15]. (Since the ground-truth effects were sparse, α mainly served to avoid degeneracies caused by strong correlations between SNPs.) We used the coefficient estimates at the value of λ minimizing the mean cross-validation error (`lambda.min`). We performed the cross-validation by calling `cv.glmnet` with `family = "gaussian"`, `alpha = 0.5`, and with all remaining arguments kept at their default values.

E.2. Group Lasso

The *Group Lasso* [39] is a penalized multivariate regression method, similar in some respects to the *Elastic Net*, except that it uses a “ ℓ_1/ℓ_2 ” penalty which gives greater preference to variables that have an effect in multiple phenotypes. As far as we know, the *Group Lasso* has not been used in the TWAS setting. However, we included *Group Lasso* in our comparisons because it has a very fast implementation in the R package `glmnet` [14], and for our application we expect it to perform well in settings where effects are shared across many phenotypes. On the other hand, since the ℓ_1/ℓ_2 penalty does not specifically account for sharing of effect sizes, nor can it model sharing of effects among subsets of phenotypes, we expect that the gains in accuracy will be weaker in some settings. Also note that current implementations of the *Group Lasso* cannot handle missing data so we did not include the *Group Lasso* in our simulations with missing data.

We used the *Group Lasso* implemented in version 4.1.1 of the `glmnet` R package. We called `cv.glmnet` with `family = "mgaussian"`, `alpha = 1`, and with all remaining arguments kept at their default values. We used the coefficient estimates at the value of λ minimizing the mean cross-validation error (`lambda.min`).

E.3. Sparse Multi-task Lasso

The *Sparse Multi-task Lasso* [25, 23] is a variation on the *Group Lasso* with a more flexible penalty. The *Sparse Multi-task Lasso* was specifically developed for eQTL studies with missing data [25, 23] and holds the promise of being able to adapt to a greater variety of data sets, and for these reasons it is a natural point of comparison to *mr.mash*. But in practice we found it was challenging to apply to the data sets considered in our simulations, and in particular in data sets without missing values it was much slower than the *Group Lasso* (implemented in the `glmnet` R package). Therefore, in practice the implementation of this method required compromises such as reducing the number of parameter settings in the cross-validation, and this may have reduced its performance.

The *Sparse Multi-task Lasso* minimizes the following penalized objective,

$$\ell^{\text{smt}}(\mathbf{B}) = \frac{1}{2} \sum_{s=1}^r \|\mathbf{y}_s - \mathbf{X}\mathbf{b}_s\|_2^2 + \lambda_1 \sum_{s=1}^r \|\mathbf{b}_s\|_1 + \lambda_2 \sum_{j=1}^p \|\mathbf{b}_j\|_2, \quad (15)$$

where \mathbf{y}_s denotes the s th column of \mathbf{Y} , \mathbf{b}_s denotes the s th column of \mathbf{B} , and \mathbf{b}_j denotes the j th row of \mathbf{B} . This formulation is equivalent to the objective function given in [23] up to a scaling of the penalty. This was previously studied as the “sparse multi-task lasso problem” [25]. The Lasso penalty on the *columns* of \mathbf{B} encourages fewer predictors to be selected [33], whereas the *Group Lasso* penalty on the *rows* of \mathbf{B} encourages the same predictors to be chosen for different responses [39].

To set the penalty parameters, we took a simple k -fold cross-validation approach. We used a uniformly spaced grid of candidate values on the log-scale for each of λ_1 and λ_2 , evaluated the mean-squared error for each fold and each combination of λ_1, λ_2 , and then selected the combination that had the smallest mean-squared error (averaged across all folds).

The implementation of the *Sparse Multi-task Lasso* made available in UTMOST [23] is difficult to use because of the nonstandard way it requires the inputs to be formatted. We also found that the memory usage could be very high when \mathbf{Y} had missing values. The main UTMOST Python interface is available from <https://github.com/Joker-Jerome/UTMOST> and the core code implementing the *Sparse Multi-task Lasso* is available from <https://github.com/yiminghu/CTIMP>. Later, another implementation of the *Sparse Multi-task Lasso* was developed in [26] (see https://github.com/RitchieLab/multi_tissue_twas_sim). This newer implementation, while easier to use, does not allow for missing values in \mathbf{Y} . Due to these considerations, we developed our own Python implementation of the *Sparse Multi-task Lasso* (available at <https://github.com/aksarkar/mtlasso>). Our Python implementation is slower than previous implementations (S6 Fig), in part owing to the core computations not being implemented in C++. On the other hand, our implementation is easier to use, allows for missing values, and, in simulated data sets, performs similarly to the `multi_tissue_twas_sim` implementation (S5 Fig). For additional details on these implementations (which differ in the initialization method and the grid of tuning parameters, among other things) and simulations, see [28] (see in particular the file `mrmash_vs_mtlasso_vs_utmash.html`) and <https://users.rcc.uchicago.edu/~aksarkar/gtex-pred> (see in particular “Sparse multi-task lasso”).

For \mathbf{Y} with missing data, one complication is that the correct updates for \mathbf{B} are expensive. Following the suggestion by [23], the `mtlasso` implements an approximate update for each column $s = 1, \dots, r$ of \mathbf{B} in which $(\mathbf{X}^{(s)})^\top \mathbf{X}^{(s)}$ is replaced with $\mathbf{X}^\top \mathbf{X}$, in which $\mathbf{X}^{(s)}$ denotes the matrix containing the rows of \mathbf{X} corresponding to non-missing entries of \mathbf{y}_s .

E.4. *mr.mash*

In the simulations with full data, we obtained an initial estimate of $\bar{\mathbf{B}}$ by running the *Group Lasso* on the same data, then setting \mathbf{B} to the *Group Lasso* coefficient estimates. We then obtained an initial estimate of \mathbf{V} as the sample covariance of the *Group Lasso* residuals. The prior mixture weight for the “spike”, $w_{0,0}$ (see eq. 16), was initialized to the proportion of *Group Lasso* coefficients that were zero, and the remaining prior mixture components were assigned equal weights. In the *mr.mash* model fitting, we estimated both \mathbf{w}_0 and \mathbf{V} , but forced \mathbf{V} to be diagonal. We ran the *mr.mash* model fitting algorithm with a convergence tolerance of 0.01.

For simulations with missing data, we ran *mr.mash* in the same way as in the simulations with full data, except for two differences. First, since the sample covariance of the residuals cannot be computed when there are missing values in \mathbf{Y} , we initialized \mathbf{V} using `flashr` [38], which can cope with missing values. Second, since the `glmnet` implementation of the *Group Lasso* does not allow missing values in \mathbf{Y} , we instead ran the *Elastic Net* separately on each column of \mathbf{Y} , and used the *Elastic Net* coefficient estimates to initialize $\bar{\mathbf{B}}$.

For the GTEx case study, we ran *mr.mash* in mostly the same way as in the simulations with missing data. Two differences were: (1) we allowed correlated residuals (*i.e.*, non-diagonal \mathbf{V}); and (2) to speed up the model fitting, at each outer-loop iteration beyond the initial 15 iterations we automatically dropped from the mixture prior any components with mixture weight smaller than 10^{-8} .

E.4.1. Prior covariance matrices

We defined the covariance matrices as combinations of different scaling factors $\omega_l \geq 0$ and normalized covariance matrices, $\mathbf{U}_{0,t}$. (By “normalized”, we mean that the largest diagonal element was always 1.) Therefore, the mixture prior was

$$\mathbf{b}_j \mid \mathbf{w}_0, \boldsymbol{\omega}, \mathcal{U}_0 \sim w_{0,0} \delta_0 + \sum_{l=1}^L \sum_{t=1}^T w_{0,l,t} N_r(\mathbf{0}, \omega_l^2 \mathbf{U}_{0,t}), \quad (16)$$

where δ_0 is the delta mass at zero (the “spike”). The scaling factors $\omega_1, \dots, \omega_L$ were chosen so that they were spaced uniformly on the log-scale, with a grid spacing of $\sqrt{2}$. The steps we took to determine the end points of the grid are described in [34].

In the next two sections we describe how the normalized covariance matrices $\mathbf{U}_{0,1}, \dots, \mathbf{U}_{0,T}$ were specified.

E.4.2. Canonical covariance matrices

When we ran *mr.mash* using “canonical” covariance matrices, the following matrices were included in the prior (16):

- The identity matrix, \mathbf{I}_r , modeling the case in which all effects are independent.
- The “equal effects” matrix, an $r \times r$ matrix of all ones, which models the case in which all effects are the same.
- Rank-1 matrices modeling tissue-specific effects. They are of the form $\mathbf{c}_s \mathbf{c}_s^\top$, in which \mathbf{c}_s is a vector of length r containing all zeros except for a 1 in the s th position. There were r of these matrices.
- Matrices with ones on the diagonal and σ on the off-diagonal, where σ is 0.25, 0.5 or 0.75. There were three of these matrices.

In total, $r + 5$ canonical matrices were included in the prior. So for the simulations the prior included 15 canonical matrices.

E.4.3. Data-driven matrices

Data-driven matrices are, as the term implies, adapted to the data, in contrast to the canonical matrices which are the same for every data set. The idea is that these data-driven covariance matrices should capture the principal patterns of effect sharing patterns across the r tissues. Estimating the data-driven covariance matrices involves combining information across many SNPs and many genes.

For the simulations, when data-driven matrices were included in the prior (16), we estimated the data-driven matrices following the analysis procedure largely as it was described in [34]. In brief, we prepared the summary statistics (least-squares effect estimates and corresponding standard errors) using training sets (real genotypes \mathbf{X} and simulated expression levels \mathbf{Y}) for all genes, then we used these summary statistics to estimate the data-driven covariance matrices using Extreme Deconvolution (ED) [7].

We used ED to estimate data-driven covariances because it is an EM algorithm for fitting mixture models of the form (2) in which the multivariate effect vectors \mathbf{b}_j are treated as unknown, and “noisy” estimates of the \mathbf{b}_j ’s are observed. In particular, ED estimates both the mixture weights $w_{0,k}$ and the covariance matrices $\mathbf{S}_{0,k}$ in (2). Once the ED algorithm converged to a solution, we extracted the covariance matrices estimated by ED to use as the data-driven covariances $\mathbf{U}_{0,t}$ in the prior.

To run the ED algorithm, one needs to provide initial estimates of the covariance matrices. We obtained rough initial estimates by running PCA and the flexible factor analysis methods implemented in the R package *flashr* [38], as well as one covariance estimate initialized to the empirical covariance matrix (that is, the maximum-likelihood estimate of the covariance). For the PCA-based initial estimates, we simply computed a singular value decomposition (implemented by R function `svd`) of a matrix formed by selected least-squares estimates of the effects \mathbf{b}_j , then obtained 3 rank-1 matrices from the top 3 PCs, and a rank-3 matrix from the linear combination of the top 3 PCs (so 4 PCA-based covariance matrices in total). After running *flashr*, we formed rank-1 covariance matrices from the estimated factors. Since *flashr* adaptively determines the number of factors—essentially, *flashr* will remove factors during the back-fitting stage if setting the factor to zero does not make the model fit worse—the number of covariances fit by ED was different for each data set. By automatically determining the number of factors, the idea is that the number of factors (and therefore the number of components in the mixture prior) should approximately reflect the complexity the effect sharing across tissues. Note that for the Equal Effects and Equal Effects + Null simulations, the final result

was always a mixture model with 9 mixture components (*i.e.*, 9 covariance matrices, 4 initialized with PCA, 4 initialized with `flashr`, and 1 initialized to the empirical covariance matrix).

The code for preprocessing the summary statistics is available from `prepare_sumstats_for_ED_prior.R` in [28], and the code implementing the remaining steps of this analysis is available at https://github.com/stephenslab/gtexresults/blob/master/workflows/mashr_flashr_workflow.ipynb (we used the “prior” workflow, version 2570aa6 from 2019-10-03). The one difference between the analysis of [34] and ours is that we replaced the sparse factor analysis (SFA) method [12] with `flashr` [38].

For the GTEx application, we made a couple small improvements to this pipeline. The key difference was the use of in-development model fitting algorithms that improve on the EM algorithms used in Extreme Deconvolution [7], and are implemented in the `udr` package available at <https://github.com/stephenslab/udr>. The script implementing this modified pipeline is available at https://github.com/cumc/bioworkflows/blob/master/multivariate-fine-mapping/mixture_prior.ipynb (we used the “extract_effects” workflow followed by the “ud” workflow in the script version with git commit id 5b7c688, with flags `--ud-method ted --ud-tol-lik 1e-3 --mixture-components flash flash_nonneg pca`). Since there were 5 training sets in the GTEx application, we ran this pipeline 5 times. In each training set, the pipeline produced a mixture model with 33–35 components.

F. Posterior computations for multivariate simple regression models

Here we derive elementary posterior expressions that will be used later on to develop the posterior computations for *mr.mash*.

F.1. Multivariate simple regression

We begin by deriving posterior computations for a multivariate regression with a single explanatory variable. Initially, we do not include an intercept.

The model is

$$\begin{aligned} \mathbf{Y} &= \mathbf{x}\mathbf{b}^\top + \mathbf{E} \\ \mathbf{E} &\sim MN_{n \times r}(\mathbf{0}, \mathbf{I}_n, \mathbf{V}), \end{aligned} \quad (17)$$

or more concisely

$$\mathbf{Y} \sim MN_{n \times r}(\mathbf{x}\mathbf{b}^\top, \mathbf{I}_n, \mathbf{V}), \quad (18)$$

where $\mathbf{Y} \in \mathbb{R}^{n \times r}$ is a matrix of observed responses for n samples across r conditions, \mathbf{x} is a vector of n observations for a single explanatory variable, $\mathbf{b} \in \mathbb{R}^r$ is the (unknown) vector of regression for the r conditions, $\mathbf{V} \in \mathbb{S}_{++}^r$ is an invertible covariance matrix, and $MN_{n \times r}(\mathbf{M}, \mathbf{U}, \mathbf{V})$ is the matrix normal distribution [10, 20] with mean \mathbf{M} and covariance matrices \mathbf{U} and \mathbf{V} (these are matrices of dimension $n \times r$, $n \times n$ and $r \times r$, respectively).

In the following, we give the expression for the matrix-normal likelihood and relate it to the more familiar multivariate normal distribution. The likelihood for the multivariate regression model (18) is

$$\begin{aligned} \ell(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) &:= p(\mathbf{Y} \mid \mathbf{x}, \mathbf{b}, \mathbf{V}) \\ &= MN_{n \times r}(\mathbf{Y}; \mathbf{x}\mathbf{b}^\top, \mathbf{I}_n, \mathbf{V}) \\ &= |2\pi\mathbf{V}|^{-n/2} \exp\left\{-\frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)]\right\}. \end{aligned} \quad (19)$$

Given \mathbf{V} , the least-squares estimate of \mathbf{b} , denoted $\hat{\mathbf{b}}$, which is also the value of \mathbf{b} maximizing the likelihood (19), and its covariance, $\hat{\mathbf{S}}$, are

$$\hat{\mathbf{b}} := \hat{\mathbf{b}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}) = \frac{\mathbf{Y}^\top \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \quad (20)$$

$$\hat{\mathbf{S}} := \hat{\mathbf{S}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}) = \frac{\mathbf{V}}{\mathbf{x}^\top \mathbf{x}}. \quad (21)$$

Using these quantities, the likelihood (19) can be rewritten as

$$\ell(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) = |2\pi\mathbf{V}|^{-n/2} \exp\left\{-\frac{1}{2}[\text{tr}(\mathbf{Y}\mathbf{V}^{-1}\mathbf{Y}^\top) + (\mathbf{b} - \hat{\mathbf{b}})^\top \hat{\mathbf{S}}^{-1}(\mathbf{b} - \hat{\mathbf{b}}) - \hat{\mathbf{b}}^\top \hat{\mathbf{S}}^{-1} \hat{\mathbf{b}}]\right\}. \quad (22)$$

This expression is proportional to a multivariate normal density on \mathbf{b} , and in particular we have that

$$\ell(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) \propto N_r(\mathbf{b}; \hat{\mathbf{b}}, \hat{\mathbf{S}}), \quad (23)$$

where $N_n(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal density at $\boldsymbol{\theta} \in \mathbb{R}^n$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

F.2. Bayesian multivariate simple regression with a normal prior

In the following proposition, we apply the above results for the multivariate simple regression to a Bayesian multivariate simple regression model with a normal prior.

PROPOSITION 2 (BAYESIAN MULTIVARIATE SIMPLE REGRESSION WITH A NORMAL PRIOR). *Consider the multivariate simple regression model (18) with a multivariate normal prior on the regression coefficients,*

$$\mathbf{b} \mid \mathbf{S}_0 \sim N_r(\mathbf{0}, \mathbf{S}_0), \quad (24)$$

where $\mathbf{S}_0 \in \mathbb{S}_+^r$ is a (possibly singular) covariance matrix. The posterior of \mathbf{b} is

$$\mathbf{b} \mid \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_0 \sim N_r(\mathbf{b}_1, \mathbf{S}_1), \quad (25)$$

where

$$\mathbf{b}_1 := \mathbf{b}_1(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_0) = \mathbf{S}_1 \hat{\mathbf{S}}^{-1} \hat{\mathbf{b}} \quad (26)$$

$$\mathbf{S}_1 := \mathbf{S}_1(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_0) = (\mathbf{S}_0^{-1} + \hat{\mathbf{S}}^{-1})^{-1}. \quad (27)$$

The Bayes Factor (BF) comparing this model against the null model ($\mathbf{b} = \mathbf{0}$), which is defined as the ratio of the two likelihoods,

$$\begin{aligned} \text{BF}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_0) &= \frac{p(\mathbf{Y} \mid \mathbf{x}, \mathbf{V}, \mathbf{S}_0)}{p(\mathbf{Y} \mid \mathbf{x}, \mathbf{b} = \mathbf{0}, \mathbf{V})} \\ &= \frac{\int \ell(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) p(\mathbf{b} \mid \mathbf{S}_0) d\mathbf{b}}{p(\mathbf{Y} \mid \mathbf{x}, \mathbf{b} = \mathbf{0}, \mathbf{V})}, \end{aligned} \quad (28)$$

works out to

$$\text{BF}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_0) = \frac{|\hat{\mathbf{S}}|^{1/2}}{|\mathbf{S}_0 + \hat{\mathbf{S}}|^{1/2}} \exp\left\{\frac{1}{2} \hat{\mathbf{b}}^\top \hat{\mathbf{S}}^{-1} \mathbf{S}_1 \hat{\mathbf{S}}^{-1} \hat{\mathbf{b}}\right\} \quad (29)$$

$$= \frac{|\hat{\mathbf{S}}|^{1/2}}{|\mathbf{S}_0 + \hat{\mathbf{S}}|^{1/2}} \exp\left\{\frac{1}{2} \mathbf{b}_1^\top \mathbf{S}_1^{-1} \mathbf{b}_1\right\}. \quad (30)$$

The same BF can also be equivalently expressed as a ratio of two multivariate normal densities,

$$\text{BF}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_0) = \frac{N_r(\hat{\mathbf{b}}; \mathbf{0}, \mathbf{S}_0 + \hat{\mathbf{S}})}{N_r(\hat{\mathbf{b}}; \mathbf{0}, \hat{\mathbf{S}})}. \quad (31)$$

F.3. Multivariate simple regression with an intercept

Now we extend the multivariate simple regression model (18) to include an intercept. We show that including the intercept in the model is equivalent to “centering” \mathbf{x} and the columns of \mathbf{Y} so that they all have means of zero. This equivalence can be understood from two different perspectives: from a point-estimation perspective, centering \mathbf{x} and the columns of \mathbf{Y} is equivalent to computing a maximum-likelihood estimate of the intercept; from a Bayesian averaging point-of-view, it is equivalent to integrating out the intercept with respect to an (improper) uniform prior (see also George and McCulloch 17). These results are summarized in Proposition 3.

The multivariate simple regression model with an intercept is

$$\mathbf{Y} \sim MN_{n \times r}(\mathbf{1}\boldsymbol{\mu}^\top + \mathbf{x}\mathbf{b}^\top, \mathbf{I}_n, \mathbf{V}), \quad (32)$$

in which $\mathbf{1}$ is a vector of ones of length n , and $\boldsymbol{\mu} \in \mathbb{R}^r$ is the (unknown) intercept. The likelihood of $\boldsymbol{\mu}, \mathbf{b}$ under this model is

$$\begin{aligned} \ell(\boldsymbol{\mu}, \mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) &:= p(\mathbf{Y} \mid \mathbf{x}, \boldsymbol{\mu}, \mathbf{b}, \mathbf{V}) \\ &= |2\pi\mathbf{V}|^{-n/2} \exp\left\{-\frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{1}\boldsymbol{\mu}^\top - \mathbf{x}\mathbf{b}^\top)^\top(\mathbf{Y} - \mathbf{1}\boldsymbol{\mu}^\top - \mathbf{x}\mathbf{b}^\top)]\right\}. \end{aligned} \quad (33)$$

PROPOSITION 3 (MULTIVARIATE SIMPLE REGRESSION WITH AN INTERCEPT). *Consider the multivariate simple regression (32). The least-squares estimate of $\boldsymbol{\mu}$ —which is also the value of $\boldsymbol{\mu}$ maximizing the likelihood $\ell(\boldsymbol{\mu}, \mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V})$ —and its covariance matrix are*

$$\hat{\boldsymbol{\mu}} := \hat{\boldsymbol{\mu}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{b}) = \bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b} \quad (34)$$

$$\hat{\mathbf{S}}_\mu := \hat{\mathbf{S}}_\mu(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{b}) = \frac{1}{n}\mathbf{V}, \quad (35)$$

in which $\bar{\mathbf{x}} = \frac{1}{n}\mathbf{x}^\top\mathbf{1} = \frac{1}{n}\sum_{i=1}^n x_i$ is the sample mean of \mathbf{x} , and $\bar{\mathbf{y}} = \frac{1}{n}\mathbf{Y}^\top\mathbf{1}$ is the vector containing the column means of \mathbf{Y} .

The profile likelihood for \mathbf{b} is

$$\begin{aligned} \ell^*(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) &:= \max_{\boldsymbol{\mu}} \ell(\boldsymbol{\mu}, \mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) \\ &= \ell(\mathbf{b}; \tilde{\mathbf{x}}, \tilde{\mathbf{Y}}, \mathbf{V}), \end{aligned} \quad (36)$$

in which $\tilde{\mathbf{x}} := \mathbf{x} - \bar{\mathbf{x}}\mathbf{1}$ and $\tilde{\mathbf{Y}} := \mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top$ are the centered \mathbf{x} and \mathbf{Y} . In other words, the profile likelihood for the multivariate simple regression with an intercept is the same as the likelihood for the multivariate simple regression without an intercept if we first center \mathbf{x} and \mathbf{Y} . Centering \mathbf{x} and \mathbf{Y} is therefore equivalent to including an intercept in the multivariate regression and estimating the intercept by maximum-likelihood.

Next, consider Bayesian calculations for $\boldsymbol{\mu}$ with a multivariate normal prior, $\boldsymbol{\mu} \mid \mathbf{S}_{0\mu} \sim N_r(0, \mathbf{S}_{0\mu})$, in which $\mathbf{S}_{0\mu} \in \mathbb{S}_+^r$ is a (possibly singular) covariance matrix. The posterior for $\boldsymbol{\mu}$ conditioned on \mathbf{b} is

$$\boldsymbol{\mu} \mid \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}, \mathbf{b} \sim N_r(\boldsymbol{\mu}_1, \mathbf{S}_{1\mu}), \quad (37)$$

where

$$\boldsymbol{\mu}_1 := \boldsymbol{\mu}_1(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}, \mathbf{b}) = \mathbf{S}_{1\mu}\hat{\mathbf{S}}_\mu^{-1}\hat{\boldsymbol{\mu}} \quad (38)$$

$$\mathbf{S}_{1\mu} := \mathbf{S}_{1\mu}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}, \mathbf{b}) = (\mathbf{S}_{0\mu}^{-1} + \hat{\mathbf{S}}_\mu^{-1})^{-1}. \quad (39)$$

The marginal likelihood obtained by averaging over the intercept is

$$\begin{aligned} \ell^*(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}) &:= \int \ell(\boldsymbol{\mu}, \mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) p(\boldsymbol{\mu} \mid \mathbf{S}_{0\mu}) d\boldsymbol{\mu} \\ &= |2\pi\mathbf{V}|^{-n/2} |\mathbf{S}_{0\mu}^{-1}\mathbf{S}_{1\mu}|^{1/2} \exp\left\{\frac{1}{2}\boldsymbol{\mu}_1^\top \mathbf{S}_{1\mu}^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)]\right\}. \end{aligned} \quad (40)$$

In the special case of an (improper) uniform prior on $\boldsymbol{\mu}$, defined as $\boldsymbol{\mu} \sim N_r(0, \mathbf{S}_{0\mu})$ with $\mathbf{S}_{0\mu}^{-1} \rightarrow 0$, the posterior mean reduces to the least-squares estimate $\boldsymbol{\mu}_1 = \hat{\boldsymbol{\mu}}$, and $\mathbf{S}_{1\mu} = \hat{\mathbf{S}}_\mu$, and the marginal likelihood simplifies to

$$\begin{aligned} \ell^*(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}) &= |2\pi\mathbf{V}|^{-n/2} |\mathbf{S}_{0\mu}^{-1}\hat{\mathbf{S}}_\mu|^{1/2} \exp\left\{\frac{1}{2}\hat{\boldsymbol{\mu}}^\top \hat{\mathbf{S}}_\mu^{-1}\hat{\boldsymbol{\mu}} - \frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)]\right\} \\ &= |\mathbf{S}_{0\mu}^{-1}\hat{\mathbf{S}}_\mu|^{1/2} \times \ell(\mathbf{b}; \tilde{\mathbf{x}}, \tilde{\mathbf{Y}}, \mathbf{V}). \end{aligned} \quad (41)$$

In other words, the marginal likelihood for multivariate regression with an intercept when we use an improper uniform prior for the intercept is the same (up to a constant of proportionality) as the likelihood for multivariate regression without an intercept when we first center \mathbf{x} and \mathbf{Y} .

A proof of Proposition 3 is given in Sec. J.

F.4. Bayesian multivariate simple regression with a mixture prior

Here we extend the Bayesian multivariate simple regression with a normal prior to the multivariate simple regression with a mixture-of-normals prior,

$$\mathbf{b} \mid \mathcal{S}_0, \mathbf{w}_0 \sim \sum_{k=1}^K w_{0,k} N_r(\mathbf{0}, \mathbf{S}_{0,k}), \quad (42)$$

in which $\mathcal{S}_0 := \{\mathbf{S}_{0,1}, \dots, \mathbf{S}_{0,K}\}$, each $\mathbf{S}_{0,k} \in \mathbb{S}_+^r$ is a (possibly singular) covariance matrix, for $k = 1, \dots, K$, and the mixture weights $\mathbf{w}_0 = (w_{0,1}, \dots, w_{0,K})$ are non-negative and sum to 1. The normal prior is a special case of (42) when $K = 1$.

To facilitate derivation of the posterior computations, we introduce the following data augmentation that recovers (42) after averaging, or integrating, over the latent variable γ :

$$\begin{aligned} p(\gamma = k \mid \mathbf{w}_0) &= w_{0,k} \\ \mathbf{b} \mid \mathcal{S}_0, \gamma = k &\sim N_r(\mathbf{0}, \mathbf{S}_{0,k}). \end{aligned} \quad (43)$$

This augmented model allows us to conveniently reuse the posterior computations for the simpler models; in particular, posterior computations conditioned on γ reduce to computations for the Bayesian multivariate regression model with a normal prior, as we show in the proposition below.

PROPOSITION 4. *Given \mathbf{w}_0 and \mathcal{S}_0 , the Bayes factor comparing this model against the null model ($\mathbf{b} = \mathbf{0}$) is*

$$\begin{aligned} \text{BF}^{\text{mix}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0) &= \frac{p(\mathbf{Y} \mid \mathbf{x}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0)}{p(\mathbf{Y} \mid \mathbf{x}, \mathbf{b} = \mathbf{0}, \mathbf{V})} \\ &= \frac{\sum_{k=1}^K w_{0,k} N_r(\hat{\mathbf{b}}; \mathbf{0}, \mathbf{S}_{0,k} + \hat{\mathbf{S}})}{N_r(\hat{\mathbf{b}}; \mathbf{0}, \hat{\mathbf{S}})}. \end{aligned} \quad (44)$$

The posterior mean and covariance of \mathbf{b} are

$$\mathbf{b}_1^{\text{mix}} := \mathbf{b}_1^{\text{mix}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0) = \sum_{k=1}^K w_{1,k} \mathbf{b}_{1,k} \quad (45)$$

$$\mathbf{S}_1^{\text{mix}} := \mathbf{S}_1^{\text{mix}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0) = \sum_{k=1}^K w_{1,k} (\mathbf{b}_{1,k} \mathbf{b}_{1,k}^\top + \mathbf{S}_{1,k}) - \mathbf{b}_1^{\text{mix}} (\mathbf{b}_1^{\text{mix}})^\top, \quad (46)$$

in which the posterior assignment probabilities $w_{1,k} := p(\gamma = k \mid \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0)$ are given by

$$w_{1,k} = \frac{w_{0,k} \text{BF}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0,k})}{\sum_{k'=1}^K w_{0,k'} \text{BF}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0,k'})}. \quad (47)$$

Here we use $\mathbf{b}_{1,k}$ and $\mathbf{S}_{1,k}$ to denote the posterior mean and covariance of \mathbf{b} given $\gamma = k$; they are given by (26) and (27), respectively, replacing \mathbf{S}_0 with $\mathbf{S}_{0,k}$.

G. Derivation of *mr.mash* variational algorithms with fully observed \mathbf{Y}

To develop the *mr.mash* algorithm, we formulate an optimization problem, then we design an iterative procedure for solving the optimization problem. Here we assume that \mathbf{Y} is fully observed; in Sec. H we extend these methods to allow for missing values in \mathbf{Y} . For convenience, we restate the *mr.mash* model for fully observed \mathbf{Y} :

$$\begin{aligned} \mathbf{Y} &\sim MN_{n \times r}(\mathbf{X}\mathbf{B}, \mathbf{I}_n, \mathbf{V}) \\ \mathbf{b}_j &\stackrel{i.i.d.}{\sim} g, \text{ for } j = 1, \dots, p, \end{aligned} \quad (48)$$

where \mathbf{Y} is an $n \times r$ matrix containing n observations of r outcomes, \mathbf{X} is an $n \times p$ matrix containing n observations of p explanatory variables, and $\mathbf{E} \in \mathbb{R}^{n \times r}$ is a matrix of residuals. The unknowns are the regression coefficients $\mathbf{B} \in \mathbb{R}^{p \times r}$, the residual covariance matrix $\mathbf{V} \in \mathbb{S}_+^r$, and the prior on the regression coefficients, g . We denote the regression coefficients for the j th explanatory variable by $\mathbf{b}_j = (b_{1j}, \dots, b_{rj})^\top$.

We develop an empirical Bayes method for *mr.mash* in which the residual covariance \mathbf{V} and prior g are estimated, and a posterior distribution of \mathbf{B} is computed conditioned on the estimates of \mathbf{V} and g . This is formalized as the following optimization problem:

$$\begin{aligned} & \text{maximize} && F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) \\ & \text{subject to} && g \in \mathcal{G}, q \in \mathcal{Q}, \mathbf{V} \in \mathbb{S}_+^r, \end{aligned} \quad (49)$$

where \mathcal{G} is a prespecified set of prior distributions on \mathbf{b} , \mathcal{Q} is a prespecified set of posterior distributions for \mathbf{B} (possibly the set of *all* valid posterior distributions on \mathbb{R}^r), and $F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y})$ is the “evidence lower bound” (ELBO), a lower bound to the marginal likelihood [3]:

$$\begin{aligned} \log p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, g) &= \log \int p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, \mathbf{B}) \prod_{j=1}^p g(\mathbf{b}_j) d\mathbf{B} \\ &\geq F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) \\ &= E_q[\log p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, \mathbf{B})] - \sum_{j=1}^p D_{\text{KL}}(q_j(\mathbf{b}_j) \parallel g(\mathbf{b}_j)), \end{aligned} \quad (50)$$

in which $D_{\text{KL}}(f(\mathbf{x}) \parallel g(\mathbf{x})) := \int f(\mathbf{x}) \log\{f(\mathbf{x})/g(\mathbf{x})\} d\mathbf{x}$ is the Kullback-Leibler (K-L) divergence measure between probability distributions $f(\mathbf{x})$ and $g(\mathbf{x})$ on $\mathbf{x} \in \mathbb{R}^d$, and $q_j(\mathbf{b}_j)$ is the marginal posterior probability for \mathbf{b}_j . When \mathcal{Q} is the set of all valid posterior distributions on $\mathbb{R}^{p \times r}$, the variational distribution $q \in \mathcal{Q}$ maximizing the ELBO (50), $\hat{q} := \arg\max_{q \in \mathcal{Q}} F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y})$, is the true posterior, $\hat{q}(\mathbf{B}) \propto p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, \mathbf{B}) g(\mathbf{B})$, and the ELBO at \hat{q} , $F(\hat{q}, g, \mathbf{V}; \mathbf{X}, \mathbf{Y})$, is equal to the marginal likelihood $\log p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, g)$ [24, 30, 35].

G.1. Estimating \mathbf{V}

To estimate \mathbf{V} , we focus on the parts of the ELBO that involve \mathbf{V} :

$$F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) = E_q[\log p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, \mathbf{B})] + \sum_{j=1}^p E_q[\log g(\mathbf{b}_j)] + \text{const}, \quad (51)$$

where “const” is a placeholder for additional terms that do not involve g or \mathbf{V} . This is the familiar expected complete log-likelihood that appears in derivations of EM algorithms [30], the one difference being that the expectations are computed with respect to $q(\mathbf{B})$ instead of the exact posterior, $p(\mathbf{B} \mid \mathbf{X}, \mathbf{Y}, \mathbf{V}, g) \propto p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, \mathbf{B}) g(\mathbf{B})$.

Expanding (51), we have

$$F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) = -\frac{n}{2} \log |2\pi\mathbf{V}| - \frac{1}{2} \text{tr}(\mathbf{V}^{-1} \text{ERSS}) + \text{const}, \quad (52)$$

where $\text{ERSS} \in \mathbb{R}^{r \times r}$ is the expected residual sum of squares, and the “const” in this expression includes additional terms in the ELBO that do not involve \mathbf{V} . Therefore, the setting of the residual covariance that maximizes the ELBO (with all other quantities fixed) is

$$\hat{\mathbf{V}} := \arg\max_{\mathbf{V}} F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) = \text{ERSS}/n. \quad (53)$$

The expected residual sum of squares (ERSS) is

$$\text{ERSS} := E_q[\mathbf{R}^\top \mathbf{R}], \quad (54)$$

where $\mathbf{R} = \mathbf{Y} - \mathbf{X}\mathbf{B}$ is the $n \times r$ matrix of residuals, and expectations are taken with respect to $q(\mathbf{B})$. After expanding and rearranging terms, the ERSS works out to

$$\text{ERSS} = \bar{\mathbf{R}}^\top \bar{\mathbf{R}} + \sum_{j=1}^p \sum_{j'=1}^p \mathbf{x}_j^\top \mathbf{x}_{j'} \text{Cov}_q(\mathbf{b}_j, \mathbf{b}_{j'}) \quad (55)$$

where $\bar{\mathbf{R}} := E_q[\mathbf{Y} - \mathbf{X}\bar{\mathbf{B}}] = \mathbf{Y} - \mathbf{X}\bar{\mathbf{B}}$ is the posterior mean residual matrix with respect to $q(\mathbf{B})$, $\bar{\mathbf{B}} := E_q[\mathbf{B}]$ is the posterior mean of the coefficients \mathbf{B} with respect to $q(\mathbf{B})$, $\mathbf{x}_j = (x_{1j}, \dots, x_{nj})^\top$ denotes the j th column of \mathbf{X} , and $\text{Cov}_q(\mathbf{b}_j, \mathbf{b}_{j'}) \in \mathbb{R}^{r \times r}$ is the covariance between \mathbf{b}_j and $\mathbf{b}_{j'}$, again taken with respect to $q(\mathbf{B})$. Notice that these expressions do not make any special assumptions about the form of $q(\mathbf{B})$, and so they are valid for any $q(\mathbf{B})$.

The solution (53) can be viewed as an M-step update in an EM algorithm for maximum-likelihood estimation of \mathbf{V} in which expectations (computed in the E-step) are computed approximately [2, 3].

G.2. Computing approximate posteriors for \mathbf{B}

In this section, we derive posterior computations for \mathbf{B} with the assumption that the prior, g , is a mixture of multivariate normals (42), and with the assumption that q factorizes over the variables j . This is summarized by the following proposition.

PROPOSITION 5. *Suppose that the prior on the regression coefficients, g , is a mixture of multivariate normals (42). Further, we restrict \mathcal{Q} to be the set of posterior distributions on \mathbf{B} that factorize over the variables j , or equivalently over the columns of \mathbf{B} ,*

$$q(\mathbf{B}) = \prod_{j=1}^p q_j(\mathbf{b}_j). \quad (56)$$

Defined in this way, each factor $q_j(\mathbf{b}_j)$ is also a marginal posterior. Define a function, BMSR-mix, that returns the posterior distribution of \mathbf{b} under the Bayesian multivariate regression model with a mixture-of-normals prior (see Sec. F.4). Since the posterior distribution is uniquely determined by the posterior assignment probabilities $w_{1,k}$, the posterior means $\mathbf{b}_{1,k}$ and the covariances $\mathbf{S}_{1,k}$, we write this as

$$\text{BMSR-mix}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0) := (w_{1,1}, \dots, w_{1,K}, \mathbf{b}_{1,1}, \dots, \mathbf{b}_{1,K}, \mathbf{S}_{1,1}, \dots, \mathbf{S}_{1,K}). \quad (57)$$

Then the update for $q_j(\mathbf{b}_j)$ —that is, the $q_j(\mathbf{b}_j)$ that maximizes the ELBO (50) while g, \mathbf{V} and the other factors $q_{j'}(\mathbf{b}_{j'}), j' \neq j$ are fixed—has a closed-form solution:

$$\begin{aligned} \hat{q}_j &:= \operatorname{argmax}_{q_j} F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) \\ &= \text{BMSR-mix}(\mathbf{x}_j, \bar{\mathbf{R}}_j, \mathbf{V}; \mathcal{S}_0, \mathbf{w}_0), \end{aligned} \quad (58)$$

in which we define $\bar{\mathbf{R}}_j := E_q[\mathbf{Y} - \sum_{j' \neq j} \mathbf{x}_{j'} \mathbf{b}_{j'}^\top] = \mathbf{Y} - \sum_{j' \neq j} \mathbf{x}_{j'} \bar{\mathbf{b}}_{j'}^\top$ as the $n \times r$ matrix of residuals that ignore the j th variable.

COROLLARY 1. *A direct corollary of Proposition 5 is that the optimal $q(\mathbf{B})$ restricted according to (56) is a product of factors in which the individual factors are mixtures of multivariate normals, which we write as*

$$q_j(\mathbf{b}_j) = \sum_{k=1}^K w_{1,k}^{(j)} N_r(\mathbf{b}_j; \mathbf{b}_{1,k}^{(j)}, \mathbf{S}_{1,k}^{(j)}). \quad (59)$$

In summary, we have imposed a conditional independence assumption (56) on the posterior of \mathbf{B} , namely that \mathbf{b}_j is independent of the other coefficients. Imposing this conditional independence assumption naturally leads to a coordinatewise updating procedure in which $q_j(\mathbf{b}_j)$ is optimized while the remaining coordinates $q_{j'}(\mathbf{b}_{j'}), j' \neq j$, are fixed, and each of the coordinatewise updates has a simple, closed-form solution (58). A corollary is that the optimal $q_j(\mathbf{b}_j)$ over all possible posterior distributions on \mathbf{b}_j is a mixture of multivariate normals.

The proof of Proposition 5 is sketched in the next two subsections.

G.2.1. Special case of $p = 1$

We have already established that posterior computations are tractable for the special case of one variable when g is a mixture-of-normals prior; this is Bayesian multivariate simple regression with a mixture-of-normals prior, and the posterior quantities were derived in Sec. F.4. In particular, for

$$g(\mathbf{b}) = \sum_{k=1}^K w_{0,k} N_r(\mathbf{0}, \mathbf{S}_{0,k}), \quad (60)$$

the posterior distribution of \mathbf{b} is

$$p(\mathbf{b} \mid \mathbf{x}, \mathbf{Y}, \mathbf{V}, g) = \sum_{k=1}^K w_{1,k} N_r(\mathbf{b}_{1,k}, \mathbf{S}_{1,k}). \quad (61)$$

The ELBO for the *mr.mash* model with $p = 1$ variable (dropping the unneeded j subscripts) is

$$\begin{aligned} F^{(p=1)}(q, g, \mathbf{V}; \mathbf{x}, \mathbf{Y}) &= E_q[\log p(\mathbf{Y} \mid \mathbf{x}, \mathbf{V}, \mathbf{b})] - D_{\text{KL}}(q(\mathbf{b}) \parallel g(\mathbf{b})) \\ &= -\frac{n}{2} \log |2\pi \mathbf{V}| - \frac{1}{2} \text{tr}(\mathbf{V}^{-1} \text{ERSS}) - D_{\text{KL}}(q(\mathbf{b}) \parallel g(\mathbf{b})), \end{aligned} \quad (62)$$

where

$$\begin{aligned} \text{ERSS} &= E_q[(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top (\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)] \\ &= (\mathbf{Y} - \mathbf{x}\bar{\mathbf{b}}^\top)^\top (\mathbf{Y} - \mathbf{x}\bar{\mathbf{b}}^\top) + \mathbf{x}^\top \mathbf{x} \text{Var}_q[\mathbf{b}], \end{aligned} \quad (63)$$

and where $\bar{\mathbf{b}} := E_q[\mathbf{b}] = \mathbf{b}_1^{\text{mix}}$ is the posterior mean of the coefficients \mathbf{b} with respect to the distribution $q(\mathbf{b})$ [see eq. 45], and $\text{Var}_q[\mathbf{b}] = \mathbf{S}_1^{\text{mix}}$ is the covariance of \mathbf{b} , again with respect to $q(\mathbf{b})$ [see eq. 46].

When \mathcal{Q} is the set of all valid posterior distributions on \mathbb{R}^r , the variational distribution $q \in \mathcal{Q}$ maximizing the ELBO (62), $\hat{q} := \arg\max_q F^{(p=1)}(q, g, \mathbf{V}; \mathbf{x}, \mathbf{Y})$, is the true posterior, $\hat{q}(\mathbf{b}) = p(\mathbf{b} \mid \mathbf{x}, \mathbf{Y}, \mathbf{V}, g)$. At \hat{q} , the ELBO (62) is equal to the marginal log-likelihood; that is, $F^{(p=1)}(\hat{q}, g, \mathbf{V}; \mathbf{x}, \mathbf{Y}) = \log p(\mathbf{Y} \mid \mathbf{x}, \mathbf{V}, g)$.

G.2.2. Updating $q_j(\mathbf{b}_j)$

To allow for posterior computations that are tractable, we restricted \mathcal{Q} to the set of posterior distributions that factorize over the variables j ; see eq. 56. With this approximation, we can now divide and conquer; we consider the problem of finding a $q_j(\mathbf{b}_j)$ that maximizes the ELBO (50) while the remaining factors are fixed.

Expanding only the terms involving q_j , the ELBO reduces to

$$\begin{aligned} F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) &= -\frac{n}{2} \log |2\pi \mathbf{V}| - \frac{1}{2} \text{tr}(\mathbf{V}^{-1} \text{ERSS}) - D_{\text{KL}}(q_j(\mathbf{b}_j) \parallel g(\mathbf{b}_j)) + \text{const} \\ &= -\frac{n}{2} \log |2\pi \mathbf{V}| - \frac{1}{2} \text{tr}[\mathbf{V}^{-1} (\mathbf{Y} - \sum_{j=1}^p \mathbf{x}_j \bar{\mathbf{b}}_j^\top)^\top (\mathbf{Y} - \sum_{j=1}^p \mathbf{x}_j \bar{\mathbf{b}}_j^\top)] \\ &\quad - \frac{1}{2} \mathbf{x}_j^\top \mathbf{x}_j \text{tr}(\mathbf{V}^{-1} \text{Var}_q[\mathbf{b}_j]) - D_{\text{KL}}(q_j(\mathbf{b}_j) \parallel g(\mathbf{b}_j)) + \text{const} \\ &= F^{(p=1)}(q_j, g, \mathbf{V}; \mathbf{x}_j, \bar{\mathbf{R}}_j) + \text{const}. \end{aligned} \quad (64)$$

where the “const” in each expression includes additional terms not involving q_j . The covariance terms $\text{Cov}_q(\mathbf{b}_j, \mathbf{b}_{j'})$ for all $j \neq j'$ disappear from the ERSS because they are zero by assumption (56), and note that $\text{Cov}_q(\mathbf{b}_j, \mathbf{b}_j) = \text{Var}_q(\mathbf{b}_j)$. In summary, the ELBO can be rearranged to exactly match the expression for the single-variable ELBO (62) if we ignore terms not involving q_j .

G.3. Computing the ELBO

While computing the *mr.mash* ELBO (50) is not required, it is useful for monitoring progress of the model fitting, and for model comparison (e.g., comparing different *mr.mash* model fits). Here we explain how we compute the ELBO.

Expanding terms in (50), we have

$$F(q, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) = -\frac{n}{2} \log |2\pi \mathbf{V}| - \frac{1}{2} \text{tr}(\mathbf{V}^{-1} \text{ERSS}) - \sum_{j=1}^p D_{\text{KL}}(q_j(\mathbf{b}_j) \parallel g(\mathbf{b}_j)), \quad (65)$$

where the ERSS, starting from (55), simplifies to

$$\text{ERSS} = \bar{\mathbf{R}}^\top \bar{\mathbf{R}} + \sum_{j=1}^p \mathbf{x}_j^\top \mathbf{x}_j \text{Var}_q[\mathbf{b}_j]. \quad (66)$$

Recall, $\text{Cov}_q(\mathbf{b}_j, \mathbf{b}_{j'}) = 0$ for all $j \neq j'$ when assuming the factorization (56). The means $\bar{\mathbf{b}}_j$ and variances $\text{Var}_q[\mathbf{b}_j]$ appearing in the ERSS are easily computed; see Sec. G.2.

The remaining terms in the ELBO (65) to work out are the KL-divergences. We show next that it is most convenient to compute each KL-divergence $D_{\text{KL}}(q_j(\mathbf{b}_j) \parallel g(\mathbf{b}_j))$ when $q(\mathbf{b}_j)$ is updated.

G.3.1. Kullback-Leibler divergence for special case of $p = 1$

Starting from (62), the KL-divergence for the *mr.mash* model with one variable is

$$\begin{aligned} D_{\text{KL}}(q(\mathbf{b}) \parallel g(\mathbf{b})) &= E_q[\log p(\mathbf{Y} \mid \mathbf{x}, \mathbf{V}, \mathbf{b})] - F^{(p=1)}(q, g, \mathbf{V}; \mathbf{x}, \mathbf{Y}) \\ &= -\frac{n}{2} \log |2\pi \mathbf{V}| - \frac{1}{2} \text{tr}(\mathbf{V}^{-1} \text{ERSS}) - F^{(p=1)}(q, g, \mathbf{V}; \mathbf{x}, \mathbf{Y}), \end{aligned} \quad (67)$$

in which the ERSS for the single-variant *mr.mash* model is given in (63). Recall, the optimal variational distribution is equal to the true posterior, $\hat{q}(\mathbf{b}) = p(\mathbf{b} \mid \mathbf{x}, \mathbf{Y}, \mathbf{V}, g)$, and when the optimal variational distribution is obtained, the ELBO is equal to the marginal log-likelihood. Therefore, we have

$$\begin{aligned} D_{\text{KL}}(\hat{q}(\mathbf{b}) \parallel g(\mathbf{b})) &= -\frac{n}{2} \log |2\pi \mathbf{V}| - \frac{1}{2} \text{tr}(\mathbf{V}^{-1} \text{ERSS}) - \log p(\mathbf{Y} \mid \mathbf{x}, \mathbf{b}, \mathbf{V}, g) \\ &= -\frac{n}{2} \log |2\pi \mathbf{V}| - \frac{1}{2} \text{tr}(\mathbf{V}^{-1} \text{ERSS}) \\ &\quad - \log \text{BF}^{\text{mix}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0) - \log p(\mathbf{Y} \mid \mathbf{x}, \mathbf{b} = \mathbf{0}, \mathbf{V}) \\ &= -\frac{1}{2} \text{tr}[\mathbf{V}^{-1}(\text{ERSS} - \mathbf{Y}^\top \mathbf{Y})] - \log \text{BF}^{\text{mix}}(\mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathcal{S}_0, \mathbf{w}_0), \end{aligned} \quad (68)$$

where the the Bayes factor is given in (44). Finally, to arrive at the desired KL-divergences in (67), $D_{\text{KL}}(\hat{q}_j(\mathbf{b}_j) \parallel g(\mathbf{b}_j))$, we set \mathbf{x} to \mathbf{x}_j , and substitute $\hat{\mathbf{R}}_j$ for \mathbf{Y} in (68).

G.4. Estimating g

If we assume that the prior covariances are $\mathbf{S}_{0,k}$ are known, or have been previously estimated using some other statistical procedure, estimating the mixture-of-normals prior g reduces to estimating the mixture weights \mathbf{w}_0 . Fitting the mixture weights by directly optimizing the ELBO (50) is not practical because it involves (intractable) expectations of sums inside logarithms, $E_q[\log g(\mathbf{b}_j)]$. Therefore, to derive a practical approach, we use an augmented *mr.mash* model that is equivalent to the original *mr.mash* model after integrating, or averaging, over the newly introduced variables. We then optimize the ELBO for this augmented model. Although this approach is optimizing a different objective, the updates for \mathbf{V} and $q(\mathbf{B})$ would remain unchanged if we derived them from the augmented model, so we can interpret the above updates as also optimizing the ELBO for the augmented model.

The augmented *mr.mash* model is

$$\begin{aligned} \mathbf{Y} &\sim MN_{n \times r}(\mathbf{X}\mathbf{B}, \mathbf{I}_n, \mathbf{V}) \\ p(\mathbf{b}_j \mid \gamma_j = k) &\sim N_r(\mathbf{0}, \mathbf{S}_{0,k}) \\ p(\gamma_j = k) &= w_{0,k}, \end{aligned} \quad (69)$$

where we have introduced latent indicator variables $\gamma = \{\gamma_1, \dots, \gamma_p\}$, $\gamma_j \in \{1, \dots, K\}$. Integrating over γ recovers the original *mr.mash* model (48).

The ELBO for the augmented model is

$$\tilde{F}(\tilde{q}, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) = E_{\tilde{q}}[\log p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, \mathbf{B})] - \sum_{j=1}^p D_{\text{KL}}(\tilde{q}_j(\mathbf{b}_j, \gamma_j) \parallel p(\mathbf{b}_j, \gamma_j)), \quad (70)$$

in which $\tilde{q}(\mathbf{B}, \gamma)$ is an approximate posterior distribution on \mathbf{B}, γ , and $\tilde{q}_j(\mathbf{b}_j, \gamma_j)$ is the (approximate) marginal posterior for \mathbf{b}_j, γ_j . As before, we restrict $\tilde{q}(\mathbf{B}, \gamma)$ to distributions that factorize over the variables j :

$$\tilde{q}(\mathbf{B}, \gamma) = \prod_{j=1}^p \tilde{q}_j(\mathbf{b}_j, \gamma_j). \quad (71)$$

To derive an update for the mixture weights \mathbf{w}_0 in the mixture-of-normals prior g , we note that g only appears in the KL-divergence terms in (70). Expanding only the parts of the ELBO that involve \mathbf{w}_0 , we have

$$\begin{aligned} \tilde{F}(\tilde{q}, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) &= \sum_{j=1}^p E_{\tilde{q}_j}[\log p(\gamma_j \mid \mathbf{w}_0)] + \text{const}, \\ &= \sum_{j=1}^p \sum_{k=1}^K w_{1,k}^{(j)} \log w_{0,k}, \end{aligned} \quad (72)$$

where $w_{1,k}^{(j)}$ is defined in (59), and “const” is a placeholder for additional terms that do not involve \mathbf{w}_0 . Therefore, the setting of the mixture weights that maximizes the ELBO (70), with all other quantities fixed, is

$$\hat{\mathbf{w}}_0 := \operatorname{argmax}_{\mathbf{w}_0} \tilde{F}(\tilde{q}, g, \mathbf{V}; \mathbf{X}, \mathbf{Y}) = (\hat{w}_{0,1}, \dots, \hat{w}_{0,K}), \quad (73)$$

where $\hat{w}_{0,k} = \frac{1}{p} \sum_{j=1}^p w_{1,k}^{(j)}$. Similar to the update for \mathbf{V} , we can view (73) as an exact M-step with an approximate E-step.

H. Derivation of *mr.mash* variational algorithms with missing data

Here we treat the case in which one or more entries of \mathbf{Y} are missing. Our approach involves treating the missing entries, \mathbf{Y}_{miss} , as random variables, and modeling them using the *mr.mash* model, then formulating a variational approximation with the assumption that \mathbf{Y}_{miss} is independent of \mathbf{B} . One difference compared to the fully observed case is that we explicitly include an intercept in the multivariate regression, and treat it as a free parameter to be estimated (by maximizing the ELBO).

The *mr.mash* model with an intercept is

$$\begin{aligned} \mathbf{Y} &\sim MN_{n \times r}(\mathbf{1}_n \mathbf{b}_0^\top + \mathbf{X} \mathbf{B}, \mathbf{I}_n, \mathbf{V}) \\ \mathbf{b}_j &\stackrel{i.i.d.}{\sim} g, \text{ for } j = 1, \dots, p, \end{aligned} \quad (74)$$

in which $\mathbf{b}_0 \in \mathbb{R}^r$ is the intercept.

Next, taking a variational inference approach, we compute an approximate posterior $q(\mathbf{Y}_{\text{miss}}, \mathbf{B})$, making the following conditional independence assumption:

$$q(\mathbf{Y}_{\text{miss}}, \mathbf{B}) = q(\mathbf{Y}_{\text{miss}}) q(\mathbf{B}), \quad (75)$$

and the approximate posterior for \mathbf{B} factorizes as before (see eq. 9).

Importantly, as we will see, with this conditional independence assumption we can reuse the computations from the fully-observed case. The ELBO with missing data is

$$\begin{aligned} F_{\text{miss}}(q, g, \mathbf{V}, \mathbf{b}_0; \mathbf{X}, \mathbf{Y}_{\text{obs}}) &= \mathbb{E}_q[\log p(\mathbf{Y} \mid \mathbf{X}, \mathbf{V}, \mathbf{b}_0, \mathbf{B})] - \mathbb{E}_q[\log q(\mathbf{Y}_{\text{miss}})] \\ &\quad - \sum_{j=1}^p D_{\text{KL}}(q_j(\mathbf{b}_j) \parallel g(\mathbf{b}_j)), \end{aligned} \quad (76)$$

where \mathbf{Y}_{obs} denotes the observed responses.

In the expressions below, we use $\bar{\mathbf{Y}}$ to denote the “imputed” \mathbf{Y} in which missing entries are replaced by their approximate posterior means $\mathbb{E}_q[\mathbf{Y}_{\text{miss}}]$, and observed entries are copied over from \mathbf{Y}_{obs} . And \mathbf{Y} denotes the combination of the observed and missing values, in which the missing values are treated as random variables.

H.1. Estimating \mathbf{b}_0

First, we focus on the parts of the ELBO (76) that involve \mathbf{b}_0 . We have,

$$F_{\text{miss}}(q, g, \mathbf{V}, \mathbf{b}_0; \mathbf{X}, \mathbf{Y}_{\text{obs}}) = -\frac{n}{2} \mathbf{b}_0^\top \mathbf{V}^{-1} \mathbf{b}_0 - \frac{1}{n} (\bar{\mathbf{m}}_y - \bar{\mathbf{B}}^\top \mathbf{m}_x) \mathbf{V}^{-1} \mathbf{b}_0 + \text{const}, \quad (77)$$

in which $\mathbf{m}_x = \mathbf{X}^\top \mathbf{1}_n / n$ is the vector of length p containing the column means of \mathbf{X} , $\bar{\mathbf{m}}_y = \bar{\mathbf{Y}}^\top \mathbf{1}_n / n$ is the vector of length r containing the column means of $\bar{\mathbf{Y}}$, and the “const” includes the additional terms in the ELBO that do not involve \mathbf{b}_0 . The maximizer is therefore at

$$\hat{\mathbf{b}}_0 = \bar{\mathbf{m}}_y - \bar{\mathbf{B}}^\top \mathbf{m}_x. \quad (78)$$

H.2. Computing approximate posteriors for \mathbf{B}

Using (78), one can show that the following identity holds:

$$\max_{\mathbf{b}_0} F_{\text{miss}}(q, g, \mathbf{V}, \mathbf{b}_0; \mathbf{X}, \mathbf{Y}_{\text{obs}}) = F(q, g, \mathbf{V}; \tilde{\mathbf{X}}, \tilde{\mathbf{Y}}) + \text{const}, \quad (79)$$

in which $\tilde{\mathbf{X}} := \mathbf{X} - \mathbf{1}_n \mathbf{m}_x^\top$ is the column-centered data matrix, $\tilde{\mathbf{Y}} := \bar{\mathbf{Y}} - \mathbf{1}_n \bar{\mathbf{m}}_y^\top$ is the column-centered imputed response matrix, and the “const” includes terms that do not involve \mathbf{B} , \mathbf{V} or the prior, g . Therefore, the problem of optimizing $q(\mathbf{B})$ (as well as estimating \mathbf{V} , \mathbf{w}_0), with $q(\mathbf{Y}_{\text{miss}})$ kept fixed, reduces to the problem of fitting a *mr.mash* model in the fully-observed setting, in which \mathbf{X} is replaced with the column-centered matrix $\tilde{\mathbf{X}}$, and \mathbf{Y} is replaced with the “imputed” matrix $\tilde{\mathbf{Y}}$ that has also been column-centered. So we can reuse the computations for the fully-observed case to implement model fitting for the missing-data case.

H.3. Computing approximate posteriors for \mathbf{Y}_{miss}

Let $\text{miss}(i) \subset \{1, \dots, r\}$ be the subset of dimensions that have missing values in the i th row of \mathbf{Y} , and let the remaining (observed) dimensions be denoted by $\text{obs}(i)$. Fixing $q(\mathbf{B})$ and the parameters of the *mr.mash* model, it can be shown that the approximate posterior for the i th row of \mathbf{Y}_{miss} that maximizes the ELBO,

$$\hat{q}(\mathbf{y}_{i,j}) := \arg\max_{q(\mathbf{y}_{i,j})} F_{\text{miss}}(q, g, \mathbf{V}, \mathbf{b}_0; \mathbf{X}, \mathbf{Y}_{\text{obs}}), \quad (80)$$

in which $j = \text{miss}(i)$, is multivariate normal with mean and covariance

$$\mathbb{E}[\mathbf{y}_{i,j}] = \hat{\mathbf{y}}_{i,j} - \boldsymbol{\Lambda}_{j,j}^{-1} \boldsymbol{\Lambda}_{j,j'} (\mathbf{y}_{i,j'} - \hat{\mathbf{y}}_{i,j'}) \quad (81)$$

$$\text{Var}[\mathbf{y}_{i,j}] = \boldsymbol{\Lambda}_{jj}^{-1}, \quad (82)$$

such that $j' = \text{obs}(i)$, $\boldsymbol{\Lambda} := \mathbf{V}^{-1}$, and $\hat{\mathbf{y}}_i$ denotes the fitted values in the i th row,

$$\hat{\mathbf{y}}_i = \mathbf{b}_0 + \bar{\mathbf{B}}^\top \mathbf{x}_i. \quad (83)$$

H.4. Computing the ELBO

We can also reuse the expressions for the ELBO in the fully-observed case to compute the ELBO for the missing-data case. Assuming \mathbf{b}_0 satisfies (78), the missing-data ELBO (76) can be rewritten as

$$F_{\text{miss}}(q, g, \mathbf{V}, \mathbf{b}_0; \mathbf{X}, \mathbf{Y}_{\text{obs}}) = F(q, g; \mathbf{V}, \tilde{\mathbf{X}}, \tilde{\mathbf{Y}}) - \frac{1}{2} \sum_{i=1}^n \mathbb{E}_q[(\mathbf{y}_i - \bar{\mathbf{y}}_i)^\top \mathbf{V}^{-1} (\mathbf{y}_i - \bar{\mathbf{y}}_i)] - \mathbb{E}_q[\log q(\mathbf{Y}_{\text{miss}})], \quad (84)$$

in which \mathbf{y}_i is the i th row of \mathbf{Y} and $\bar{\mathbf{y}}_i$ is the i th row of $\bar{\mathbf{Y}}$.

I. Implementation notes

Although BMSR-mix outputs all the statistics needed to fully reconstruct the posterior distribution, in practice we do not need to keep track of all these statistics. In particular, if both g and \mathbf{V} are not estimated, we only need to keep track of the marginal posterior means (45). For updating \mathbf{V} , we also need to keep track of the marginal posterior variances (46). If the mixture weights in the prior g are also updated, we also need to keep track of the posterior assignment probabilities $w_{1,k}^{(j)}$ (or sums of these probabilities).

The posterior computations described above are repeated many, many times, so performing these computations efficiently is key to an efficient algorithm for fitting *mr.mash* models. Our general approach is to avoid unnecessarily computation of all inverses, (Cholesky) factorizations or determinants of $r \times r$ matrices that are used to compute the key posterior quantities and the ELBOs by precomputing these quantities when possible. (One has to be careful to update these precomputed quantities whenever the relevant model parameters are also updated.)

The special case in which \mathbf{X} is standardized so that all the columns have unit variance, there are many computations that can be reused. For example, in this special case the posterior covariance from the Bayesian multivariate simple regression with a normal prior (Sec. F.4) is the same for all the variables $j = 1, 2, \dots, p$, and therefore posterior covariances for all the mixture components can be precomputed and reused.

Even when \mathbf{X} is not standardized, posterior computations can still be sped up by precomputing Cholesky factorizations and eigendecompositions of quantities involving \mathbf{V} and $\mathbf{S}_{0,k}$.

J. Proof of Proposition 3

We first show the proof from the maximum-likelihood estimation perspective. We treat $\boldsymbol{\mu}$ as a free parameter to be optimized. The log-likelihood for $\boldsymbol{\mu}$ and \mathbf{b} is

$$\log \ell(\boldsymbol{\mu}, \mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) = -\frac{1}{2} \text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{1}\boldsymbol{\mu}^\top - \mathbf{x}\mathbf{b}^\top)^\top(\mathbf{Y} - \mathbf{1}\boldsymbol{\mu}^\top - \mathbf{x}\mathbf{b}^\top)] + \text{const},$$

in which the “const” includes additional terms that do not involve $\boldsymbol{\mu}$ or \mathbf{b} . Differentiating with respect to $\boldsymbol{\mu}$, setting the partial derivatives to zero, and solving for $\boldsymbol{\mu}$ yields

$$\mathbf{V}^{-1}\boldsymbol{\mu} = \mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b}),$$

and so

$$\hat{\boldsymbol{\mu}} = \bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b}.$$

The profile likelihood for \mathbf{b} is therefore

$$\begin{aligned} \ell^*(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) &:= \max_{\boldsymbol{\mu}} \ell(\boldsymbol{\mu}, \mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) \\ &= |2\pi\mathbf{V}|^{-n/2} \exp\left\{-\frac{1}{2} \text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{1}\hat{\boldsymbol{\mu}}^\top - \mathbf{x}\mathbf{b}^\top)^\top(\mathbf{Y} - \mathbf{1}\hat{\boldsymbol{\mu}}^\top - \mathbf{x}\mathbf{b}^\top)]\right\} \\ &= |2\pi\mathbf{V}|^{-n/2} \exp\left\{-\frac{1}{2} \text{tr}[\mathbf{V}^{-1}((\mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top) - (\mathbf{x} - \bar{\mathbf{x}}\mathbf{1})\mathbf{b}^\top)^\top((\mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top) - (\mathbf{x} - \bar{\mathbf{x}}\mathbf{1})\mathbf{b}^\top)]\right\} \\ &= \ell(\mathbf{b}; \tilde{\mathbf{x}}, \tilde{\mathbf{Y}}, \mathbf{V}). \end{aligned}$$

The conditional posterior for $\boldsymbol{\mu}$ given \mathbf{b} is

$$\begin{aligned} p(\boldsymbol{\mu} \mid \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}, \mathbf{b}) &\propto \exp\left\{-\frac{1}{2}(\boldsymbol{\mu} - n(\mathbf{S}_{0\mu}^{-1} + n\mathbf{V}^{-1})^{-1}\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b}))^\top(\mathbf{S}_{0\mu}^{-1} + n\mathbf{V}^{-1})\right. \\ &\quad \left. \times (\boldsymbol{\mu} - n(\mathbf{S}_{0\mu}^{-1} + n\mathbf{V}^{-1})^{-1}\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b}))\right\}, \end{aligned}$$

which is the multivariate normal density with mean

$$n(\mathbf{S}_{0\mu}^{-1} + n\mathbf{V}^{-1})^{-1}\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b}) = \mathbf{S}_{1\mu}\hat{\mathbf{S}}_{\mu}^{-1}\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}_1,$$

and covariance

$$(\mathbf{S}_{0\mu}^{-1} + n\mathbf{V}^{-1})^{-1} = (\mathbf{S}_{0\mu}^{-1} + \hat{\mathbf{S}}_{\mu}^{-1})^{-1} = \mathbf{S}_{1\mu}.$$

The marginal likelihood for \mathbf{b} is

$$\begin{aligned} \ell^*(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}) &= \int \ell(\boldsymbol{\mu}, \mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}) p(\boldsymbol{\mu} \mid \mathbf{S}_{0\mu}) d\boldsymbol{\mu} \\ &= \int |2\pi\mathbf{S}_{0\mu}|^{-1/2} |2\pi\mathbf{V}|^{-n/2} \\ &\quad \times \exp\left\{-\frac{1}{2} \text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top - \mathbf{1}\boldsymbol{\mu}^\top)^\top(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top - \mathbf{1}\boldsymbol{\mu}^\top) + \mathbf{S}_{0\mu}^{-1}\boldsymbol{\mu}\boldsymbol{\mu}^\top]\right\} d\mathbf{b}_0. \end{aligned}$$

Expanding terms, we get

$$\begin{aligned}
\ell^*(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}) &= |2\pi\mathbf{S}_{0\mu}|^{-1/2} |2\pi\mathbf{V}|^{-n/2} \int \exp\left\{-\frac{1}{2}\text{tr}[(n\mathbf{V}^{-1} + \mathbf{S}_{0\mu}^{-1})\boldsymbol{\mu}\boldsymbol{\mu}^\top - 2\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top \mathbf{1}\boldsymbol{\mu}^\top \right. \\
&\quad \left. + \mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top (\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)]\right\} d\boldsymbol{\mu} \\
&= |2\pi\mathbf{S}_{0\mu}|^{-1/2} |2\pi\mathbf{V}|^{-n/2} \int \exp\left\{-\frac{1}{2}\text{tr}[(n\mathbf{V}^{-1} + \mathbf{S}_{0\mu}^{-1})(\boldsymbol{\mu} - n(n\mathbf{V}^{-1} + \mathbf{S}_{0\mu}^{-1})^{-1}\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b})) \right. \\
&\quad \times (\boldsymbol{\mu} - n(n\mathbf{V}^{-1} + \mathbf{S}_{0\mu}^{-1})^{-1}\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b}))^\top \\
&\quad \left. - n\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b})(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b})^\top n\mathbf{V}^{-1}(n\mathbf{V}^{-1} + \mathbf{S}_{0\mu}^{-1})^{-1} \right. \\
&\quad \left. + \mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top (\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)]\right\} d\boldsymbol{\mu} \\
&= |\mathbf{S}_{0\mu}|^{-1/2} |2\pi\mathbf{V}|^{-n/2} |n\mathbf{V}^{-1} + \mathbf{S}_{0\mu}^{-1}|^{-1/2} \exp\left\{-\frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top (\mathbf{Y} - \mathbf{x}\mathbf{b}^\top) \right. \\
&\quad \left. - n\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b})(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b})^\top n\mathbf{V}^{-1}(n\mathbf{V}^{-1} + \mathbf{S}_{0\mu}^{-1})^{-1}]\right\} \\
&= |2\pi\mathbf{V}|^{-n/2} |\mathbf{S}_{0\mu}^{-1}\mathbf{S}_{1\mu}|^{1/2} \exp\left\{\frac{1}{2}\boldsymbol{\mu}_1^\top \mathbf{S}_{1\mu}^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top (\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)]\right\}.
\end{aligned}$$

When $\mathbf{S}_{0\mu}^{-1} \rightarrow \mathbf{0}$, the marginal likelihood simplifies to

$$\begin{aligned}
\ell^*(\mathbf{b}; \mathbf{x}, \mathbf{Y}, \mathbf{V}, \mathbf{S}_{0\mu}) &= |2\pi\mathbf{V}|^{-n/2} |\mathbf{S}_{0\mu}^{-1}\hat{\mathbf{S}}_\mu|^{1/2} \exp\left\{-\frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{x}\mathbf{b}^\top)^\top (\mathbf{Y} - \mathbf{x}\mathbf{b}^\top) - n\mathbf{V}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b})(\bar{\mathbf{y}} - \bar{\mathbf{x}}\mathbf{b})^\top]\right\} \\
&= |2\pi\mathbf{V}|^{-n/2} |\mathbf{S}_{0\mu}^{-1}\hat{\mathbf{S}}_\mu|^{1/2} \exp\left\{-\frac{1}{2}\text{tr}[\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top - (\mathbf{x} - \bar{\mathbf{x}}\mathbf{1})\mathbf{b}^\top)^\top (\mathbf{Y} - \mathbf{1}\bar{\mathbf{y}}^\top - (\mathbf{x} - \bar{\mathbf{x}}\mathbf{1})\mathbf{b}^\top)]\right\} \\
&= |\mathbf{S}_{0\mu}^{-1}\hat{\mathbf{S}}_\mu|^{1/2} \times \ell(\mathbf{b}; \tilde{\mathbf{x}}, \tilde{\mathbf{Y}}, \mathbf{V}).
\end{aligned}$$

The marginal likelihood for \mathbf{b} in the model with an intercept is therefore proportional to the likelihood for the model without an intercept (eq. 19) when \mathbf{Y} and \mathbf{x} are column-centered.

References

- [1] C. Benner, C. C. Spencer, A. S. Havulinna, V. Salomaa, S. Ripatti, and M. Pirinen. FINEMAP: efficient variable selection using summary data from genome-wide association studies. *Bioinformatics*, 32(10):1493–1501, 2016.
- [2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [3] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: a review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- [4] L. Bottolo and S. Richardson. Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis*, 5(3):583–618, 2010.
- [5] L. Bottolo, E. Petretto, S. Blankenberg, F. Cambien, S. A. Cook, L. Tiret, and S. Richardson. Bayesian detection of expression quantitative trait loci hot spots. *Genetics*, 189(4):1449–1459, 2011.
- [6] L. Bottolo, M. Chadeau-Hyam, D. I. Hastie, T. Zeller, B. Lique, P. Newcombe, L. Yengo, P. S. Wild, A. Schillert, A. Ziegler, S. F. Nielsen, A. S. Butterworth, W. K. Ho, R. Castagné, T. Munzel, D. Tregouet, M. Falchi, F. Cambien, B. G. Nordestgaard, F. Fumeron, A. Tybjærg-Hansen, P. Froguel, J. Danesh, E. Petretto, S. Blankenberg, L. Tiret, and S. Richardson. GUESS-ing polygenic associations with multiple phenotypes using a GPU-based evolutionary stochastic search algorithm. *PLoS Genetics*, 9(8):e1003657, 2013.
- [7] J. Bovy, D. W. Hogg, and S. T. Roweis. Extreme Deconvolution: inferring complete distribution functions from noisy, heterogeneous and incomplete observations. *Annals of Applied Statistics*, 5(2B):1657–1677, 2011.

- [8] P. Carbonetto and M. Stephens. Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Analysis*, 7(1):73–108, 2012.
- [9] M. A. Clyde, J. Ghosh, and M. L. Littman. Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics*, 20(1):80–101, 2011.
- [10] A. P. Dawid. Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.
- [11] P. Dellaportas, J. J. Forster, and I. Ntzoufras. On Bayesian model and variable selection using MCMC. *Statistics and Computing*, 12:27–36, 2002.
- [12] B. E. Engelhardt and M. Stephens. Analysis of population structure: a unifying framework and novel methods based on sparse factor analysis. *PLoS Genetics*, 6(9):e1001117, 2010.
- [13] M. Erbe, B. Hayes, L. Matukumalli, S. Goswami, P. Bowman, C. Reich, B. Mason, and M. Goddard. Improving accuracy of genomic predictions within and between dairy cattle breeds with imputed high-density single nucleotide polymorphism panels. *Journal of Dairy Science*, 95(7):4114–4129, 2012.
- [14] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.
- [15] E. R. Gamazon, H. E. Wheeler, K. P. Shah, S. V. Mozaafari, K. Aquino-Michaels, R. J. Carroll, A. E. Eyler, J. C. Denny, D. L. Nicolae, N. J. Cox, and H. K. Im. A gene-based association method for mapping traits using reference transcriptome data. *Nature Genetics*, 47(9):1091–1098, 2015.
- [16] E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.
- [17] E. I. George and R. E. McCulloch. Approaches to Bayesian variable selection. *Statistica Sinica*, 7:339–373, 1997.
- [18] D. Gianola, G. de los Campos, W. G. Hill, E. Manfredi, and R. Fernando. Additive genetic variability and the Bayesian alphabet. *Genetics*, 183(1):347–363, 2009.
- [19] Y. Guan and M. Stephens. Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *Annals of Applied Statistics*, 5(3):1780–1815, 2011.
- [20] A. Gupta and D. Nagar. *Matrix variate distributions*. Chapman & Hall, Boca Raton, FL, 2000.
- [21] C. Hans, A. Dobra, and M. West. Shotgun stochastic search for “large p” regression. *Journal of the American Statistical Association*, 102(478):507–516, 2007.
- [22] C. J. Hoggart, J. C. Whittaker, M. De Iorio, and D. J. Balding. Simultaneous analysis of all SNPs in genome-wide and re-sequencing association studies. *PLoS Genetics*, 4(7):e1000130, 2008.
- [23] Y. Hu, M. Li, Q. Lu, H. Weng, J. Wang, S. M. Zekavat, Z. Yu, B. Li, J. Gu, S. Muchnik, et al. A statistical framework for cross-tissue transcriptome-wide association analysis. *Nature Genetics*, 51(3):568–576, 2019.
- [24] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.
- [25] S. Lee, J. Zhu, and E. P. Xing. Adaptive multi-task lasso: with application to eQTL detection. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 1306–1314. 2010.
- [26] B. Li, Y. Veturi, A. Verma, Y. Bradford, E. S. Daar, R. M. Gulick, S. A. Riddler, G. K. Robbins, J. L. Lennox, D. W. Haas, and M. D. Ritchie. Tissue specificity-aware TWAS (TSA-TWAS) framework identifies novel associations with metabolic, immunologic, and virologic traits in HIV-positive adults. *PLoS Genetics*, 17(4):e1009464, 2021.

- [27] T. H. Meuwissen, B. J. Hayes, and M. E. Goddard. Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, 157(4):1819–1829, 2001.
- [28] F. Morgante, P. Carbonetto, G. Wang, Y. Zou, A. Sarkar, and M. Stephens. Code and data accompanying this manuscript, 2023. URL <https://doi.org/10.5281/zenodo.8014360>.
- [29] G. Moser, S. H. Lee, B. J. Hayes, M. E. Goddard, N. R. Wray, and P. M. Visscher. Simultaneous discovery, estimation and prediction analysis of complex traits using a Bayesian mixture model. *PLoS Genetics*, 11(4):e1004969, 2015.
- [30] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Springer, New York, 1998.
- [31] P. Perez and G. de los Campos. Genome-wide regression and prediction with the BGLR statistical package. *Genetics*, 198(2):483–495, 2014.
- [32] R Core Team. *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL <https://www.R-project.org>.
- [33] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.
- [34] 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(1):187–195, 2019.
- [35] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2007.
- [36] C. Wallace, A. J. Cutler, N. Pontikos, M. L. Pekalski, O. S. Burren, J. D. Cooper, A. R. García, R. C. Ferreira, H. Guo, N. M. Walker, D. J. Smyth, S. S. Rich, S. Onengut-Gumuscu, S. J. Sawcer, M. Ban, S. Richardson, J. A. Todd, and L. S. Wicker. Dissection of a complex disease susceptibility region using a Bayesian stochastic search approach to fine mapping. *PLoS Genetics*, 11(6):e1005272, 2015.
- [37] G. Wang, A. Sarkar, P. Carbonetto, and M. Stephens. A simple new approach to variable selection in regression, with application to genetic fine mapping. *Journal of the Royal Statistical Society, Series B*, 82(5):1273–1300, 2020.
- [38] W. Wang and M. Stephens. Empirical Bayes matrix factorization. *Journal of Machine Learning Research*, 22(120):1–40, 2021.
- [39] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.
- [40] G. Zanella and G. Roberts. Scalable importance tempering and Bayesian variable selection. *Journal of the Royal Statistical Society, Series B*, 81(3):489–517, 2019.
- [41] X. Zhou, P. Carbonetto, and M. Stephens. Polygenic modeling with Bayesian sparse linear mixed models. *PLoS Genetics*, 9(2):e1003264, 2013.
- [42] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.