UPDATES TO BSFG

Contents

1. abstract

The linear mixed effect model is a workhorse of modern statistical genetics: including GWAS, QTL analysis, Genomic Prediction, Evolutionary Genetics, transcriptomics, growth curve analysis, etc. Recent advances in computational capacity and algorithms has made mixed model accessible to a wide range of researchers. Widely available software has made analyses with thousands (or millions) of individuals feasible. However, most available methods are limited to one (or a few) responses per individual, allow a single random effect (besides the residual), or are limited to Gaussian or other exponential family distributions; non-linear response functions are less common. Here, we propose a general model for high-dimensional linear mixed effect models, which we call SLAMglmm and provide as an extendible R package. SLAMglmm builds on our earlier work (Runcie and Mukherjee 2013) that proposed using sparse factor models to efficiently estimate genetic covariance matrices for high dimensional traits from data on related individuals. We build on the earlier model in four key ways:

(1) Fully generalize the mixed effect model, allowing multiple random effects for both the individual traits and the latent factor traits, and "fixed" effects per-trait and per-factor.
(2) Adapt the discrete prior structure for random effects used for the latent factor traits to the individual traits. This allows more intuitive prior elicitation, especially in models with multiple random effects.
(3) Develop a new, more efficient Gibbs sampler for mixed effect models that greatly improves mixing and posterior convergence.
(4) Add a new level between observed data and the linear mixed effect model based on a flexible link distribution. We develop link distributions for several disparate data types below including: partially missing observations, multiple-probe-per gene data, RNAseq data, time-series data.

The SLAMglmm package is written in R, and draws heavily from the following packages: *Matrix*, *Rcpp*, *RcppArmadillo*, *lme4*, *MCMCglmm*. The model syntax closely follows that of the widely used *lme4* package. Prior specification is similar to *MCMCglmm*.

2. Methods

2.1. SLAMglmm model.

The SLAMglmm model is specified as:

$$\begin{aligned} \mathbf{y}_i &\sim g(\boldsymbol{\eta}_i, \Sigma_i, \theta_y) \\ [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \ldots, \boldsymbol{\eta}_n]^T = \boldsymbol{H} &= \boldsymbol{F}\boldsymbol{\Lambda}^T + \boldsymbol{X}\boldsymbol{B} + \sum \boldsymbol{Z}_i\boldsymbol{U}_{R_i} + \boldsymbol{E}_R \\ \mathbf{F} &= \boldsymbol{X}_F\boldsymbol{B}_F + \sum \boldsymbol{Z}_i\boldsymbol{U}_{F_i} + \boldsymbol{E}_F \end{aligned} \tag{1}$$

$$\begin{aligned} \mathbf{y}_i &\sim g(\boldsymbol{\eta}_i, \Sigma_i, \theta_y) \\ [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \ldots, \boldsymbol{\eta}_n]^T = \boldsymbol{H} &= \boldsymbol{F}\boldsymbol{\Lambda}^T + \boldsymbol{X}\boldsymbol{B} + \sum \boldsymbol{Z}_i\boldsymbol{U}_{R_i} + \boldsymbol{E}_R \\ \mathbf{F} &= \boldsymbol{X}_F\boldsymbol{B}_F + \sum \boldsymbol{Z}_i\boldsymbol{U}_{F_i} + \boldsymbol{E}_F \end{aligned} \tag{2}$$

where $\boldsymbol{y}_i$ is a vector of observations for the ith individual, $\boldsymbol{\eta}_i$ is a vector of p potentially unobserved characteristics (traits) for the ith individual. These may correspond directly to the elements of $\boldsymbol{y}_i$, or may be parameters of a more complicated function / distribution g. $\boldsymbol{H}$ is an $n \times p$ matrix of these characteristics for the n individuals. In the original BSFG model, g was the identity function, so $\boldsymbol{Y} = \boldsymbol{H}$. The utility of this hierarchical specification will be demonstrated below.

In 1, $\boldsymbol{H}$ and $\boldsymbol{F}$ are $n \times p$ and $n \times k$ matrices of latent traits for the n individuals. These $p + k$ latent traits are related through a structural equation model given by the $p \times k$ factor loadings matrix $\boldsymbol{\Lambda}$. This structural equation model is the key feature of SLAMglmm (and BSFG), as the paths described by $\boldsymbol{\Lambda}$ explain all of the covariance among the $p + k$ traits; the k traits $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_k]$ are assumed to be independent, and the p traits $\boldsymbol{H} = [\boldsymbol{\eta}_{\bullet 1}, \boldsymbol{\eta}_{\bullet 2}, \ldots, \boldsymbol{\eta}_{\bullet p}]$ are conditionally independent, conditional on $\boldsymbol{F}$ and $\boldsymbol{\Lambda}$. Through priors on $\boldsymbol{\Lambda}$ (and k), we impose sparsity on the among-trait covariances for all random effects, ensuring that the model can scale efficiently with increasing numbers of traits.

2.1.1. *fixed effects.* The matrices $\boldsymbol{B}$ and $\boldsymbol{B}_F$ are $b \times p$ and $b_F \times k$ matrices of fixed effect coefficients for each of the $p + k$ latent traits, corresponding to the fixed effect design matrices $\boldsymbol{X}$ and $\boldsymbol{X}_F$. Fixed effects are used to model the effect of individual-level covariates and design features such as sex, gender, environment, or genetic loci. In BSFG, we allowed fixed effects only for the p observational-level traits (ie $\boldsymbol{B}$), and modeled them with a flat prior (independent Gaussian distributions with variance = 10^6). Here, we generalize this to allow fixed effects for the k factors. However, by simply introducing $\boldsymbol{X}_F$ with similarly flat priors, the $\boldsymbol{B}$ and $\boldsymbol{B}_F$ would not be identifiable, as all association between the covariates and $\boldsymbol{H}$ could be explained by $\boldsymbol{B}$. Here, we address this problem by favoring solutions with fewer large coefficients in $\boldsymbol{B}$ and $\boldsymbol{B}_F$; solutions with a small number of factor coefficients in $\boldsymbol{B}_F$ that explain the trait-covariate associations for many traits are favored over solutions with a larger number of trait-specific coefficients in $\boldsymbol{B}$. We apply ARD-type priors to $\boldsymbol{B}$ and $\boldsymbol{B}_F$ to favor these sparser solutions, with covariate-specific (row) shrinkage on the coefficients of each matrix:

$$
\begin{aligned}
\mathbf{B} &= [b_{ij}], \;\; b_{ij} \sim \mathrm{N}(0, \tau_{B_i}^{-1}\psi_{B_{ij}}^{-1}), i \in 1 \dots b, j \in 1 \dots p \\
\mathbf{B}_F &= [b_{ij}^F], \;\; b_{ij}^F \sim \mathrm{N}(0, \tau_{B_i^F}^{-1}\psi_{B_{ij}^F}^{-1}), i \in 1 \dots b_F, j \in 1 \dots k \\
\tau_{B_i} &\sim Ga(\alpha_b, \beta_b), \tau_{B_i^F} \sim Ga(\alpha_{b_F}, \beta_{b_F}) \\
\psi_{B_{ij}} &\sim Ga(\nu_b/2, \nu_b/2), \psi_{B_{ij}^F} \sim Ga(\nu_{b_F}/2, \nu_{b_F}/2)
\end{aligned} \tag{3}
$$

with τ_{B_i} and $\tau_{B_i^F}$ providing row-shrinkage on the matrices, and $\psi_{B_{ij}}$ and $\psi_{B_{ij}^F}$ providing the ARD on each element of each matrix.

In general, we allow the fixed effect design matrices $\boldsymbol{X}$ and $\boldsymbol{X}_F$ to be similar or different depending on context. However we impose two additional constraints. First, we assume that the first column of $\boldsymbol{X}$ corresponds to a global intercept, and set $\sigma_{b_1}^2 = \mathrm{Inf}$ so as not to penalize this coefficient. Second, we force the intercept of each of the $\boldsymbol{f}_j$ traits to be zero by setting $\boldsymbol{B}_{F_{1\bullet}} = 0$ and the corresponding variance to be zero as well.

2.1.2. *random effects.* The matrices $\boldsymbol{U}_{R_i}$ and $\boldsymbol{U}_{F_i}$ are $r \times p$ and $r \times k$ coefficient matrices for the ith random effect. As in BSFG, columns of these matrices are independent multivariate normal distributions:

$$
\begin{aligned}
\mathbf{U}_{R_i} &= [\boldsymbol{u}_{1_i}, \boldsymbol{u}_{2_i}, \dots, \boldsymbol{u}_{p_i}], \;\; \boldsymbol{u}_{j_i} \sim \mathrm{N}_{r_i}(\mathbf{0}, \sigma_{R_j}^2 h_{R_{i_j}}^2 \boldsymbol{K}_i) \\
\mathbf{U}_{F_i} &= [\boldsymbol{u}_{1_i}^F, \boldsymbol{u}_{2_i}^F, \dots, \boldsymbol{u}_{k_i}^F], \;\; \boldsymbol{u}_{j_i}^F \sim \mathrm{N}_{r_i}(\mathbf{0}, h_{F_{i_j}}^2 \boldsymbol{K}_i) \\
\boldsymbol{\Sigma}_{h_i^2}^R &= \mathrm{Diag}(h_{R_{i_j}}^2), \;\; \boldsymbol{\Psi}^R = \mathrm{Diag}(\sigma_{R_{i_j}}^2) \\
\boldsymbol{\Sigma}_{h_i^2}^F &= \mathrm{Diag}(h_{F_{i_j}}^2)
\end{aligned} \tag{4}
$$

where the $\boldsymbol{K}_i, i \in 1 \dots R$ are $r_i \times r$ "kinship" matrices describing expected covariances of random effects. These replace the additive-genetic covariance matrix $\boldsymbol{A}$ used in BSFG, and could be any positive-semidefinite matrices. The number of levels for each of the R random effects, r_i may not equal n. The design matrices $\boldsymbol{Z}_i$ are $n \times r_i$ matrices linking each random effect level to a corresponding individual. BSFG allowed only a single random effect for

the factors, although a second random effect was used for the p observational-level traits in one example.

The specification of the column-variances of $\boldsymbol{U}_{R_i}$ is new relative to BSFG (this replaces $\boldsymbol{E}_a$). Instead of a single $\sigma^2_{a_j}$, we decompose this variance into a total variance $\sigma^2_{R_j}$ and a percentage of the total attributable to the ith random effect $h^2_{R_{i_j}} = \sigma^2_{a_i}/\sigma^2_p$. This is now identical to the random effect variance specification for the factors $\boldsymbol{f}_j$, except that for the factors $\sigma^2_{F_j} = 1$. This re-parameterization makes prior elicitation easier, as elaborated below in section 3.2.

2.1.3. *link distribution.* Several link distributions are described below in section 4. Beyond the identity $\boldsymbol{y}_i = \boldsymbol{\eta}_i$, the next simplest link function is of the form:

$$\mathbf{y}_i \sim \mathrm{N}(g(\boldsymbol{\eta}_i), \Sigma_y) \tag{5}$$

2.1.4. *prior on factors.* We use the same prior on $\boldsymbol{\Lambda}$ as in BSFG. This is the "infinite factor model" prior proposed by Bhattacharya and Dunson (2011) that imposes increasing column-wise shrinkage on higher order columns, which imposes sparsity by shrinking the number of important factors (ie paths), as well as element-wise shrinkage on each element of $\boldsymbol{\Lambda}$ through and ARD-type prior. Other priors such as the *TPB* prior of Engelhardt could be substituted.

3. New concepts

As mentioned above, SLAMglmm proposes three new features that together greatly facility the analysis of high-dimensional linear mixed effect models. These are described more here:

3.1. **Structural equation models of among-trait covariances.** The central statistical challenge of large multivariate mixed effect models is that the number of parameters necessary to specify the among-trait covariance matrices grows as $p \times (p-1)/2 \times (R+1)$ with the number of traits and the number of random effects. Unconstrained estimates of all these covariance parameters requires an unrealistic number of observations for even moderately-sized trait vectors,

As proposed for BSFG, SLAMglmm uses a hierarchical sparse-factor structural equation model to explain the among-trait covariance structure. This prioritizes the strongest, most important covariance signals in the data. A key development in BSFG was the idea that a single set of factors $\boldsymbol{\Lambda}$ could be used to explain both the additive-genetic and residual covariances, with the the factors re-weighted for each covariance matrix:

$$\begin{aligned} \mathbf{G} &= \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{h^2}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}_G \\ \mathbf{R} &= \boldsymbol{\Lambda}(\boldsymbol{I}_k - \boldsymbol{\Sigma}_{h^2})\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}_R \end{aligned} \tag{6}$$

This "sharing" of the factors between the covariance matrices does not force them to be similar: whenever one of the diagonal elements of $\boldsymbol{\Sigma}_{\boldsymbol{h^2}}$ ($\sigma_{h^2_j}$) equals 0(1), the column contributes only to the residual (additive-genetic) covariance matrix. But when the same factor contributes to both covariances, the same $\boldsymbol{\lambda}_j$ parameters can be re-used.

SLAMglmm uses this same strategy to efficiently estimate a set of $R+1$ covariance matrices for the R random effects and the residuals. Each diagonal element of the $\mathbf{\Sigma}^F_{h_i^2}$ matrices is allowed to equal 0 or 1, but can also take values inbetween, providing flexible sharing of covariance components among random effects:

$$\begin{aligned} \mathbf{\Sigma}_i &= \mathbf{\Lambda}\mathbf{\Sigma}^F_{h_i^2}\mathbf{\Lambda}^T + \mathbf{\Sigma}^R_{h_i^2}\mathbf{\Psi}^R \\ \mathbf{\Sigma}_R &= \mathbf{\Lambda}(\boldsymbol{I}_k - \sum \mathbf{\Sigma}^F_{h_i^2})\mathbf{\Lambda}^T + (\boldsymbol{I}_p - \sum \mathbf{\Sigma}^R_{h_i^2})\mathbf{\Psi}^R \end{aligned} \tag{7}$$

SLAMglmm also extends BSFG by leveraging the latent factors to estimate the fixed effects. If the the same sets of traits are associated with the fixed effect covariates as distinguish the random effect groupings (or residuals), the same factors can also be used to explain the fixed effect responses. These factor - covariate associations can be explored directly (ex sex or condition effects on the latent traits), or used to provide more robust fixed effect coefficients for the observation-level traits.

3.2. **prior elicitation.** Intuitive prior distributions are important for effective prior elicitation in Bayesian models. In most implementations of mixed effect models, inverse-Gamma priors are used for the variance components. This prior form is useful because of conjugacy to the likelihood. However, specifying hyperparameters for multiple variance components of this form can be difficult, especially when the prior information is highly diffuse. Gibbs samplers based on inverse-Gamma priors are known to have poor mixing properties, and so "parameter-expanded" forms are commonly used (ex. *MCMCglmm*, Gelman et al 2006?), which improve computation, but can complicate prior elicitation. These problems are exacerbated in the multivariate mixed model. The conjugate distribution of the covariance of multivariate random effects is the inverse-Wishart distribution, which is not very flexible, especially for vague prior knowledge.

In BSFG, we avoided the inverse-Wishart distribution by the hierarchical factor structure, with prior covariance specified through the prior on $\mathbf{\Lambda}$. For the residual variances of each of the p observational-level traits, we did use inverse-Gamma priors. But for priors for the factor variances, we re-formulated the random effect model in terms of heritability and total variance, and used a recently proposed discrete prior on heritability (Zhou?). Heritability ($h^2 = \sigma^2_a/(\sigma^2_p)$) defined as the proportion of variance attributable to genetics, is more intuitive and easier to visualize than a variance, and is independent of the scale of the data, making prior elicitation easier. The discrete distribution is extremely flexible for conveying specific priors. The discrete prior also facilitates a highly efficient partially collapsed Gibbs sampler for the factor traits which we described in Runcie and Mukherjee 2013 and detail below in section 3.3

In SLAMglmm, we extend this re-formulation of the mixed effect model to the observational-level traits as well. Conditional on the latent factors, the covariance of a single trait (j) is:

$$\begin{aligned} \sigma(\boldsymbol{\eta}_{\bullet j})|\mathbf{\Lambda}, \boldsymbol{F}, \boldsymbol{B} = \boldsymbol{P} &= \sum \sigma^2_{R_{i_j}} \boldsymbol{K}_i + \sigma^2_{R_{e_j}} \boldsymbol{I}_n \\ &= \sigma^2_{R_{P_j}} (\sum h^2_{R_{i_j}} \boldsymbol{K}_i + (1 - \sum h^2_{R_{i_j}}) \boldsymbol{I}_n) \end{aligned} \tag{8}$$

and we let $\boldsymbol{\Sigma}_i^R = \text{Diag}(h^2_{R_{ij}})$ and $\boldsymbol{\Psi}^R = \text{Diag}(\sigma^2_{R_{P_j}})$.

Thus, for each trait, we specify a total variance $\sigma^2_{R_{P_j}}$, and then a set of R variance fractions $h^2_{R_{ij}}$ corresponding to each random effect, the sum of which is less than 1. For prior elicitation, we specify a joint prior on these R variance fractions using an evenly spaced grid over all valid combinations, and then can elicit priors based on marginal distributions or over the full space. The prior on $\sigma^2_{R_{ij}}$ is an independent inverse-Gamma distribution for conjugacy.

Extending the formulation of BSFG, the random effect structure of each of the k factors is specified similarly, conditional on the fixed effects:

$$\begin{aligned} \sigma(\boldsymbol{f}_{\bullet j})|\boldsymbol{B}_F &= \sum \sigma^2_{F_{ij}} \boldsymbol{K}_i + \sigma^2_{F_{e_j}} \boldsymbol{I}_n \\ &= \sigma^2_{F_{P_j}} (\sum h^2_{F_{ij}} \boldsymbol{K}_i + (1 - \sum h^2_{F_{ij}}) \boldsymbol{I}_n) \end{aligned} \tag{9}$$

with the exception that we constrain $\sigma^2_{F_{P_j}} = 1$ for identifiability. However, to improve mixing of $\boldsymbol{\Lambda}$ and $\boldsymbol{F}$, we implement the parameter expansion proposed by (Ghosh and Dunson 2009) for factor models by allowing $\sigma^2_{F_{P_j}} \neq 1$, but then correcting $\boldsymbol{F}, \boldsymbol{U_F}$, and $\boldsymbol{\Lambda}$ by this factor in the posterior (see 3.3).

3.3. **Gibbs sampler.** In the SLAMglmm R package we develop a new Gibbs sampler with considerable performance enhancements both in per-iteration speed and especially in parameter mixing relative to BSFG. The key feature of the sampler is that by re-parameterizing the random effects in terms of proportions of variance rather than variance components, and restricting the prior to a discrete set of variance proportions (h^2_{ij}'s), we can sample these proportions directly in a single Gibbs or MH step, marginalizing over all random effects. The individual random effects can be sampled afterwards conditioning on the current values of the h^2_{ij}'s. This partial collapsing of the Gibbs sampler (XX) significantly reduces the posterior correlation between the random effects and their variance components, a feature that commonly plagues MCMC algorithms for mixed effect models (MCMCglmm, Gelman (2006)?). The Gibbs algorithm iterates through the following steps:

1. Sample $\boldsymbol{B}$ and $\boldsymbol{\Lambda}$ jointly, conditioning on $\boldsymbol{F}, \boldsymbol{\Sigma}^R_{h^2}, \boldsymbol{\Psi}^R$, and the prior precisions of $\boldsymbol{B}$ and $\boldsymbol{\Lambda}$, but marginalizing over $\boldsymbol{U}_{R_i}$ and $\boldsymbol{E}_{R_i}$.
2. Sample $\boldsymbol{\Psi}^R$, conditioning on $\boldsymbol{B}, \boldsymbol{\Lambda}, \boldsymbol{F}$, and $\boldsymbol{\Sigma}^R_{h^2}$, still marginalizing over $\boldsymbol{U}_{R_i}$ and $\boldsymbol{E}_{R_i}$.
3. Sample $\boldsymbol{\Sigma}^R_{h^2}$ jointly, conditioning on $\boldsymbol{B}, \boldsymbol{\Lambda}, \boldsymbol{F}$, and $\boldsymbol{\Psi}^R$, still marginalizing over $\boldsymbol{U}_{R_i}$ and $\boldsymbol{E}_{R_i}$.
4. Finally, sample the $\boldsymbol{U}_{R_i}$ jointly, conditioning on $\boldsymbol{B}, \boldsymbol{\Lambda}, \boldsymbol{F}, \boldsymbol{\Psi}^R$, and $\boldsymbol{\Sigma}^R_{h^2}$.
5. Sample $\boldsymbol{B}_F$, conditioning on $\boldsymbol{F}, \boldsymbol{\Sigma}^F_{h^2}, \boldsymbol{\Psi}^F$, and the prior precisions of $\boldsymbol{B}$, but marginalizing over $\boldsymbol{U}_{F_i}$ and $\boldsymbol{E}_F$.
6. Sample $\boldsymbol{\Psi}^F$, conditioning on $\boldsymbol{F}, \boldsymbol{\Sigma}^F_{h^2}$, and $\boldsymbol{B}_F$, still marginalizing over $\boldsymbol{U}_{F_i}$ and $\boldsymbol{E}_F$.
7. Sample $\boldsymbol{\Sigma}^F_{h^2}$ jointly, conditioning on $\boldsymbol{F}, \boldsymbol{\Psi}^F$, and $\boldsymbol{B}_F$, still marginalizing over $\boldsymbol{U}_{F_i}$ and $\boldsymbol{E}_F$.

(8) Finally, sample $\boldsymbol{U}_{F_i}$ jointly, conditioning on $\boldsymbol{F}, \boldsymbol{\Sigma}_{h^2}^F, \boldsymbol{\Psi}^F$, and $\boldsymbol{B}_F$.
(9) Sample $\boldsymbol{F}$, conditioning on all other parameters.
(10) Sample precision parameters of $\boldsymbol{B}$ and $\boldsymbol{\Lambda}$.
(11) Sample $\boldsymbol{H} = [\boldsymbol{\eta}_i]^T$ conditioning on all other parameters and the observed data $\boldsymbol{y}_i$.

Outside of the Gibbs iteration, posterior samples of $\boldsymbol{F}$, $\boldsymbol{U}_{F_i}$ and $\boldsymbol{B}_F$ are re-normalized by post-multiplying by$\boldsymbol{\Psi}^{F^{-1/2}}$ so that $\boldsymbol{F}$ has unit total variance. Similarly, posterior samples of $\boldsymbol{\Lambda}$ are re-nomralized by post-multiplying by $\boldsymbol{\Psi}^{F^{1/2}}$ to keep the product $\boldsymbol{F}\boldsymbol{\Lambda}^T$ the same.

The key feature of this Gibbs algorithm is the partial collapsing over the random effect parameters in steps 1-3 and 5-7. This dramatically improves mixing of the sampler. Step 7 is (nearly) identical to the BSFG Gibbs sampler, but the other steps are new in SLAMglmm.

The collapsed sampling in this algorithm is facilitated by several computational tricks avoiding repeated matrix inversions.

The first involves the method used by *MCMCglmm* to sample location effects from the mixed model equations using only a single cholesky decomposition. This algorithm is used in steps 1, 4, 5, and 8.

The second uses the fact that the number of covariance matrices for the p observation-level and k factor traits is limited by the discrete prior structure, up to a scalar factor determined by $\boldsymbol{\Psi}^R$ or $\boldsymbol{\Psi}^F$. Unless many random effects are specified, and the discrete prior has many divisions, the number of possible covariance matrices is typically much smaller than the number of traits times the number of iterations necessary to generate sufficient posterior samples. Therefore computational speedups are possible by pre-calculating matrix inversions and cholesky decompositions of every possible covariance matrix up-front (which can be done in a parallel fashion), and then re-using the decompositions repeatedly across traits and Gibbs iterations. This technique is facilitated by the fact that all possible covariance matrices have the same patters of non-zero entries, so a sparse symbolic Cholesky decomposition can be used to speedup the matrix inversions once the first decomposition has been calculated. Sparse matrices are used throughout the implementation of SLAMglmm for efficient computation.

Finally, in the specific case when only a single random effect is used, additional speedup can be achieved by pre-calculating matrices that diagonalize $\boldsymbol{Z}\boldsymbol{K}\boldsymbol{Z}^T$ and the pair $(\boldsymbol{Z}^T\boldsymbol{Z}, \boldsymbol{K}^{-1})$ allowing the mixed model equations to be solved without matrix inversions. The SLAMglmm package implements two Gibbs samplers, one ('general') that works with any number of random effects, and one ('fast') optimized for a single random effect.

3.4. **R package.** We provide SLAMglmm as an R package available here: `https://github.com/deruncie/SparseFactorMixedModel`. At the user level, SLAM is designed to work similarly to the widely used packages *lme4* and *MCMCglmm*. Models including both fixed and random effects are specified for the latent factors ($\boldsymbol{f}_j$) and the trait residuals ($\boldsymbol{\eta}_j - \boldsymbol{F}\boldsymbol{\lambda}_{\bullet j}^T$) are specified symbolically using the same syntax as *lmer*. Prior specification is similar to *MCMCglmm*. The format for the link functions is relatively straightforward, and several commonly-used functions are provided. Fitting a model involves:

- Prepare data as for *lmer* or *MCMCglmm*

- Initialize model. This provides initial values for all parameters and pre-calculates several factors that are repeatedly used throughout the computation
- Call the sampler function for a specific number of iterations.
- Check for convergence with a set of diagnostic plots, or using the diagnostics of *shinystan*
- If model has not converged, re-call the sampler. The state of the random number generator is saved so that additional samples are the same as if the chain had not been interrupted.
- Posterior samples are saved in a format compatible with functions in *MCMCpack* or *rstan*.

4. Case studies

Here, we describe several case studies that demonstrate potential uses of SLAMglmm, based on different link distributions g.

4.1. **Partial missing data.** In the original BSFG paper, we analyzed the Ayroles 2009 gene expression data in conjunction with data on the fitness of each line. Since flies with expression data were not evaluated for fitness, and flies with fitness data were not evaluated for gene expression, we treated the un-observed traits for each fly as missing data and included a step in the Gibbs sampler to impute these missing values conditional on the current state of the model parameters. This was possible because conditional on all other parameters, each missing data point followed an independent Gaussian distribution.

Here, we generalize this imputation step as the following link distribution:

$$\begin{aligned} \eta_{ij} &\sim \begin{cases} y_{ij} & \text{if } y_{ij} \text{ is observed} \\ \mathrm{N}(\hat{\eta_{ij}}, \sigma^2_{R_{i_j}}(1 - \sum h^2_{R_{i_j}})) & \text{if } y_{ij} \text{ is not observed} \end{cases} \\ \hat{\eta_{ij}} &= \boldsymbol{x}_{\bullet i}\boldsymbol{B} + \boldsymbol{f}_{\bullet i}\boldsymbol{\Lambda}^T + \sum \boldsymbol{Z}_i \boldsymbol{u}_{F_{\bullet i}} \end{aligned} \tag{10}$$

Updates for $\boldsymbol{\eta}_i$ for only those un-observed traits are independent draws from univariate Gaussian distributions.

4.2. **Microarray with multiple probes per gene.** Microarrays for measuring gene expression commonly have multiple separate probes per gene on the same slide. These probes provide technical replication for gene (transcript) quantification. However, conditional on transcript expression, probes on the same gene, and probes on different genes should be uncorrelated (once arrays are normalized). In this case, we use the link distribution to move from the observational data (probes per transcript per individual, $\boldsymbol{y}_i$ a $(pd) \times 1$ vector with d probes per transcript to a multivariate (correlated) mixed effect model on the transcripts ($\boldsymbol{\eta}_i$, a $p \times 1$ vector). The link distribution is:

$$\begin{aligned} \mathbf{y}_i &\sim \mathrm{N}(\boldsymbol{X}_Y \boldsymbol{\eta}_i, \boldsymbol{\Psi}^P) \\ \boldsymbol{\eta}_i &\sim \mathrm{N}(\hat{\boldsymbol{\eta_{ij}}}, \sigma^2_{R_{i_j}}(1 - \sum h^2_{R_{i_j}})) \end{aligned} \tag{11}$$

with $\hat{\boldsymbol{\eta}}_i$ specified as above, and $\boldsymbol{\Psi}^P$ a $pd \times pd$ diagonal matrix of probe-specific technical variances. Updates for $\boldsymbol{\eta}_i$ are independent draws from univariate Gaussian distribution because of the orthogonal structure of $\boldsymbol{X}_Y$. Updates for the diagonal elements of $\boldsymbol{\Psi}^P$ are independent draws from inverse-Gamma distributions, based on a conjugate prior.

This model was used in a study by Hine et al (submitted) on gene expression pleiotropy in mutation-accumulation lines of Drosophila.

4.3. **RNAseq data.** RNAseq data are counts of transcripts (or transcript fragments). These counts are expected to follow a Poisson distribution around the true transcript expression. Most methods for analyzing RNAseq data rely on generalized linear models of either over-dispersed Poisson or Negative-Binomial distributions to account for this expected sampling distribution. However, recent research has shown that it is more important to accurately model the observation-specific variances than the exact form of the sampling distribution, and that when log-transformed, heteroscedastic linear models generally outperform the generalized linear models with Poisson of NB distributions. This technique is implemented in the *voom* function of the *limma* package originally designed for microarray data. *voom* performs the log-transformation on RNAseq counts and uses a spline function to estimate the mean-variance relationship across all genes, with optional sample-specific weights as well. The log-counts-per-million estimate as well as a precision-weight for each observation are outputted. *limma* provides limited support for mixed effect models (single repeated measures random effects for balanced data, I think), but does not consider among-gene covariance.

We propose to model RNAseq data following the *voom* method by linking observations $\boldsymbol{y}_i$ to latent true transcript log-cpm $\boldsymbol{\eta}_i$ using the estimated observation-specific precision weights:

$$\begin{aligned} \mathbf{y}_i &\sim \mathrm{N}(\boldsymbol{\eta}_i, \boldsymbol{\Psi}_i^w) \\ \boldsymbol{\eta}_i &\sim \mathrm{N}(\hat{\boldsymbol{\eta}_{ij}}, \sigma^2_{R_{ij}}(1 - \sum h^2_{R_{ij}})) \end{aligned} \tag{12}$$

with $\hat{\boldsymbol{\eta}}_i$ specified as above, and $\boldsymbol{\Psi}_i^w$ a $p \times p$ diagonal matrix of transcript-specific inverse-precision weights for the p genes of sample i. Updates for $\boldsymbol{\eta}_i$ are independent draws from univariate Gaussian distributions.

4.4. **Time-series data.** The above link-distributions cover a wide range of potential applications for SLAMglmm. However, potentially more interesting applications involve considerations of more complex observations of individual subjects, such as time-series data, growth model analysis, or shape analysis. The general idea of this set of applications is that a (potentially non-)linear model with parameters $\boldsymbol{\eta}_i$ could be fit to the vector of observations $\boldsymbol{y}_i$ of each individual separately. But these individual-specific parameters (potentially after pre-defined non-linear transformations) could be modeled as correlated latent traits using SLAMglmm. This way highly-parameterized models could be "regularized" based on expected similarities among individuals based on the fixed and random effects.

A motivating example is time-series measurements on a set of related individuals. Time series data has an inherent covariance structure. However, a popular approach is to approximate this covariance using splines for each individual (sometimes termed Random Regression). Here, we implement Random Regression using b-splines in SLAMglmm. In particular, a b-spline basis is calculated spanning the range of observation times across all individuals, potentially with a large number of knots. Using this basis function, design matrices $\boldsymbol{X}_i^s$ are calculated for each individual. It is not necessary that all individuals have the same number of observations, or are observed at the same time. The link distribution is:

$$\begin{aligned} \mathbf{y}_i &\sim \mathrm{N}(\boldsymbol{X}_i^s \boldsymbol{\eta}_i, \sigma_y^2 \boldsymbol{I}_{n_i}) \\ \boldsymbol{\eta}_i &\sim \mathrm{N}(\hat{\boldsymbol{\eta}_{ij}}, \sigma^2_{R_{i_j}}(1 - \sum h^2_{R_{i_j}})) \end{aligned} \tag{13}$$

with $\hat{\boldsymbol{\eta}}_i$ specified as above, and σ_y^2 is a single observation-level variance for all observations. This could potentially be generalized to be a function of time. Updates for $\boldsymbol{\eta}_i$ are independent draws from a multivariate Gaussian distribution. This is similar to GAMMs, except the penalization of spline coefficients is different.

4.5. **Genomic Selction.** Genomic Selection (GS), or Genomic Prediction, is related to GWAS, and uses dense marker data to build a predictive model for traits given genotypes. A number of GS algorithms have been proposed, most building on the linear mixed effect model with varying prior distributions on marker coefficients. However, few models have explores GS on multivariate traits. Potential uses include predicting traits values across multiple environments, simultaneously predicting suites of traits, or predicting later-development traits based on early-development observations. For example, in a plant breeding program, if a GS model could be trained on a trait vector including seedling gene expression and final yield, seedling tissue could be harvested on new plants and fed into the GS model to perform indirect selection on yield without having to grow all plants through the whole life cycle.

An efficient GS model may include all marker data in $\boldsymbol{X}_F$ (where all information on trait covariances must lie), but only limited covariates may be needed in $\boldsymbol{X}$. This would be a massive dimension reduction from a full $m \times p$ GS model for m markers and p traits to a $m \times k$ GS model only on the factor traits

4.6. **QTL mapping and GWAS for expression traits.** QTL or GWAS methods differ from GS methods in that their goal is to identify specific genomic loci associated with trait variation, rather than phenotype prediction *per se*. Therefore, priors on $\boldsymbol{B}$ and $\boldsymbol{B}_F$ should be tailored to favor especially sparse solutions. The mixed prior of BSLMM (Zhou et al 2013) may be useful for GWAS, while Bayesian LASSO - type priors have been used for QTL mapping. However most QTL models treat QTL genotypes as unknowns and attempt to identify QTL between loci (predictors in $\boldsymbol{X}$). This could be built in to SLAMglmm in an additional step to infer $\mathbf{X}$ genotypes in a grid along the chromosomes.

Another potential modification of the model for SLAMglmm would be to specify a unique *cis*-model for each gene. *cis*-variants are extremely common in population samples

of gene expression, but many of these changes may not be associated with variation in gene networks. Therefore, a link-distribution could estimate *cis*-effects based on local genotypes for each gene, and then model the covariance among the residuals for each gene using the sparse factor model to search for *trans*-eQTL. Almost by definition, *trans*-eQTL should be captured by factors, while *cis*-eQTL should be limited to the residual (or observational)-level trait variation.

5. Statistical issues

The following are several statistical issues with SLAMglmm that should be addressed:

5.1. **Identifiability of fixed effects.**

In SLAMglmm, we expand the model for the effects of fixed effects beyond BSFG. In BSFG, fixed effects $\boldsymbol{XB}$ were specified as:

$$\mathbf{Y} = \boldsymbol{XB} + \boldsymbol{F}\boldsymbol{\Lambda}^T + \boldsymbol{E}^*$$

where $\boldsymbol{E}^*$ is a random matrix representing the sum of all random effects (genetic and residuals).

In SLAMglmm, effects of the fixed covariates $\boldsymbol{X}$ and $\boldsymbol{X}_F$ are modeled in two locations:

$$\begin{aligned}\mathbf{H} &= \boldsymbol{XB} + \boldsymbol{F}\boldsymbol{\Lambda}^T + \boldsymbol{E}^* \\ \mathbf{F} &= \boldsymbol{X}_F\boldsymbol{B}_F + \boldsymbol{E}^*_F\end{aligned}$$

Expanding this out, we have:

$$\begin{aligned}\mathbf{H} &= \boldsymbol{XB} + [\boldsymbol{X}_F\boldsymbol{B}_F + \boldsymbol{E}^*_F]\boldsymbol{\Lambda}^T + \boldsymbol{E}^* \\ &= [\boldsymbol{XB} + \boldsymbol{X}_F\boldsymbol{B}_F\boldsymbol{\Lambda}^T] + \boldsymbol{E}^*_F\boldsymbol{\Lambda}^T + \boldsymbol{E}^*\end{aligned}$$

with the brackets highlighting the fact that there are really two parallel models for the fixed effects. In the case that $\boldsymbol{X} = \boldsymbol{X}_F$, either $\boldsymbol{B}$ or $\boldsymbol{B}_F\boldsymbol{\Lambda}^T$ could account for the relationship between $\boldsymbol{X}$ and $\boldsymbol{H}$. $\boldsymbol{B}$ has $b \times p$ parameters and $\boldsymbol{B}_F\boldsymbol{\Lambda}^T$ has $b \times k + k \times p$ parameters. When $b > k$, the factors provide a potentially more parsimonious model for these fixed effects, under the assumption that these effects may be correlated. When $b < k$, $\boldsymbol{B}$ has fewer parameters. However, the $k \times p$ parameters of $\boldsymbol{\Lambda}^T$ are also shared by the "genetic" model: $\boldsymbol{E}^*_F\boldsymbol{\Lambda}^T$, so they come for "free". In this case, $\boldsymbol{B}_F$ should be used if the among-trait correlation induced by the fixed effects is similar to that caused by genetic effects or microenvironments, while $\boldsymbol{B}$ should be chosen if the fixed effects induce correlation that is not related to the underlying biology (ex. batch or lane effects).

The logic of this model is that the columns of $\boldsymbol{F}$ are considered to be "traits" just like the columns of $\boldsymbol{Y}$ (BSFG) or $\boldsymbol{H}$ (SLAMglmm), and every trait will vary among individuals due to a combination of environmental and genetic factors. Since in SLAMglmm, the higher-level traits $\boldsymbol{H} = [\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_p]$ are "latent" too, I really would like to treat all traits equivalently with respect to the modeled factors.

However, I am concerned about the structural consequences of this in the model, and identifiability. The issue is the following:

In a standard factor model, $\boldsymbol{f}_j \sim \mathrm{N}(0,1)$, with unit variance. This constraint is introduced because for any nonsingular matrix $\boldsymbol{\Sigma}$:

$$\begin{aligned}\mathbf{F}^* &:= \boldsymbol{\Sigma}^{1/2}\boldsymbol{F}, \boldsymbol{\Lambda}^* := \boldsymbol{\Lambda}\boldsymbol{\Sigma}^{-1/2} \\ \mathbf{F}\boldsymbol{\Lambda}^T &= \boldsymbol{F}^*\boldsymbol{\Lambda}^{*T} \\ \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T &= \boldsymbol{\Lambda}^*\boldsymbol{\Sigma}\boldsymbol{\Lambda}^{*T}\end{aligned}$$

so that modeled covariance is the same despite different $\boldsymbol{F}$ and $\boldsymbol{\Lambda}$ matrices. Therefore, these parameters are not identifiable.

In BSFG, I the replaced the above model for $\boldsymbol{F}$ with $\boldsymbol{f}_j \sim \mathrm{N}(0, h^2\boldsymbol{Z}\boldsymbol{A}\boldsymbol{Z}^T + (1-h^2)\boldsymbol{I})$. Assuming $\boldsymbol{Z}\boldsymbol{A}\boldsymbol{Z}^T$ is parameterized such that the diagonal is 1, this maintains the marginal variance of each $\boldsymbol{f}_{ij} = 1$. I then assessed the importance of each factor based on the magnitude of the elements λ_{ij}. The magnitude of the loadings of each factor is important because the infinite-factor model (Bhattacharya and Dunson) imposes shrinkage on higher-order columns of $\boldsymbol{\Lambda}$ through the stochastically increasing sequence of precision factors $\{\tau_h\} = \left\{ \prod_{j=0}^{h} \delta_j, \delta_j \sim \mathrm{Ga}(\alpha,\beta) \right\}$. If factor variances $\sigma^2(\boldsymbol{f}_j)$ were not fixed, then the sequence $\{\delta_j\}$ would not be identifiable.

In SLAMglmm, I do two things that may break this infinite-factor prior.

(1) I introduce the non-identified working parameter $\boldsymbol{\psi}^F = \mathrm{Diag}(\psi_j^F)$, a diagonal matrix of column-specific variances for $\boldsymbol{F}$. This is modeled after the factor model parameter expansion of (Ghosh and Dunson 2009). $\boldsymbol{\psi}^F$ allows $\sigma^2(\boldsymbol{f}_j)$ to vary during the sampling chain, reducing autocorrelation between λ_{ij} and f_{ij}. This factor is removed when samples of $\boldsymbol{F}, \boldsymbol{\Lambda}, \boldsymbol{B}_F$, and $\boldsymbol{U}_F$ are saved and the transformed parameters are identifiable. However, $\boldsymbol{\Psi}^F$ and δ_j interact during the sampling since one penalizes $\boldsymbol{F}$ and the other $\boldsymbol{\Lambda}$, and I am concerned that this may break the increasing column-shrinkage on $\boldsymbol{\Lambda}$. I think it's OK, and that the column-shrinkage sequence is now $\tau_j^* = \tau_j/\psi_j^F$, which simply has a more diffuse prior distribution, and as long as $\mathbb{E}[\delta_j/\psi_j^F] > 1$ the prior will induce increasing shrinkage on higher-order columns. My early experimentation with this parameter expansion seemed to show big improvements in the mixing of λ_{ij}. But I wonder if it's really needed given the similarity in how δ_j and ψ_j^F penalize the model, and haven't fully explored this issue with the re-formulated Gibbs sampler.

(2) I introduce fixed effects $\boldsymbol{X}_F\boldsymbol{B}_F$ into the model for $\boldsymbol{F}$. Now, the model for each column of $\boldsymbol{F}$ is $\boldsymbol{f}_j \sim \mathrm{N}(\boldsymbol{X}_F\boldsymbol{b}_{F_j}, h^2\boldsymbol{Z}\boldsymbol{A}\boldsymbol{Z}^T + (1-h^2)\boldsymbol{I})$. My concern is that if we compare models with and without these fixed effects:

$$\begin{aligned}\mathbf{f}_j^{(1)} &\sim \mathrm{N}(0, h^2\boldsymbol{Z}\boldsymbol{A}\boldsymbol{Z}^T + (1-h^2)\boldsymbol{I}) \\ \mathbf{f}_j^{(2)} &\sim \mathrm{N}(\boldsymbol{X}_F\boldsymbol{b}_{F_j}, h^2\boldsymbol{Z}\boldsymbol{A}\boldsymbol{Z}^T + (1-h^2)\boldsymbol{I}),\end{aligned}$$

the factor $\boldsymbol{X}_F\boldsymbol{b}_{F_j}$ should reduce the the residual variation (either from the genetic effects $h^2\boldsymbol{Z}\boldsymbol{A}\boldsymbol{Z}^T$ or the noises $(1-h^2)\boldsymbol{I}$) by explaining some of the variation in this latent trait. However, in the above parameterization, the noises are fixed to

have total variance equal to 1. Therefore $|\boldsymbol{f}^{(2)}| > |\boldsymbol{f}^{(1)}|$, and to compensate, the model will reduce the loadings of the jth column of $\boldsymbol{\Lambda}$ and choose a larger column-shrinkage parameter δ_j so that and the error variance can stay 1. Because the loadings are now smaller, we will conclude that this factor is less important. This is a problem because the factors with variation attributable to the fixed effects (ex. SNP genotype, environmental factor) will in some cases be the most interesting. A possible solution is to interpret factor importance based on $|\boldsymbol{f}^{(2)}| \times |\lambda_{ij}|$ rather than $|\lambda_{ij}|$, similarly to the re-scaling by $\boldsymbol{\Psi}^F$. However $\boldsymbol{\Psi}^F$ is a parameter of the parameter expanded model and is explicitly sampled, while $|\boldsymbol{f}^{(2)}|$ is not, so I'm not sure if this is valid.

5.1.1. *Additional Questions.*

(1) Is it necessary to center the columns of $\boldsymbol{X}_F$? The model for $\boldsymbol{f}_j$ is: $\boldsymbol{f}_j \sim \mathrm{N}(\boldsymbol{X}_F \boldsymbol{b}_{F_j}, \boldsymbol{\Sigma}_j)$. A standard factor model would have $\boldsymbol{f}_j \sim \mathrm{N}(0, \boldsymbol{I})$. Not centering the columns would allow the mean of $\boldsymbol{f}_j \neq 0$. But this could be absorbed by the global intercept which is not penalized, so not sure it matters. I don't think this will affect the estimation of $\boldsymbol{\Lambda}$.