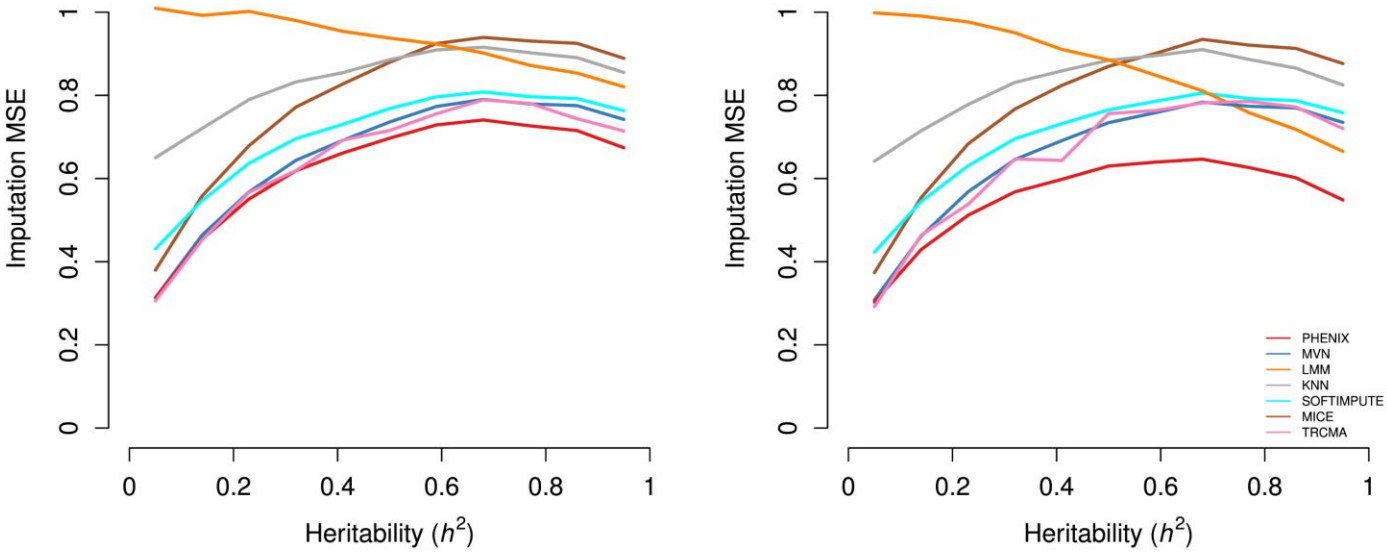
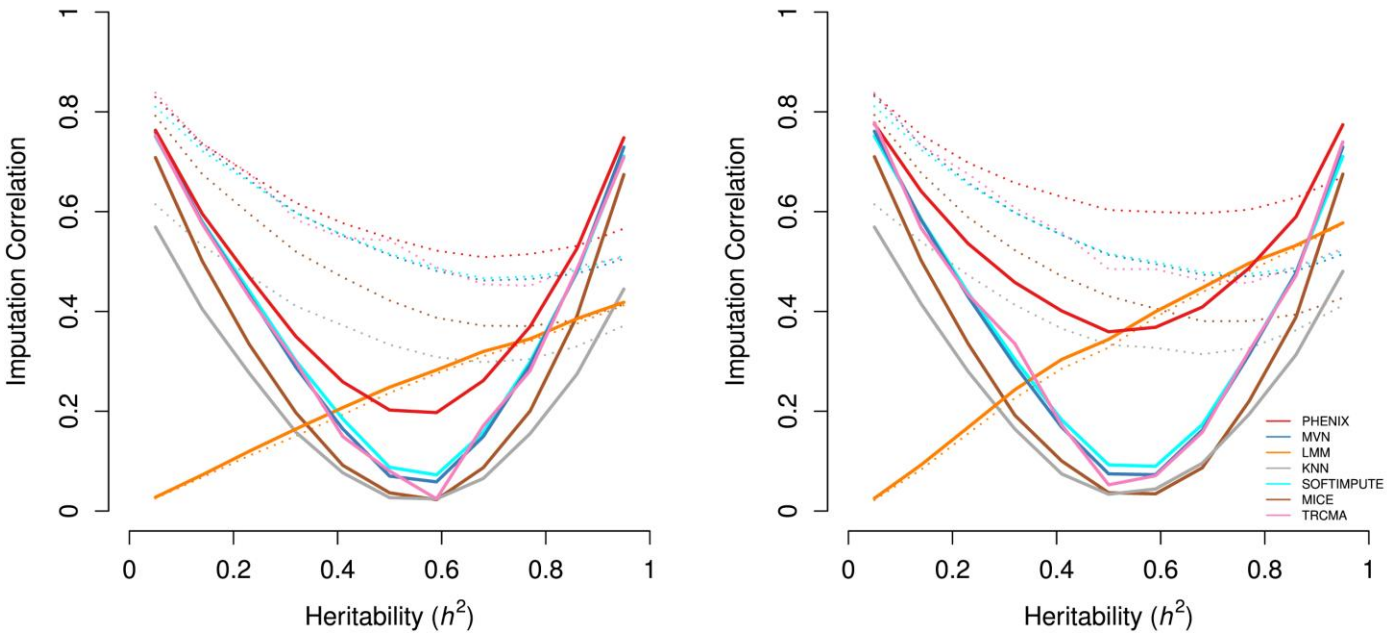


Supplementary Figure 1

Assessing phenotype imputation on simulated data using mean-squared error.

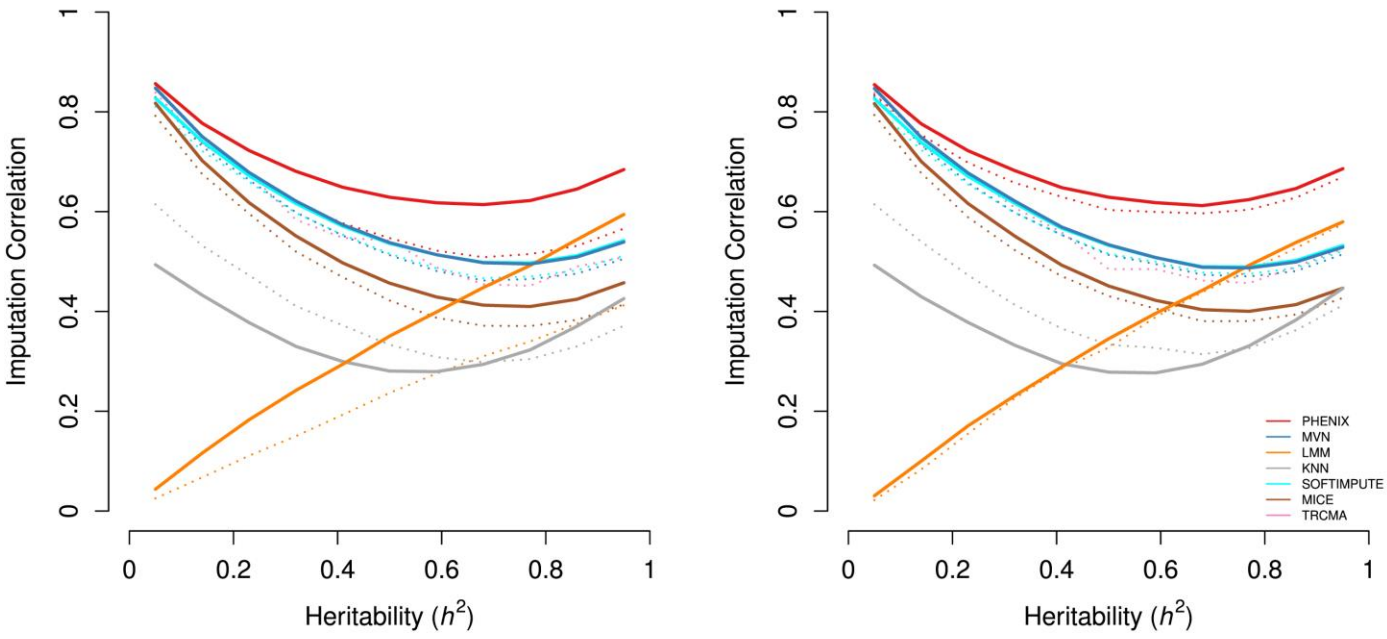
Simulation results measuring imputation accuracy with mean-squared error (MSE) rather than correlation. Model 1: the scenario simulated using an empirical kinship matrix derived from the human NSPHS study. Model 2: the scenario simulated using 75 unrelated families of four siblings. Data sets were simulated at various levels of heritability (*x* axis) for the traits. Three hundred individuals and 15 traits were simulated. Five percent of phenotype values were set to missing before imputation. Seven different methods (legend) were applied to impute the missing values. The MSE between the imputed values and the true values is plotted on the *y* axis for each method. Perfect imputation has MSE = 0, and, because phenotypes are centered and standardized, imputing all entries to 0 has MSE = 1. Compared to **Figure 1**, which uses correlation as an imputation metric, the results do not qualitatively change.



Supplementary Figure 2

Cancellation of genetic and environmental covariances.

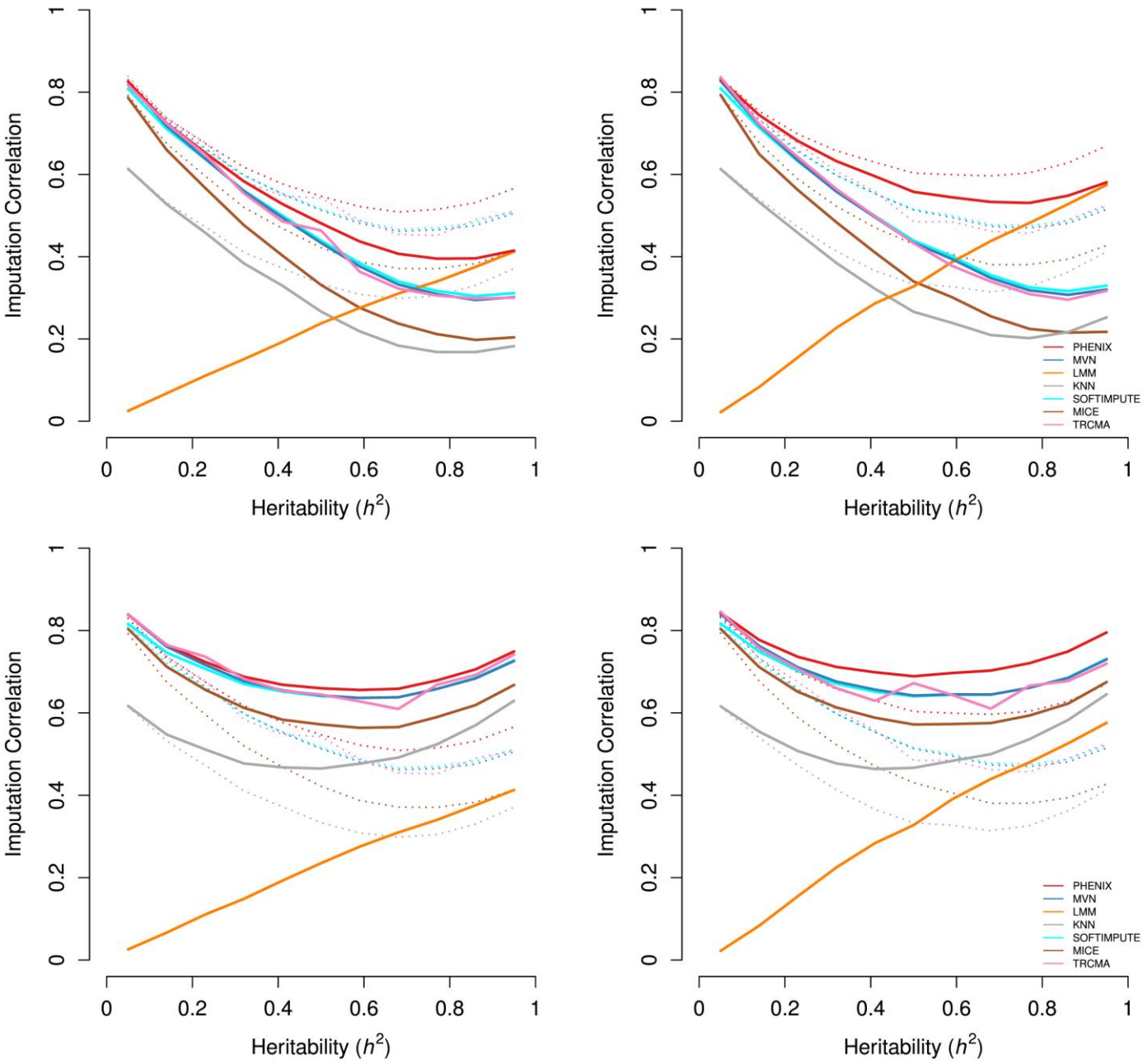
Simulation results with opposing genetic and environmental correlations. Rather than an AR matrix, this plot chooses genetic correlation B to cancel the environmental correlation, $B_{pq} = -E_{pq}$ for $p \neq q$. Five percent of phenotype values were held out, and the correlation between the true and imputed values is plotted on the y axis for each method. The dotted lines show the results from **Figure 1** for reference.



Supplementary Figure 3

Increasing sample size and number of phenotypes to *N* = 1,000 and *P* = 50.

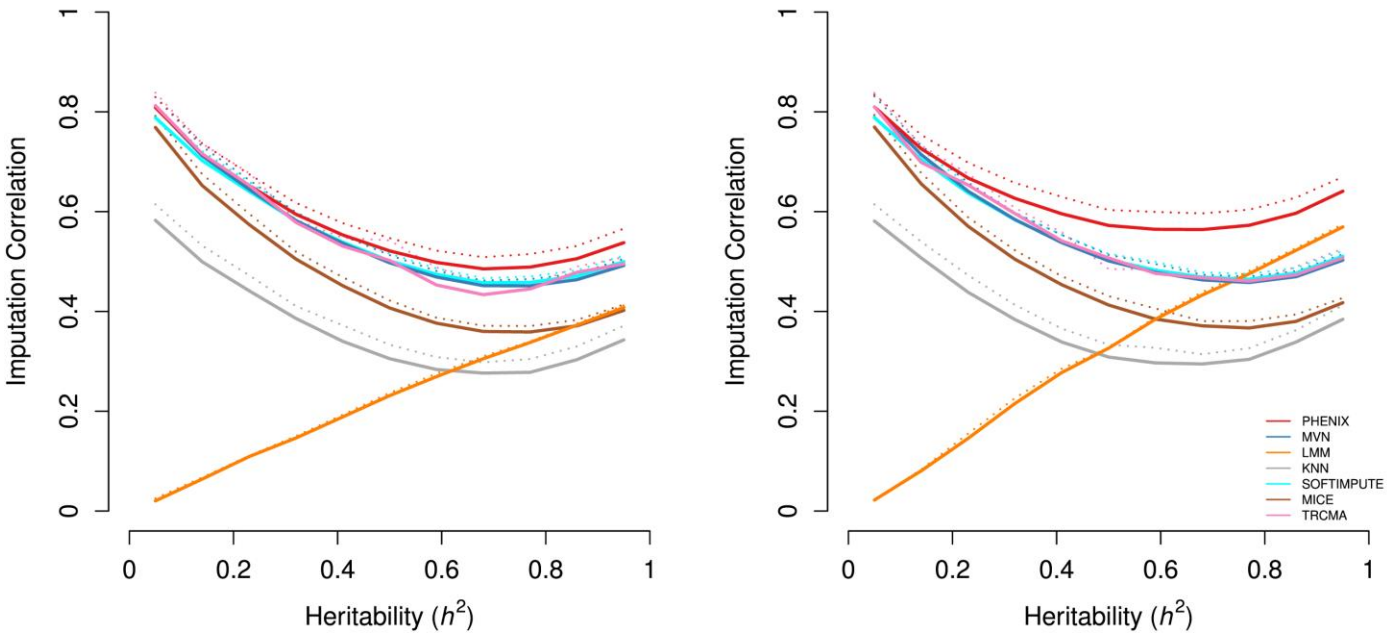
Simulation results using larger data sets. This figure uses (*N*, *P*) = (1,000, 50), while the dotted lines use (*N*, *P*) = (300, 15). Five percent of phenotype values were held out, and the correlation between the true and imputed values is plotted on the *y* axis for each method. Increasing the data size nearly always improves imputation accuracy, although this effect is attenuated when using the sibling relatedness matrix, as family sizes are fixed and increasing *N* does not increase the amount of between-sample correlation. The dotted lines show the results from **Figure 1** for reference.



Supplementary Figure 4

Varying levels of genetic correlation between phenotypes.

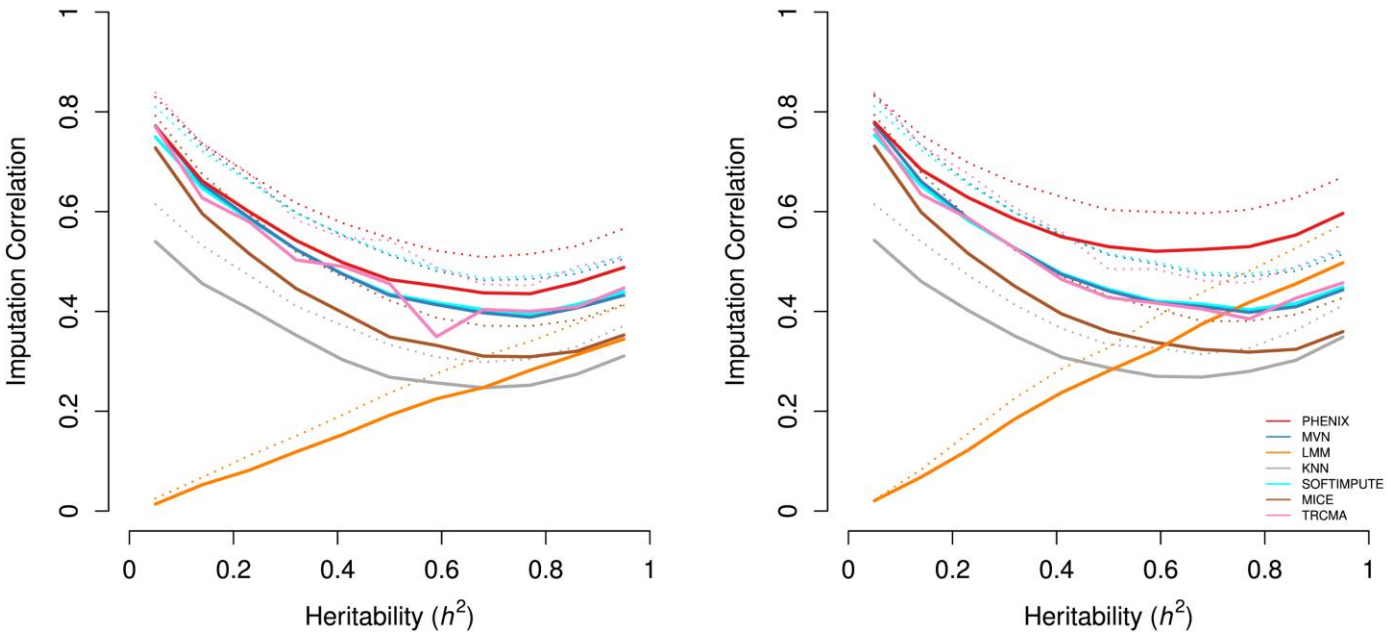
Simulation results varying the amount of genetic correlation. We vary the overall genetic correlation matrix B by changing ρ, the AR parameter. The top row shows simulations with ρ =.275, decreasing the average genetic correlation between traits compared to the dotted lines (from **Figure 1**) that use the baseline choice ρ =.45; the bottom row shows simulations with ρ = 0.675. Analogous results were obtained using ρ = –0.275 (data not shown). Five percent of phenotype values were held out, and the correlation between the true and imputed values is plotted on the y axis for each method. The imputation accuracy of multi-trait methods increases with genetic correlation, and this effect increases with h^2.



Supplementary Figure 5

Increasing data missingness to 10%.

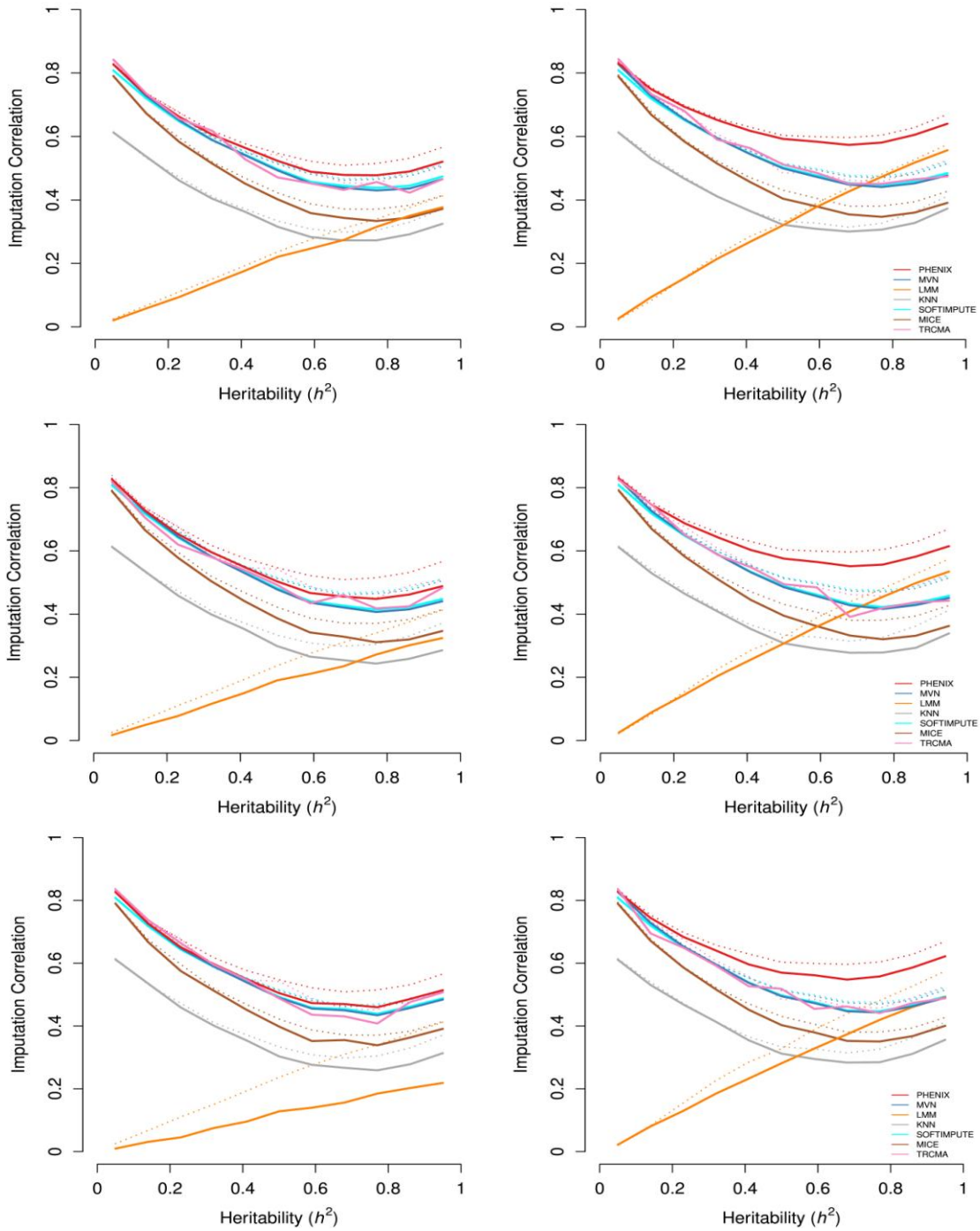
Simulation results at a higher level of missingness. Ten percent of phenotype values were set to missing before imputation, rather than 5% as for the dotted lines. The correlation between the imputed and true values is plotted on the *y* axis for each method. The dotted lines show the results from **Figure 1** for reference.



Supplementary Figure 6

Effect of non-random missingness.

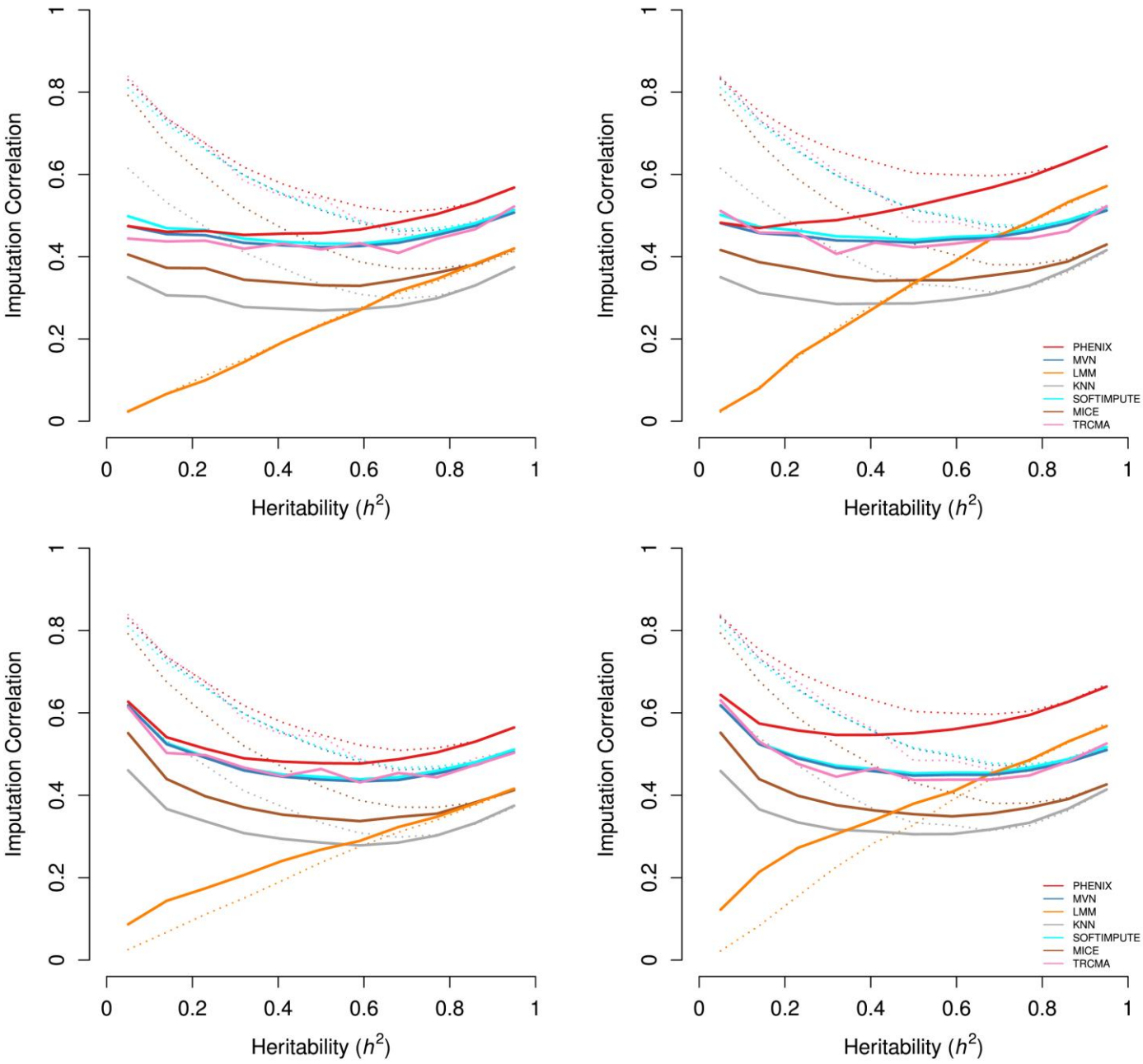
Simulation results with non-ignorable missingness. We hold out 5% of the entries of the phenotype matrix with probability increasing in their values, and the correlation between the true and imputed values is plotted on the *y* axis for each method. The dotted lines show the results from **Figure 1** for reference.



Supplementary Figure 7

Effect of unmodeled, shared environment.

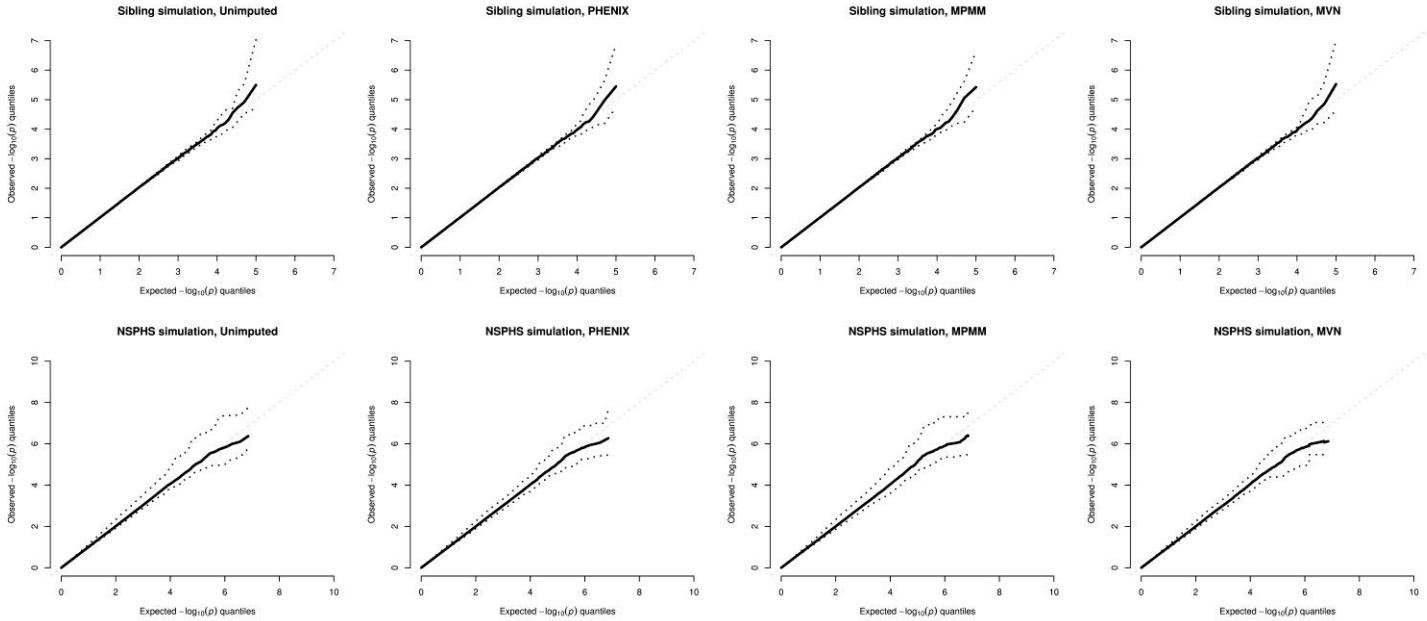
Simulation results with confounding cryptic relatedness. The contribution of the additive genetic term U in a typical MPMM is a^2; each row increases the contribution of the contaminating shared environment, c^2, to the overall heritability, here defined as $h^2 = a^2 + c^2$. The first row uses $c^2 = 0.1a^2$; the second uses $c^2 = 0.3a^2$; and the last uses $c^2 = a^2$. Five percent of phenotype values were held out, and the correlation between the true and imputed values is plotted on the y axis for each method. The dotted lines show the results from **Figure 1** for reference.



Supplementary Figure 8

Non-normally distributed phenotypes.

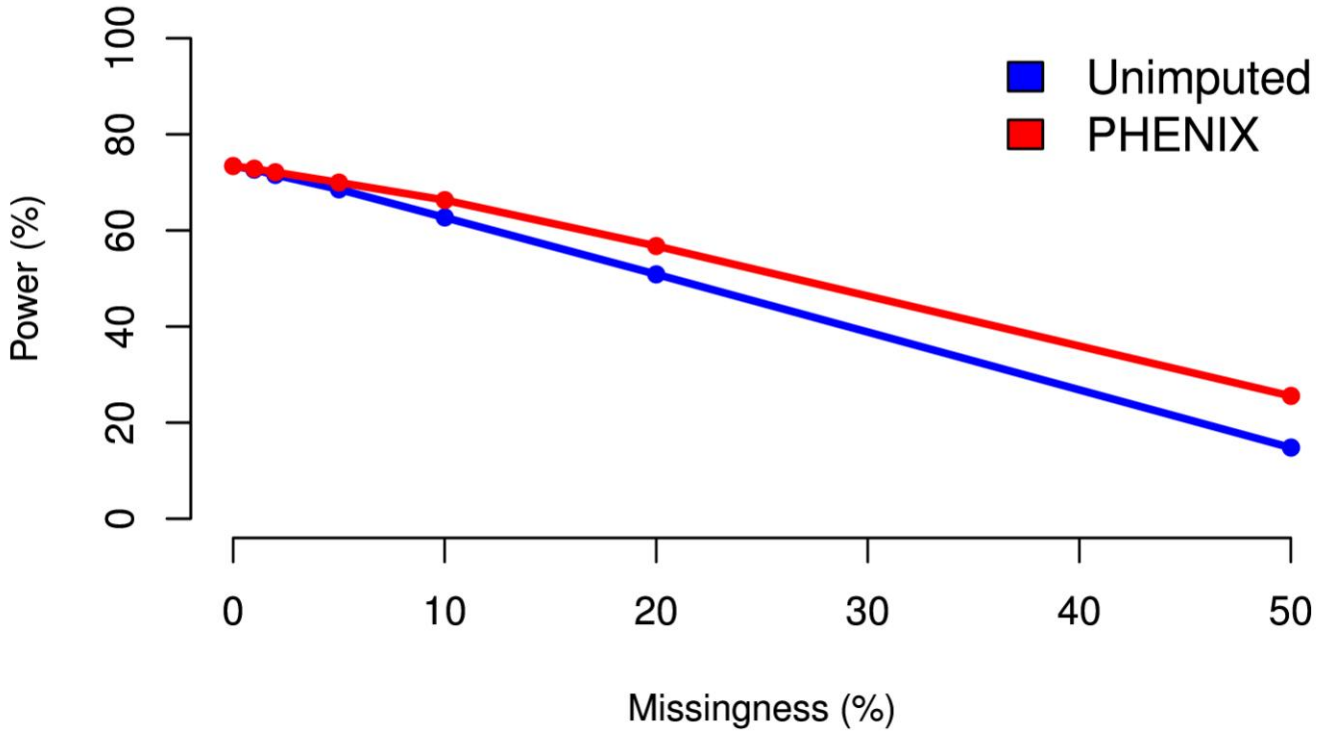
Simulation results with non-normal noise. We exponentially transform the environmental contribution ε to create log-normal noise. The resulting phenotypes are imputed without (top) or with (bottom) quantile normalization. Five percent of phenotype values were held out, and the correlation between the true and imputed values is plotted on the *y* axis for each method. The dotted lines show the results from **Figure 1** for reference.



Supplementary Figure 9

Type I error calibration after phenotype imputation.

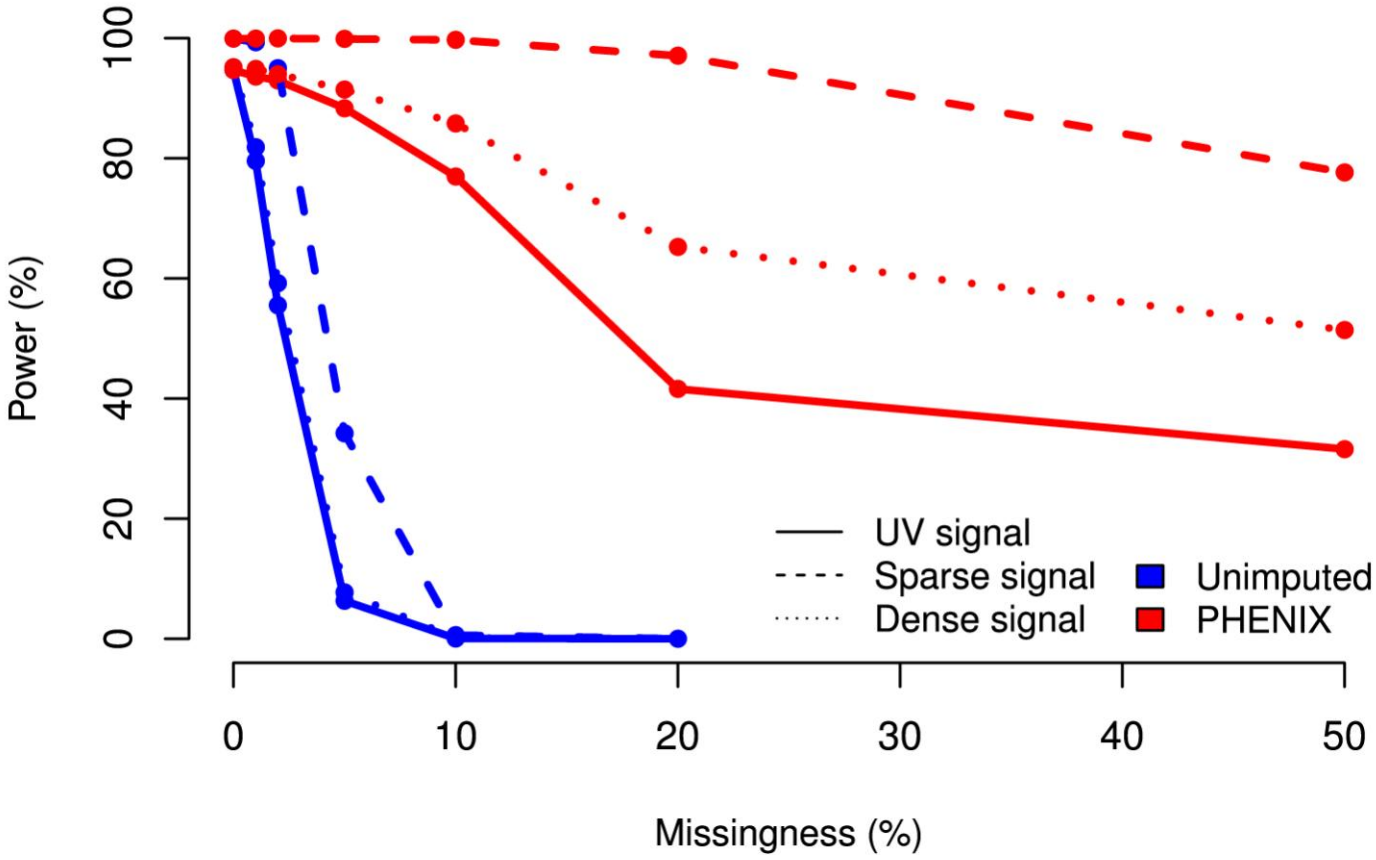
QQ plots from performing GWAS on 15 truly unassociated phenotypes with different imputation options (panel titles). Phenotypes are generated from our baseline simulation with the relevant *K* matrix. Rather than represent each of the 15 GWAS for each panel, we plot the point-wise minimum and maximum (dotted lines) and median (solid line) for the 15 lines. Top row, kinship and genotypes correspond to independent sets of four siblings. Bottom row, kinship and genotypes taken from the NSPHS study.



Supplementary Figure 10

Power of single-phenotype tests after phenotype imputation.

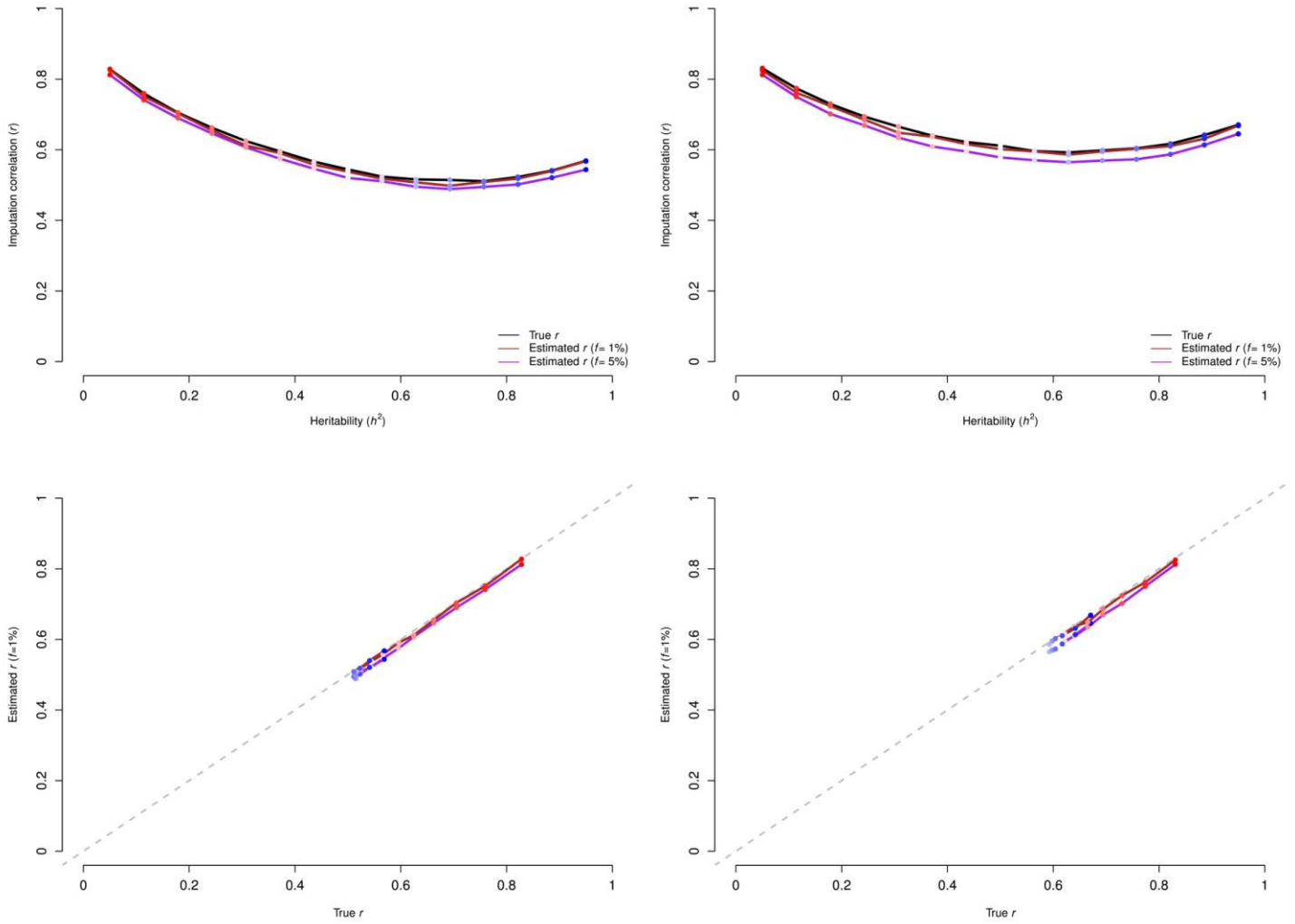
Power to detect a simulated, causal SNP using a univariate mixed model (LMM). Five thousand samples, comprising independent sets of four siblings, have 15 simulated phenotypes with pleiotropy. Five percent of phenotypes are deleted, and an LMM is then run with GEMMA after dropping missing data (unimputed) or imputing with PHENIX. Power is calculated by averaging over 1,000 independently simulated data sets using the standard GWAS *P*-value threshold 5×10^{-7}.



Supplementary Figure 11

Power of multiple-phenotype tests after phenotype imputation.

Power to detect a simulated, causal SNP using a multiple-phenotype mixed model (MPMM). Five thousand samples, comprising independent sets of four siblings, have 15 simulated phenotypes with three levels of pleiotropy (legend). Five percent of phenotypes are deleted, and an MPMM is then run with our method by dropping samples with any missing phenotype data (unimputed) or imputing with PHENIX. Power is calculated by averaging over 5,000 independently simulated datasets using the standard GWAS *P*-value threshold 5×10^{-7}.



Supplementary Figure 12

Calibration of the imputation metric *r*.

Calibration of our *r* metric for imputation accuracy. Data are from the baseline model, but we now record estimated imputation accuracies, which we call *r*, as well as the true imputation accuracies. Top row, imputation correlation is plotted against h^2. The black line is the true imputation accuracy and agrees with the PHENIX line (red) in **Figure 1**. We estimate *r* in two ways: by hiding 1% (brown line) or 5% (purple line) of observed entries. Point colors correspond to values of h^2. Bottom row, estimated *r* is compared to the true *r*, with variability created by varying h^2. Each point corresponds to the point in the above plot with the same color.

Multiple phenotype imputation for genetic studies

Andrew Dahl, Valentina Iotchkova, Amelie Baud, Asa Johansson,
Ulf Gyllensten, Nicole Soranzo, Richard Mott, Andreas Kranis,
Jonathan Marchini

January 24, 2016

Contents

1 The PHENIX model

1.1 Definitions and notation

The Kronecker product of matrices is denoted by $\otimes$ and the Kronecker sum, $\oplus$, is defined

$$A \oplus B := A \otimes I + I \otimes B$$

For a matrix X, we let the lower case x refer to the column-wise vectorization of X, written $x = \text{vec}(X)$; similarly, we let $\text{mat}(x) = X$ be the 'inverse' operation (the dimensions being implicitly defined by context). If M is an $NP \times NP$ matrix, we can represent it in terms of $N \times N$ blocks:

$$M = \begin{bmatrix} M_{11} & \dots & M_{1P} \\ \vdots & \ddots & \vdots \\ M_{P1} & \dots & M_{PP} \end{bmatrix}$$

Then the partial trace $tr_P(M)$ is the $P \times P$ matrix of traces of such blocks

$$tr_P(M) = \begin{bmatrix} tr(M_{11}) & \dots & tr(M_{1P}) \\ \vdots & \ddots & \vdots \\ tr(M_{P1}) & \dots & tr(M_{PP}) \end{bmatrix}$$

We write the matrix variate normal with mean M, row covariance R and column covariance C as

$$\mathcal{MN}(M, R, C)$$

This is a special case of a multivariate normal as the vectorization of this matrix has mean $\text{vec}(M)$ and covariance $C \otimes R$.

1.2 Model description

Let $Y \in \mathbb{R}^{N \times P}$ be a partially observed matrix of P phenotypes measured on N individuals. We assume that the columns of Y have been demeaned and standardized to unit variance. We start with the additive model

$$Y = U + \epsilon \tag{1}$$

where U represents the aggregate genetic contribution to phenotypic variance and ϵ is idiosyncratic noise. One model we consider uses independent matrix-variate normal distributions for U and ϵ:

$$\begin{array}{rcl} Y & = & U + \epsilon \\ U & \sim & \mathcal{MN}(0, K, B) \\ \epsilon & \sim & \mathcal{MN}(0, I, E) \end{array} \tag{2}$$

K is the kinship matrix between individuals in the sample, which we assume is known from pedigree or genotype data [23, 8, 13, 31, 30, 29]. This model has recently attracted attention in genetics [33, 10, 3, 24] and we refer to it as a multiphenotype mixed model (MPMM).

MPMMs arise as a multiphenotype generalization of the typical univariate linear mixed model (LMM): when B and E are diagonal in (2), the MPMMs reduce to P independent LMMs of the form

$$\begin{array}{rcl} Y_{,p} & = & u_p + \epsilon_p \\ u_p & \overset{\text{ind}}{\sim} & \mathcal{N}(0, B_{pp}K) \\ \epsilon_p & \overset{\text{ind}}{\sim} & \mathcal{N}(0, E_{pp}I) \end{array} \tag{3}$$

Unfortunately, MPMMs can handle only a small number phenotypes, roughly 10 [33]–as P grows, maximum likelihood covariance estimates quickly become both statistically unstable and computationally intractable. Moreover, missing observations are hard to incorporate into MPMMs as the vector of observed phenotypes inherits the matrix normal structure of the full data only if entire rows are missing (see section 2.7). Removing samples with even one missing phenotype [33] thus eliminates the computational aspect of this missing data hurdle, but at the cost of throwing away data; if entries are missing uniformly at random with probability θ, a sample is fully observed with probability $(1-\theta)^P$ and the data waste is exponential in P.

To simultaneously address both of these limitations, we develop an alternative multiphenotype generalization of LMMs[1] by assuming an entirely different model for the genetic term U. In particular, we use a Bayesian low-rank matrix factorization model for the genetic term U. Such low rank models are computationally tractable and, additionally, we believe this rank constraint is often biologically plausible: U will have (approximately) low-rank M when the P observed phenotypes share a simple biological structure that is (mostly) summarized by M latent factors.

Specifically, for $M \leq N, P$, we use the model

$$\begin{array}{rcl} Y|S, \beta, \epsilon & \sim & U + \epsilon \\ U & = & S\beta \\ S & \sim & \mathcal{MN}(0, K, I_M) \\ \beta & \sim & \mathcal{MN}(0, C, B), \\ \epsilon|\Lambda_\epsilon & \sim & \mathcal{MN}\left(0, I, \Lambda_\epsilon^{-1}\right) \\ \Lambda_\epsilon & \sim & \text{Wishart}(e, E) \end{array} \tag{4}$$

If C is allowed to be an arbitrary diagonal matrix[2], then the matrix factorization model in (4) is equivalent to reduced-rank regression in the same sense that MPMM and LMM are equivalent to genome-wide linear regression. For simplicity, we set $C = I_M$, $B = (\tau I_P)^{-1}$, $e = P + 5$ and $E = e^{-1}I_P$ (so that $\mathbb{E}(\Lambda_\epsilon) = I_P$). Though τ can be tuned by cross-validation, we use the improper $\tau = 0$ by default (see section 1.3.2).

We note that many fast, powerful and robust penalized likelihood methods exist for estimating a spectrally-regularized U in (1), including many focused on imputing missing entries [21, 2, 16, 18]. However, we know of no method that incorporates, or can be easily generalized to incorporate, a non-spherical kinship matrix K. But K is the central element of LMMs in genetics (and random effect models generally). Moreover, by comparing to a competitive spectral-regularization algorithm

[1] It actually generalizes a slightly different, Bayesian version of the LMM in (3), where B_{pp} has a scaled χ^2 prior and E_{pp} has an inverse-gamma prior.

[2] Due to scaling and rotation non-identifiablity, C can be assumed diagonal without loss of generality; see, for example, [18].

from the literatue on generic matrix completion [16] (see section 2.6), our simulations and real data analyses suggest incorporating K is always beneficial, and sometimes vital, for imputation accuracy when there is genetic signal.

1.3 Variational Bayesian matrix factorization

We use variational Bayes (VB) to approximate the posterior in model (4). In matrix factorization models, VB is an established alternative to MCMC (which can be computationally expensive) and maximum *a posteriori* [22, 7, 12] (which can suffer from over-fitting). Moreover, VB matrix factorization has known theoretical properties in special cases [18] (see section 1.3.1). Our implementation iteratively updates approximate posteriors on S, β, Λ_ϵ and Y^m, the missing entries of Y, assuming that these parameters are independent in the posterior. Though this independence assumption does not hold and is potentially problematic [22], it simplifies computation while hopefully retaining much of the exact problem's structure.

Specifically, we require Q, the variational approximation to the posterior, to factorize over the partition $\{S, \beta, \Lambda_\epsilon, Y^m\}$ of the parameter space:

$$Q(Y^m, S, \beta, \Lambda_\epsilon | Y \setminus Y^m) := Q_Y(Y^m) Q_S(S) Q_\beta(\beta) Q_\epsilon(\Lambda_\epsilon)$$

The goal is then to find Q's that best approximate the posterior (in Kullback-Leibler divergence). Defining m_i as the missing phenotypes for sample i, section 1.4 shows that the Q's belong to simple parametric families:

$$\begin{aligned} Q(Y_{i,m_i}) &\sim \mathcal{N}\left(\mu^{Y_i}, \Sigma^{Y_i}\right) \\ Q(\mathrm{vec}\,(S)) &\sim \mathcal{N}\left(\mu_s, \Lambda_s^{-1}\right) \\ Q(\mathrm{vec}\,(\beta)) &\sim \mathcal{N}\left(\mu_b, \Lambda_b^{-1}\right) \\ Q(\Lambda_\epsilon) &\sim \mathrm{W}\left(e', \frac{1}{e'}\Omega\right) \end{aligned}$$

The problem of optimizing the Q's thus reduces to finding optimal variational parameters for the above approximate marginals.

This minimization is performed by iterating through conditional modes, optimizing each approximate marginal given the others (see Section 1.4.1). Because the conditional optimizers have analytic expressions, this hill-climbing is fast. Unfortunately, this coordinate ascent need not reach a global optimum as our variational objective is non-convex (in addition to the rotation ambiguity in the product $S\beta$, which is inconsequential since we never jointly update S and β) [7]. Nonetheless, we have not found this problematic in our setting: maybe this is because we initialize at full rank $S\beta$ and allow the fitted rank to converge from above (see 1.3.1 and 1.3.2); maybe it is because we initialize with another method (MVN); maybe it is because we update all of S or β at once, avoiding the typical practice of conditionally updating each component given the others.

As written, the approximate marginals for S and β depend on very large precision matrices–Λ_s and Λ_b–that induce $O(M^3(N^3 + P^3))$ computations. Though these matrices are not Kronecker products–and so S and β are not matrix normal, even in our variational approximation to the posterior–they do have a simple structure that admits much faster computations. If N_m is the number of unique missingness patterns among samples, our algorithm costs $O(N_m P^3 + NP^2 + N^2 M)$ for each VB iteration; additionally, we perform a one-off, full-rank eigendecomposition of K at $O(N^3)$.

1.3.1 Properties of a special case

The globally optimal VB matrix factorization parameters have analytic expressions when Y is fully observed and covariances are spherical ($\Lambda_\epsilon = I_P$ and $K = I_N$) [18]. As those authors note, these equations do not easily generalize either to missing data or to non-spherical priors, and this result is not directly useful for us.

Nonetheless, these analytic solutions reveal a surprising property of VB matrix factorization: $\hat{U}$, the expected U under the approximate posterior, may have rank strictly less than M, the *a priori* maximum rank of U and the almost-sure rank of U under both the prior and the (exact) posterior. This is because the singular values of $\hat{U}$ are, roughly, the soft-threshholded singular values of Y[3]. As τ controls the magnitude of this soft-threshholing, the search over τ can replace the search over M, much as (convex) lasso relaxes (non-convex) subset search for regression. In fact, reasonable conditions guarantee that optimizing τ is enough to recover the correct rank of U [19].

Though these automatic rank selection properties have not been proven in our context, we assume that analogues apply as we have consistently observed that our model fits low-rank $\hat{U}$. Specifically, we assume that the automatic rank determination is reliable, so we always set $M = \min(N, P)$–a computational impossibility for truly large P–and allow the algorithm to decide the rank of the putatively low-rank component through τ.

1.3.2 Choosing the regularization parameter τ

Surprisingly, even when $\tau = 0$ and the prior on β is flat, the implied prior on the product $U = S\beta$ is non-flat and shrinks the singular values of U to zero (see section 4). Nonetheless, increasing τ increases regularization, motivating $\tau = 0$ as a widely applicable default, as this value is optimal for all datasets where even this minimal amount of shrinkage is too much; for example, cross-validation chose $\tau = 0$ of its own accord in the NSPHS data set. In all analyses in the paper we have only used $\tau = 0$.

1.4 Details of the PHENIX algorithm

1.4.1 Variational Bayes overview

VB aims to approximate a complicated posterior distribution $P(\theta|D)$, where D is the data and $\theta \in \Theta$ are the model parameters, by a function $Q(\theta)$ chosen from a class of simple functions, $\mathcal{Q}$. Once found, exact properties of the approximate posterior, Q, can be used to approximate properties of the exact posterior, $P(\cdot|D)$, such as parameter means and covariances and marginal likelihoods.

For any approximate posterior Q, the true log marginal likelihood can be written as

$$\log P(D) \quad = \quad F(Q) + D_{KL}(Q||P(\cdot|D)) \tag{5}$$

where D_{KL} is the Kullback-Liebler divergence and $F(Q) = \int \log \left[\frac{P(\theta,D)}{Q}\right] dQ(\theta)$. We choose $Q \in \mathcal{Q}$ to minimize D_{KL} which, since the marginal likelihood $P(D)$ does not depend on Q, is equivalent to maximizing $F(Q)$. Moreover, since D_{KL} is non-negative, $F(Q)$ lower-bounds, and approximates, the log marginal likelihood.

[3]This is made formal in [18]; see also [9], which relates the variational Bayesian matrix factorization objective to nuclear norm regularization and, thus, to the matrix completion methods in [16, 2, 21]

Mean field approximations are one way to specify $\mathcal{Q}$, which require that each $Q \in \mathcal{Q}$ factorizes over some partition of Θ:

$$Q \in \mathcal{Q} \iff Q(\theta) = \prod_i Q_i(\theta_i) \quad \forall \theta \in \Theta$$

With this mean field assumption, it is natural to iteratively optimize one coordinate of Q given the others:

$$Q_i \leftarrow \arg\max_{Q_i'} F(Q_i', Q_{-i}) \tag{6}$$

Since we are minimizing D_{KL}, these updates take a particularly simple form:

$$\log Q_i \leftarrow \arg\max_{Q_i} F(Q_i, Q_{-i}) \equiv \mathbb{E}_{\theta_{-i} \sim Q_{-i}} \Big(\log P(D, \theta) \Big) \tag{7}$$

The precise form of each Q_i will depend on the likelihood and priors, and one key feature is that the Q_i are not chosen in advance but rather chosen to minimize Kullback-Leibler divergence from the posterior. Nonetheless, the usefulness of VB typically relies on each Q_i reducing to a tractable parametric form, which we index by variational parameters $\tilde{\theta}_i$. With this simplification, the coordinate ascent problem (6), which in general optimizes Q_i over a function space, reduces to optimizing $\tilde{\theta}_i$.

Since we require Q to factorize over the parameter partition $\{S, \beta, Y^m, \Lambda_\epsilon\}$, our mean field algorithm iteratively updates Q_S, Q_β, Q_ϵ and Q_Y. Below, we use (7) to derive these updates.

1.4.2 The parametric forms of the approximate posterior marginals

$\boldsymbol{Y}$: $Q_{Y_{i,m_i}} \overset{\text{ind}}{\sim} \mathcal{N}\left(\mu^Y_{i,m_i}, \Sigma^{Y_i}\right)$

$$\begin{aligned}
-2 \log Q_{Y^m} &\equiv -2\mathbb{E}_{-Y^m} \left(\log P(Y | S, \beta, \Lambda_\epsilon) \right) \\
&\equiv \mathbb{E}_{-Y^m} \left(\operatorname{tr} \left((Y - S\beta)\Lambda_\epsilon (Y - S\beta)^T \right) \right) \\
&\equiv \operatorname{tr} \left((Y - \mathbb{E}(S\beta)) \mathbb{E}(\Lambda_\epsilon) (Y - \mathbb{E}(S\beta))^T \right) \implies \\
Q_{Y_i} &\overset{\text{ind}}{\sim} \mathcal{N}\left((\mu_S \mu_\beta)_{i,}, \Omega^{-1} \right)
\end{aligned}$$

where μ_S, μ_β and Ω are moments of the other marginals and defined by their respective updates (see below). The distribution of $Y^m | Y^o$ follows from this unconditional distribution:

$$\begin{aligned}
Y_{i,m_i} | Y_{i,o_i} &\overset{\text{ind}}{\sim} \mathcal{N}\left(\mu^Y_{i,m_i}, \Sigma^{Y_i}\right) \\
\mu^Y_{i,m_i} &= (\mu_S \mu_\beta)_{i,m_i} + (\Omega^{-1})_{m_i,o_i} \left(\Omega^{-1}\right)_{o_i,o_i} \left(Y_{i,o_i} - (\mu_s \mu_\beta)_{i,o_i}\right)^T \qquad (8) \\
\Sigma^{Y_i} &= (\Omega_{m_i,m_i})^{-1} \qquad (9)
\end{aligned}$$

Updating μ^Y_{i,m_i} and Σ^{Y_i} for each i costs $O(NP^3)$. But, since the $O(P^3)$ operations for each i depend on i only through o_i, the complexity can be reduced to $O(NP^2 + N_m P^3)$, where N_m is the number of unique trait missingness patterns among the N samples. In real datasets, where experimental and observational constraints often induce highly structured missingness patterns, N_m is often much smaller than N: for example, in the chicken data, $N = 11,575$ but $N_m = 36$.

$\boldsymbol{\beta}: Q_{\mathbf{vec}(\beta)} \sim \mathcal{N}\left(\mu_b, \Lambda_b^{-1}\right)$

$$\begin{aligned}
-2\log Q_\beta &\equiv -2\mathbb{E}_{-\beta}\left(\log P(Y|S,\beta,D,\Lambda_\epsilon) + \log P(\beta)\right) \\
&\equiv \mathbb{E}_{-\beta}\left(||(Y - S\beta)\Lambda_\epsilon^{1/2}||_F^2 + \tau||\beta||_F^2\right) \\
&\equiv \mathrm{tr}\left(\beta\mathbb{E}\left(\Lambda_\epsilon\right)\beta^T\mathbb{E}\left(S^TS\right)\right) - 2\mathrm{tr}\left(\beta\mathbb{E}\left(\Lambda_\epsilon Y^TS\right)\right) + \tau\,\mathrm{tr}\left(\beta\beta^T\right) \\
&\equiv \mathrm{vec}\left(\beta\right)^T\left[\mathbb{E}\left(\Lambda_\epsilon\right)\otimes\mathbb{E}\left(S^TS\right) + \tau I\right]\mathrm{vec}\left(\beta\right) - 2\mathrm{vec}\left(\beta\right)^T\mathrm{vec}\left(\mathbb{E}\left(S^TY\Lambda_\epsilon\right)\right) \implies \\
\mathrm{vec}\left(\beta\right) &\sim \mathcal{N}\left(\mu_b, \Lambda_b^{-1}\right)
\end{aligned}$$

giving the updates

$$\Lambda_b = \Omega_\beta \otimes V_S + \tau I \qquad \text{(implicit)}$$
$$\mu_b = \Lambda_b^{-1}\mathrm{vec}\left(\mu_S^T\mu_Y\Omega_\beta\right) \qquad (10)$$
$$\Omega_\beta = \Omega \qquad (11)$$
$$V_S = \mu_S^T\mu_S + \mathrm{tr}_P\left(\Lambda_s^{-1}\right) \qquad (12)$$

Using lemma 2, (10) can be computed in $O(P^3 + MNP)$ rather than $O(M^3P^3 + MNP)$. Similarly, using lemma 1, (12) can be found in $O(M^3 + NM^2)$ rather than $O(N^3M^3)$. In both cases, explicitly forming Λ_b is unnecessary; because Λ_b is a function of a specific Ω, not whatever Ω has become since last updating Q_β, we perform (11) so we can at all times evaluate terms involving Λ_b.

$\boldsymbol{S}: Q_{\mathbf{vec}(S)} \sim \mathcal{N}\left(\mu_s, \Lambda_s^{-1}\right)$

$$\begin{aligned}
-2\log Q_{-S} &\equiv -2\mathbb{E}_{-S}\left(\log P(Y|S,\beta,D,\Lambda_\epsilon) + \log P(S)\right) \\
&\equiv \mathbb{E}_{-S}\left(||(Y - S\beta)\Lambda_\epsilon^{1/2}||_F^2 + ||K^{-1/2}S||_F^2\right) \\
&\equiv \mathrm{tr}\left(S\mathbb{E}\left(\beta\Lambda_\epsilon\beta^T\right)S^T\right) - 2\mathrm{tr}\left(S\mathbb{E}\left(\beta\Lambda_\epsilon Y^T\right)\right) + \mathrm{tr}\left(S^TK^{-1}S\right) \\
&\equiv \mathrm{vec}\left(S\right)^T\left(\mathbb{E}\left(\beta\Lambda_\epsilon\beta^T\right)\otimes I + I\otimes K^{-1}\right)\mathrm{vec}\left(S\right) - 2\mathrm{vec}\left(S\right)^T\mathrm{vec}\left(\mathbb{E}\left(Y\Lambda_\epsilon\beta^T\right)\right) \implies \\
\mathrm{vec}\left(S\right) &\sim \mathcal{N}\left(\mu_s, \Lambda_s^{-1}\right)
\end{aligned}$$

where

$$\Lambda_s = V_\beta \oplus K^{-1} \qquad \text{(implicit)}$$
$$\mu_s = \Lambda_s^{-1}\mathrm{vec}\left(\mu_Y\Omega\mu_\beta^T\right) \qquad (13)$$
$$V_\beta = \mu_\beta\Omega\mu_\beta^T + \mathrm{tr}_P\left(\left(\Omega\otimes I\right)\Lambda_b^{-1}\right) \qquad (14)$$

Since only explicitly evaluated parameters depend on Ω, there is no need to store a copy.

Unfortunately, $\mathrm{tr}_P\left(\left(\Omega\otimes I\right)\Lambda_b^{-1}\right)$ does not generally simplify as $\Omega \neq \Omega_\beta$ in general. However, I ensure Q_β was updated more recently than Q_ϵ when updating Q_S, and so $\Omega = \Omega_\beta$ and

$$\mathrm{tr}_P\left(\left(\Omega\otimes I\right)\Lambda_b^{-1}\right) = \mathrm{tr}_P\left(\left[\left(\tau\Omega^{-1}\right)\otimes V_S\right]^{-1}\right)$$

With this simplification, lemma 2 computes (13) in $O(N^2M + P^2M)$ instead of $O(N^3M^3 + P^2M)$, lemma 1 computes (14) in $O(P^3)$ rather than $O(M^3P^3)$ and Λ_s need not be evaluated.

Equation (13) is the reason our method has $O(N^2M)$ iterations while most mixed models only have one $O(N^2P)$ step: typical mixed models assume Y is complete and so the problematic step, whitening Y (or, in our case, μ_Y), only needs to be performed once[4].

Equation (13) is also where low-rank kinship models pay off: if $\text{rk}(K) = R$, the cost of this step becomes $O(NRM + P^2M)$ and the overall complexity drops from $O(N_mP^3 + NP^2 + N^2M)$ to $O(N_mP^3 + NP^2 + NRM)$. Though this change will be crucial for small P, huge N–where N is, say, tens or hundreds of thousands and P is, say, tens–it is unlikely to matter much in our currently studied applications; a similar logic applies to the one-off, low-rank eigendecomposition of K, which can be sped up to $O(RN^2)$.

$\mathbf{\Lambda_\epsilon} : Q_\epsilon \sim \mathbf{W}\left(e', \frac{1}{e'}\Omega\right)$

Define $\tilde{\Sigma}^{Y_i} \in \mathbb{R}^{P\times P}$ by padding $\Sigma^{Y_i} \in \mathbb{R}^{m_i\times m_i}$ with 0s in the natural way. Then

$$\begin{aligned}
\Omega_0 &:= \mathbb{E}\left((Y - S\beta)^T(Y - S\beta)\right) \\
&= \mathbb{E}\left((Y - \mathbb{E}\left(S\beta\right))^T(Y - \mathbb{E}\left(S\beta\right))\right) + \mathbb{E}\left(\left(S\beta - \mathbb{E}\left(S\beta\right)\right)^T\left(S\beta - \mathbb{E}\left(S\beta\right)\right)\right) \\
&= (\mu_Y - \mu_S\mu_\beta)^T(\mu_Y - \mu_S\mu_\beta) + \sum_n \tilde{\Sigma}^{Y_n} \qquad (16) \\
&\quad + \mu_\beta^T \text{tr}_P\left(\Lambda_s^{-1}\right)\mu_\beta + \text{tr}_P\left((I \otimes \left[\mu_S^T\mu_S + \text{tr}_P\left(\Lambda_s^{-1}\right)\right])\Lambda_b^{-1}\right) \qquad (17)
\end{aligned}$$

If Q_β has been updated more recently than Q_S, $V_S = \mu_S^T\mu_S + \text{tr}_P\left(\Lambda_s^{-1}\right)$ and then

$$\text{tr}_P\left((I \otimes \left[\text{tr}_P\left(\Lambda_s^{-1}\right) + \mu_S^T\mu_S\right])\Lambda_b^{-1}\right) = \text{tr}_P\left(\left[\Omega_\beta \oplus (\tau V_S^{-1})\right]^{-1}\right)$$

Now the $\text{tr}_P(\cdot)$ terms are inverse Kronecker sums and so, by lemma 1, (17) costs $O(P^3 + NM)$ to evalate; (16) costs $O(NP^2)$ as written.

[4]We could save some computation by storing a whitened version of the observed parts of Y. Let $Y^0_{ij} = Y_{ij}$ if observed, $Y^0_{ij} = 0$ otherwise. Then store

$$Y' = Q^TY^0$$

where Q are the eigenvectors of K. Then at each iteration, $Q^T\mu_Y$ can be computed by

$$Q^T\mu_Y = Y' + Q^TY^1 \qquad (15)$$

where $Y^1_{ij} = 0$ if Y_{ij} is observed and $Y^1_{ij} = \mu^Y_{ij}$ otherwise. Since Y^1 has only n_{miss} nonzero entries, the multiplication in (15) is $O(Nn_{miss})$, which may be substantially cheaper than $O(N^2M)$ in some applications. Nonetheless, n_{miss} will almost always be $O(NP)$ and so the $O(Nn_{miss})$ cost is only superficially linear in N; in fact, this cost may be greater than $O(N^2M)$ when $M << P$.

Letting $e' = e + N$, it then follows that

$$\begin{aligned}
\log Q_\epsilon(\Lambda_\epsilon) &\equiv \mathbb{E}_{-\Lambda_\epsilon}\left(\log P(Y|S,\beta,\Lambda_\epsilon) + \log P(\Lambda_\epsilon)\right) \\
&\equiv \mathbb{E}_{-\Lambda_\epsilon}\left(-\frac{1}{2}\mathrm{tr}\left((Y-S\beta)\Lambda_\epsilon(Y-S\beta)^T\right) + \frac{N}{2}\log|\Lambda_\epsilon|\right) + \left(\frac{e-P-1}{2}\log|\Lambda_\epsilon| - \frac{1}{2}\mathrm{tr}\left(E^{-1}\Lambda_\epsilon\right)\right) \\
&\equiv -\frac{1}{2}\mathrm{tr}\left(\Lambda_\epsilon\left(\mathbb{E}\left((Y-S\beta)^T(Y-S\beta)\right) + E^{-1}\right)\right) + \frac{N+e-P-1}{2}\log|\Lambda_\epsilon| \implies \\
Q_\epsilon &\sim \mathrm{Wi}\left(e', \frac{1}{e'}\Omega\right)
\end{aligned}$$

where

$$\Omega := e'\left(\Omega_0 + E^{-1}\right)^{-1}$$

1.4.3 The marginal likelihood lower bound

We assess convergence by monitoring relative change in the marginal likelihood lower bound ($F(Q)$ in (5)); by default, we terminate once either 1,000 iterations have been performed or the relative change in $F(Q)$ is less than 10^{-8}.

At the current set of variational parameters $\tilde{\theta}$, the variational posterior is $Q_{\tilde{\theta}}$–Q for short–and the marginal likelihood lower bound is

$$\begin{aligned}
F(Q) &= \mathbb{E}_{\theta\sim Q}\left(\log P(Y^o,\theta) - \log Q(\theta)\right) \\
&= \mathbb{E}_Q\left(\log P(Y^o, Y^m, \beta, S, \Lambda_\epsilon) - \log Q(Y^m,\beta,S,\Lambda_\epsilon)\right) \\
&= \mathbb{E}_Q\left(\log P(Y|\beta,S,\Lambda_\epsilon) + \log P(\Lambda_\epsilon) - \log Q_Y(Y^m) - \log Q_\epsilon(\Lambda_\epsilon)\right) && (18) \\
&\quad + \mathbb{E}_Q\left(\log P(\beta) - \log Q_\beta(\beta)\right) && (19) \\
&\quad + \mathbb{E}_Q\left(\log P(S) - \log Q_S(S)\right) && (20)
\end{aligned}$$

We now compute each part:

$$\begin{aligned}
(18) &= 2\mathbb{E}_Q\left(\log P(Y|\beta,S,\Lambda_\epsilon) + \log P(\Lambda_\epsilon) - \log Q_Y(Y^m) - \log Q_\epsilon(\Lambda_\epsilon)\right) \\
&\equiv \mathbb{E}_Q\Big(N\log|\Lambda_\epsilon| - \left|\left|Y - S\beta\right|\right|^2_{\Lambda_\epsilon} + (e-P-1)\log|\Lambda_\epsilon| - \mathrm{tr}\left(E^{-1}\Lambda_\epsilon\right) \\
&\qquad - \sum_n\left(-\log|\Sigma^{Y_n}| - (Y_{nm}-\mu^Y_{nm})\Sigma^{Y_n}{}^{-1}(Y_{nm}-\mu^Y_{nm})^T\right) \\
&\qquad - \left(-e'\log|\Omega| + (e'-P-1)\log|\Lambda_\epsilon| - \mathrm{tr}\left(\Omega^{-1}\Lambda_\epsilon\right)\right) \\
&\equiv \mathbb{E}_Q\left(-\mathrm{tr}\left(\left[(Y-S\beta)^T(Y-S\beta) + E^{-1}\right]\Lambda_\epsilon\right)\right) + \sum_n \log|\Sigma^{Y_n}| + e'\log|\Omega| \\
&\equiv -\mathrm{tr}\left(\left[\Omega_0' + E^{-1}\right]\Omega\right) + \sum_n \log|\Sigma^{Y_n}| + e'\log|\Omega|
\end{aligned}$$

where Ω_0' is an up-to-date version of the Ω_0 defined above; in particular, I ensure Ω was the last

update, so $\text{tr}\left(\left[\Omega_0' + E^{-1}\right]\Omega\right) = e' \equiv 0$.

$$\begin{aligned}
(19) &= 2\mathbb{E}_Q\left(\log P(\beta) - \log Q_\beta(\beta)\right) \\
&= \mathbb{E}_Q\left(-\tau||\beta||_F^2 - \log|\Lambda_b| + (b-\mu_b)^T\Lambda_b(b-\mu_b)\right) \\
&\equiv -\tau\mathbb{E}_Q\left(||\beta||_F^2\right) - \log|\Lambda_b| \\
&\equiv -\tau\left(||\mu_\beta||_F^2 + \text{tr}\left(\Lambda_b^{-1}\right)\right) - \log|\Lambda_b| \\
(20) &= 2\mathbb{E}_Q\left(\log P(S) - \log Q_S(S)\right) \\
&= \mathbb{E}_Q\left(-||S||_{K^{-1}} - \log|\Lambda_s| + (s-\mu_s)^T\Lambda_s(s-\mu_s)\right) \\
&= -\text{tr}\left(\mathbb{E}_Q\left(SS^T\right)K^{-1}\right) - \log|\Lambda_s| \\
&= -\text{tr}\left(\mu_S^T K^{-1}\mu_S\right) - \text{tr}\left((I\otimes K^{-1})\Lambda_s^{-1}\right) - \log|\Lambda_s|
\end{aligned}$$

Altogether, the marginal likelihood lower bound is

$$\sum_n \log|\Sigma^{Y_n}| + e'\log|\Omega| - \tau\left(||\mu_\beta||_F^2 + \text{tr}\left(\Lambda_b^{-1}\right)\right) - \log|\Lambda_b| - \text{tr}\left(\mu_S^T K^{-1}\mu_S\right) - \text{tr}\left((I\otimes K^{-1})\Lambda_s^{-1}\right) - \log|\Lambda_s|$$

All terms can be computed in $O(N_m P^3 + NP^2)$, again assuming updates have been performed in the order necessary for computations to simplify.

2 Other methods for imputing missing phenotypes

2.1 MVN: an EM algorithm assuming unrelated samples

Rows of Y are not independent in the presence of genetic relatedness between samples due to either population structure or causal genes. Nonetheless, a simple EM algorithm can be derived assuming

$$Y_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma)$$

The resulting EM algorithm infers Σ in an M-step and, among other things, the missing entries of Y in an E-step [14]. As this method ignores correlation across samples, it should do well when there is either little relatedness or little heritability.

Derivation

Given a current parameter estimate $\hat{\Sigma}$, the expected log likelihood is

$$Q(\Sigma|\hat{\Sigma}) \equiv -N\log|\Sigma| - \sum_{n=1}^N \text{tr}\left(\Sigma^{-1}\mathbb{E}_{Y^m|Y^o,\hat{\Sigma}}\left(Y_n Y_n^T\right)\right)$$

where m and o are missing and observed entries, respectively. Letting m_n and o_n be the missing and observed entries of sample n, respectively, define $\hat{Y}$, the implicitly imputed phenotypes, by

$$\hat{Y}_{no_n} = Y_{no_n}, \quad \hat{Y}_{nm_n} = \mathbb{E}\left(Y_{nm_n}|Y_{no_n}, \hat{\Sigma}\right) = \hat{\Sigma}_{m_n o_n}\hat{\Sigma}_{o_n o_n}^{-1} Y_{no_n}$$

Now define the expected sample covariance

$$S := \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{Y^m|Y^o,\hat{\Sigma}}\left(Y_n Y_n^T\right)$$

where

$$\begin{aligned}\mathbb{E}_{Y^m|Y^o,\hat{\Sigma}}\left(Y_n Y_n^T\right)_{i,j} &= \left(\hat{Y}_n \hat{Y}_n^T\right)_{i,j} + \mathrm{Cov}\left(Y_{ni}, Y_{nj}|Y^o, \hat{\Sigma}\right) \\ &= \left(\hat{Y}_n \hat{Y}_n^T\right)_{i,j} + \mathbb{I}\{i,j \in m_n\}\,\Sigma_{ij}^{(n)} \\ \text{where } \Sigma_{ij}^{(n)} &:= \Sigma_{ij} - \Sigma_{i,o_n}\left(\Sigma_{o_n,o_n}\right)^{-1}\Sigma_{o_n,j}\end{aligned}$$

so that

$$Q(\Sigma|\hat{\Sigma}) \equiv -N\log|\Sigma| - \mathrm{tr}\left(\Sigma^{-1}S\right) \implies \Sigma^{(t+1)} = S$$

2.2 LMM: univariate linear mixed models

For each phenotype independently, we run a linear mixed model (LMM) on the observed samples to find the MLE variance components (B_{pp} and E_{pp} in terms of (2)) and then, using these estimates, impute missing samples to their conditional expectations, or BLUPs:

$$\hat{Y}_{m_p,p} := B_{pp}K_{m_p,o_p}\left(B_{pp}K_{o_p,o_p} + E_{pp}I\right)^{-1}Y_{o_p,p}$$

We use the computational trick from [25, 13] to expedite variance component estimation; that is, we first rotate Y by the eigenvectors of K so that the entries of the resulting vector are independent.

2.3 TRCMA: transposable regularized covariance model

The transposable regularized covariance model of [1] (TRCM) uses a mean-restricted matrix normal:

$$Y \sim \mathcal{MN}\left(1_N\mu^T + \nu 1_P^T, R, C\right)$$

The model optionally includes regularization on R^{-1} and/or C^{-1}. An EM algorithm fits maximum penalized likelihood parameter estimates and, as a by-product, imputes missing entries of Y.

TRCMA, a one-step approximation to this EM algorithm, was proposed as a computationally tractable alternative. But even this approximation is much slower than all other methods we have worked with in this paper, especially for large N–all other methods that explicitly model sample relatedness are given K and so can leverage a one-off eigendecomposition of K to derive iterations that are linear or quadratic in N; in contrast, TRCMA has $O(N^3)$ iterations (though it presumably could be modified to use K, or just its eigenvectors, in a similar way). The computational expense is also partially due to the search over regularization parameters: for both precisions in the matrix normal, a penalty amount and type (ℓ_1 or ℓ_2) must be chosen.

We use two shortcuts to mitigate this computational expense. First, we use only ℓ_2 penalization: it is much faster than ℓ_1 (as conditional updates have analytic solutions instead of calls to `glasso`) and [1] found that the ℓ_2 penalty worked well even when the true precision matrices were sparse. Second, we performed preliminary simulations to find a set of reasonable regularization parameters for the model to choose from via cross-validation. Specifically, we searched over $(\rho_{\text{row}}, \rho_{\text{column}}) \in$

$\mathcal{G} := 10^{\{-5,-3.5,-2,-0.5,1\}} \times 10^{\{-6,-4.5,-3,-1.5,0\}}$ in all our analyses. We regularly observed that TRCMA chose regularization parameters in the interior of this grid, suggesting that these ranges are, very roughly speaking, sufficiently wide.

While these two speedups will certainly attenuate accuracy–we could have tried ℓ_1 regularization, tuned the range of $\mathcal{G}$ to each dataset and increased the density of $\mathcal{G}$–we hope our compromise between run time and accuracy is reasonable and representative of the typical choices of end users.

2.4 KNN: k-nearest neighbors

We use the function `impute.knn` from the R package `impute` as a non-paramteric imputation benchmark [26, 6]. We use the default parameters–including, in particular, $k = 10$–except we allow phenotypes with arbitrary amounts of missingness (by default, the program returns an error when phenotypes have $> 80\%$ missingness). The method finds the k-nearest neighbors for each phenotype and then imputes missing values to the average of their observed neighbors.

2.5 mice: multiple imputation by chained equations

We implement this method with the R package `mice` [27]. We use default parameters and average over 5 (the default value) multiply-imputed datasets; we have observed this performs dramatically better than simply taking the first imputed dataset.

`mice` implements a variety of imputation methods, but we only used predictive mean matching (`pmm`), the default for numeric variables. Iterating over phenotypes, the method predicts values for observed and missing samples using the other phenotypes and then matches each missing entry with the closest observed entries based on these predictions (we used the 5 closest matches, which is the default). Missing entries are then imputed to the observed value a randomly chosen partner.

The predictions on which matching is based are made by combining frequentist and Bayesian linear regression on covariates, X. In our implementation of the package, each phenotype p is reqressed on all other phenotypes, so $X = \hat{Y}_{-p}$, where $\hat{Y}_{,-p}$ is the current imputed data matrix after removing phenotype p.

For observed entries, predictions are the OLS fitted values:

$$\hat{Y}_{obs,p} := X_{obs,}\hat{\beta}$$

where $\hat{\beta}$ is the MLE. The missing entries are also of the form $X\beta$, except now the regression coefficients β^* are now drawn randomly from their posterior (using the default $\mathcal{N}\left(0, 10^{-5}I\right)$ prior):

$$\hat{Y}_{miss,p} := X_{miss,}\beta^*$$

2.6 softImpute

We use the softImpute method of [16] as a benchmark from the matrix completion literature in machine learning. We consider this method roughly representative of the state-of-the-art in this field [28, 15], though reported comparisons suggest that the relative performances of the many matrix completion methods depend heavily on the dataset.

softImpute maximizes the penalized likelihood

$$\min_M \sum_{n,p \in obs} (Y_{np} - M_{np})^2 + \lambda ||M||_*$$

where $||M||_*$ is the nuclear norm of M, or the ℓ_1 norm of M's singular values, and measures the complexity of M and thus discourages overfitting. Since the ℓ_1 penalty induces sparsity, the fitted M typically has low rank, which is the key to softImpute's computational efficiency.

Our implementation follows the guide at

`http://web.stanford.edu/~hastie/swData/softImpute/vignette.html`

Specifically: we use the alternating least squares algorithm; we start with the maximum rank set to zero and then, as we shrink the regularization, allow the solution's rank to grow by at most two at each new λ; we vary $\log \lambda$ along 100 evenly spaced points on the interval $[-3 \log 10, \log(\lambda_0 + .2)]$, where λ_0 is the minimum λ such that the solution, $\hat{M}_\lambda$, is 0; and we choose λ by 10-fold cross validation to maximize predictive accuracy.

2.7 MPMM: multiphenotype mixed models

We fit MPMM by estimating the B and E parameters of model (2) on the rows of Y that have been fully observed (i.e. case-wise deletion). We use our `R` implementation from [3], though the command line tool from [33] fits the same model in essentially the same way (modulo a Newton step once the EM algorithm has nearly converged).

Given observed phenotypes and variance component estimates, MPMM imputes missing entries to their conditional expectations, or BLUPs. Defining $\Sigma := (B \otimes K + E \otimes I_N)$,

$$\begin{aligned}
\mathbb{E}\left(y_{miss} | y_{obs}, B, E\right) &= \mathrm{Cov}\left(y_{miss}, y_{obs} | B, E\right) \mathbb{V}\left(y_{obs} | B, E\right)^{-1} y_{obs} \\
&= \Sigma_{miss,obs} \left[\Sigma_{obs,obs}\right]^{-1} y_{obs}
\end{aligned}$$

In general, these computations cost $O(|obs|^3)$ (or $O(|miss|^3)$ if a Schur complement identity is used), and thus the cost of imputing is $O(N^3 P^3)$ if some fixed fraction of entries are missing as N and P vary.

In the special case where samples are either entirely observed or entirely missing, the above conditional expectation can be computed in $O(N^3 + P^3)$. This is because, in this special case, the subsetting operations that select missing or observed entries commute with the Kronecker product structure. Specifically, if M are missing samples and O are observed samples, we can write, by assumption on the missingness pattern, $\mathrm{vec}\left(Y_{O,}\right) = y_{obs}$ and $\mathrm{vec}\left(Y_{M,}\right) = y_{miss}$, and so

$$\begin{aligned}
\mathbb{E}\left(y_{miss} | y_{obs}, C, D\right) &= \left(B \otimes K\right)_{miss,obs} \left[\left(B \otimes K + E \otimes I\right)_{obs,obs}\right]^{-1} y_{obs} \\
&= \left(B \otimes K_{MO}\right) \left[B \otimes K_{OO} + E \otimes I_{|O|}\right]^{-1} \mathrm{vec}\left(Y_{O,}\right) \\
&= \left(B^{1/2} \otimes K_{MO}\right) \left(\left[B^{-1/2} E B^{-1/2}\right] \oplus K_{OO}\right)^{-1} \mathrm{vec}\left(Y_{O,} B^{-1/2}\right)
\end{aligned}$$

By lemma 2, this can be computed in $O(N_O^2 P + N_O N_M P + P^3)$ (by retaining the eigendecomposition of K_{OO} from the parameter learning step).

While this pattern of missingness will essentially never occur in a real dataset–and if it did one would prefer to drop unphenotyped samples since this results in no loss of phenotype data–it does occur in out-of-sample prediction problems, as discussed in [20].

3 Simulation descriptions

3.1 Simulations to assess phenotype imputation accuracy

The results presented in Figure 1 use data simulated from a standard MPMM. Defining `cov2cor` to map covariance matrices to their respective correlation matrices, we draw

$$Y = U + \epsilon \tag{21}$$

$$U \sim \mathcal{MN}\left(0, K, h^2 \texttt{cov2cor}(B)\right) \tag{22}$$

$$\epsilon \sim \mathcal{MN}\left(0, I, (1-h^2)\texttt{cov2cor}(E)\right) \tag{23}$$

We generally take $N = 300$, $P = 15$, B to be an AR(1) matrix with autocorrelation $\rho = .45$ and $E \sim \text{Wi}\left(P, \frac{1}{P}I\right)$, with E being redrawn for each simulated dataset. We use two types of K matrices: either a block diagonal matrix with blocks corresponding to independent sets of 4 siblings or a random subsample, redrawn for each simulated datset, of the kinship matrix derived from the human NSPHS study [11]. Finally, 5% of entries are hidden, completely at random, and their values retained to assess imputation accuracy.

We refer to this as our baseline simulation, and Figure 1 shows the resulting imputation correlations for each method. Supplementary Figures 2-8 all take the same basic form, with each modifying one aspect of the baseline simulation and then plotting the resulting imputation accuracy as in Figure 1. The changes are explained in the plot captions or, when necessary, in the below text. For reference, the results of the baseline simulation from Figure 1 are plotted as dotted lines in the background.

We assessed h^2 at 11 evenly spaced points between .05 and .95. All methods were run on 250 independently simulated datasets for each value of h^2, and averages over these 250 replicates are plotted in all figures. Two hours on a server was more than enough time for all methods to run the 2,750=11×250 datasets, with two exceptions: TRCMA ran only ≈ 125 datasets in the same amount of time and, for the larger data size in Supplementary Figure 3, we ran methods for four hours (LMM still only ran ≈ 1500 datasets and TRCMA ran none).

3.2 Cancellation of genetic and environmental covariances

Simulation results shown in Figure 1 of the main paper suggest that performance generally decreases as heritability increases, but slightly increases at very high levels of heritability. Our hypothesis was that this occured due to cancellation of genetic and environmental covariances. To investigate this we repeated the simulations in Figure 1 with a different model for the genetic covariance (B in (22)) with opposing genetic and environmental correlations i.e. $B_{pq} = -E_{pq}$ for $p \neq q$. In this model, the cancellation is exact at $h^2 = .5$, in that $\mathbb{V}(Y_{i,})$ is diagonal for all i. The results are shown in Supplementary Figure 2. For moderate h^2, genetic and environmental correlations cancel, impeding imputation for multitrait methods relative to the dotted lines, which show the results from Figure 1. At large h^2, the cancellation effect is outweighed by the increased size of $|B_{pq}|$ and so imputation improves.

3.3 Effect of non-random missingness

Our model implicitly assumes that missingness is ignorable in the update for Q_Y (equations (8) and (9)) and we simulate this in our baseline by removing 5% of entries uniformly at random. We can

simulate data with non-ignorable missingness, however, by removing entries of Y, independently, with probability depending on the values of the entries:

$$P(\text{ entry } (i,j) \text{ is missing }) \propto \Phi(Y_{ij})$$

where Φ is the standard normal cdf. The proportionality constant is chosen to ensure 5% overall missingness (in expectation over the random missingness pattern).

3.4 Effect of unmodelled shared environment

We investigated the performance of the different methods in the presence of (unmodelled) shared environmental effects. To do this we added a random effect representing shared environment to the simulated data, in addition to the genetic relatedness and idiosyncratic noise random effects in a standard MPMM:

$$\begin{aligned} Y &= a^2U + c^2C + e^2\epsilon \\ U &\sim \mathcal{MN}\left(0, K, \texttt{cov2cor}(B)\right) \\ C &\sim \mathcal{MN}\left(0, R, \texttt{cov2cor}(D)\right) \\ \epsilon &\sim \mathcal{MN}\left(0, I, \texttt{cov2cor}(E)\right) \end{aligned}$$

Such models are often called ACE models, where U is the Additive efffect, C is a Common environmental effect and ϵ is the purely independent Environmental contribution [4].

We take K, B and E as in the baseline model and D is drawn (independently) from the same distribution as E for each simulated dataset. We define R to be block diagonal with 10 independent environments and each block/environment to be an AR(1) matrix with autocorrelation $\rho = .5$.

Defining the heritability as $h^2 = (a^2 + c^2)/(a^2 + c^2 + e^2)$ and fixing the relative sizes of a^2 and c^2 to three different values given in the caption, the x-axis in Supplementary Figure **??** determines the relative contributions of the unstructured ϵ and the structured U and C.

3.5 Effect of non-normally distributed phenotypes

To create non-normal phenotypes, we start with the baseline MPMM but transform the noise:

$$Y = U + \left(\exp(\epsilon_{ij})\right)_{ij}$$

Phenotype imputation is then performed either on Y or on a quantile normalized version; quantile normalization is natural for most downstream analyses, including GWAS.

3.6 Type I error calibration

To assess the impact of phenotype imputation on the null distribution of p-values in a GWAS, we simulated phenotype data from an MPMM with no genetic contribution beyond the background term U. We imputed missing data and then tested the resulting phenotypes against SNP data and assessed the null distribution of the resulting p-values (Supplementary Figure 9).

We present results for simulations with $N = 300$, $P = 15$, $h^2 = .2$, B an AR(1) with autocorrelation parameter $\rho = .2$ and $E \sim \text{Wi}\left(P, \frac{1}{P}I\right)$; we note the results did not qualitatively change when varying $\rho \in \{-.2, .2, .5\}$ and $h^2 \in \{.1, .2, .5\}$. We chose two types of K matrix, one corresponding

to independent sets of 4 siblings and one a random subsample of the kinship matrix derived from the human NSPHS study [11]. We then added 10% missingness and either dropped missing samples in testing (Unimputed) or imputed with PHENIX, MVN or MPMM; we note the results did not qualitatively change for missingness levels in $\{.01, .05, .1, .2, .5\}$.

We tested both real and simulated genotypes. For the sibling K simulations, we generated SNPs in a hierarchical way: first, we drew parental alleles independently and then we simulated sibling genotypes via Mendel's rules. We simulated 100,000 unlinked loci on which we performed GWAS, for each of the $P = 15$ phenotypes, with `gemma` using the default QC filters (top row of Supplementary Figure 9) [32].

For the simulations where K is a subset of the NSPHS dataset, we used real SNPs corresponding to the same subset of the NSPHS dataset. SNPs were imputed (see Online Methods) and we performed GWAS on the resulting 9,165,236 SNPs with `gemma` using the default QC filters (bottom row of Supplementary Figure9) for each of the 15 phenotypes.

3.7 Power of single phenotype tests

We performed a simulation study to assess the power gains from phenotype imputation. We simulated data using a standard MPMM as before, except now we add a causal SNP:

$$\begin{aligned} Y &= X\beta + U + \epsilon \\ U &\sim \mathcal{MN}(0, K, B) \\ \epsilon &\sim \mathcal{MN}(0, I, E) \end{aligned}$$

We choose $N = 5,000$ and $P = 15$. We also choose B to be AR(1) with autocorrelation parameter $\rho = -.2$ so that, in particular, there is a mixture of positive and negative genetic correlations amongst the phenotypes. We again take $E \sim \text{Wi}\left(P, \frac{1}{P}I\right)$ except now we do not resample E for each dataset but rather fix it at the outset (though U and ϵ are still randomly drawn for each dataset). We choose K to represent independent sets of 4 siblings. $X \in \mathbb{R}^N$ is a common SNP that we draw independently for each dataset by $X_i \overset{\text{iid}}{\sim} \text{Binomial}(2, .2)$.

We choose a pleiotropic β so that the SNP X has a substantial effect on the first phenotype, which represents a phenotype of primary interest, and lesser but non-negligible effects on the other fourteen phenotypes, which represent phenotypes related to and collected with the first, primary phenotype. In this section, we are interested only in the first phenotype, and the other fourteen are valuable only as a means for imputing missing entries in the first. Specifically, we choose β in terms of the implied percent variance explained (PVE) in each of the phenotypes: the PVE for phenotype 1 is 8%, and the other 14 PVEs were drawn randomly:

$$\text{PVE}_{2:15} \overset{\text{iid}}{\sim} \frac{2\text{PVE}_1}{3}\left|\mathcal{N}(0,1)\right|$$

To introduce sparsity into β, the smallest 5 PVE values were then hard-thresholded to 0. The realized values used to create Supplementary Figure 10 are displayed in the first columns of the below table.

	Univariate Test		MV Test, One		MV Test, Sparse		MV Test, Dense	
Phenotype	PVE	Coeff	PVE	Coeff	PVE	Coeff	PVE	Coeff
1	8.00	0.28	8.00	0.28	7.30	0.27	6.00	0.24
2	2.70	0.16	0.00	0.00	2.40	0.15	2.00	0.14
3	2.30	0.15	0.00	0.00	2.10	0.14	1.70	0.13
4	5.40	0.23	0.00	0.00	4.90	0.22	4.00	0.20
5	1.90	-0.14	0.00	0.00	1.70	-0.13	1.40	-0.12
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.30	-0.05
8	7.60	-0.28	0.00	0.00	7.10	-0.27	5.90	-0.24
9	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.03
10	3.40	0.18	0.00	0.00	3.10	0.18	2.60	0.16
11	5.90	-0.24	0.00	0.00	5.50	-0.23	4.60	-0.21
12	2.10	-0.14	0.00	0.00	1.90	-0.14	1.60	-0.13
13	0.00	0.00	0.00	0.00	0.00	0.00	0.70	-0.08
14	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.08
15	4.90	-0.22	0.00	0.00	4.50	-0.21	3.80	-0.19

Randomly generated PVEs and corresponding regression coefficients used to generate Supplementary Figures 10 (first 2 columns) and 11 (last six columns, each pair corresponding to a different line type in S11) for 15 simulated phenotypes. Univariate tests (columns 1 and 2) are performed on phenotype 1; mulitvariate tests (rows 3-8) are performed on all 15 phenotypes. The first entry in each column is non-random while all others were drawn randomly (once) and fixed to the resulting values for all simulated datasets.

3.8 Power of multiple phenotype tests

For each SNP of interest at a time, we use a multi-phenotype mixed model (MPMM) to test association with a set of P phenotypes:

$$\begin{aligned} Y &= X\beta + U + \epsilon \\ U &\sim \mathcal{MN}(0, K, B) \\ \epsilon &\sim \mathcal{MN}(0, I, E) \end{aligned}$$

where $X \in \mathbb{R}^{N\times 1}$ is the vector of genotypes. Specifically, we test $\beta = 0_P$ with the likelihood ratio

$$\text{LRT} = -2\left(ll(\beta = 0, \hat{B}_0, \hat{E}_0) - ll(\beta = \hat{\beta}, \hat{B}_1, \hat{E}_1)\right)$$

where ll is the log-likelihood in the above MPMM and all estimated parameters are MLEs.

Forming the LRT requires fitting variance components (B's and E's), estimating β and evaluating log-likelihoods. Due to the cost of fitting the variance components, we fit fit only $\hat{B}_0$ and $\hat{E}_0$ and then make the approximation $(\hat{B}_0, \hat{E}_0) = (\hat{B}_1, \hat{E}_1)$. Because

$$\max_{\beta,B,E} ll(\beta, B, E) \geq \max_{\beta} ll(\beta, \hat{B}_0, \hat{E}_0) = ll\left(\hat{\beta}\left(\hat{B}_0, \hat{E}_0\right), \hat{B}_0, \hat{E}_0\right)$$

the approximate LRT lower-bounds the exact LRT and our method is conservative. Nonetheless, this approximation is expected to be good for typical analyses, where individual SNPs are expected

to explain a nearly negligible fraction of the overall variance; however, it may attenuate power when analyzing SNPs with very large effect sizes [32].

3.8.1 Simulation details

As in the univariate simulations for Supplementary Figure 10, we choose $N = 5,000$, $P = 15$, B to be AR(1) with autocorrelation parameter $\rho = -.2$, K to represent independent sets of 4 siblings and we draw the common SNP, independently for each dataset, by $X_i \overset{\text{iid}}{\sim} \text{Binomial}\,(2, .2)$. We also take the same E from the univariate simulations, which was drawn $\text{Wi}\left(P, \frac{1}{P}I\right)$

We use three different choices for β in this section to represent varying levels of pleiotropy. In the first situation (UV signal), the causal SNP affects only the first phenotype; in the second (sparse), the SNP affects some (10), but not all, of the phenotypes; in the third (dense), the SNP affects all (15) phenotypes. All 15 phenotypes are tested for association with the SNP X.

We again parameterize our choices for β in terms of the implied PVE. For the first simulation set the PVE to 8% for the first phenotype (and 0 for the others). The other PVEs were derived from the univariate test power simulations: the dense and sparse PVEs were proportional to the PVEs drawn in the previous section prior to and after, respectively, the hard-thresholding step. Proportionalty constants were chosen to yield power away from 0 and 1 (for the tests without added missingness). The resulting PVEs and effect sizes are displayed in the table in Section 3.7.

3.8.2 Computational simplification

In general, the normal equation for regressing the response y on covariates X with noise precision Ω is

$$\hat{\beta}^{MLE} = \left(X^T \Omega X\right)^{-1} X^T \Omega y$$

In our application, we take the covariates to be $I_P \otimes X \in \mathbb{R}^{NP \times P}$ ($X \in \mathbb{R}^{N \times 1}$ by assumption), the response to be $\text{vec}\,(Y) \in \mathbb{R}^{NP}$, and the noise precision, which incorporates the heritable random effect, to be

$$\Omega = (B \otimes K + E \otimes I_N)^{-1} = (L \otimes Q)\,\Lambda^{-1}\,(L \otimes Q)^T \tag{24}$$

where $Q\Lambda_N Q^T$ is an eigendecomposition of K; $Q_P \Lambda_P Q_P^T$ is an eigendecomposition of $B^{-1/2} E B^{-1/2}$; $L := B^{-1/2} Q_P$; $\Lambda := \Lambda_P \oplus \Lambda_N$. This decomposition is closely related to those in [5, 33, 20].

Returning to the normal equation and plugging in the MPMM-specific values for y, X and Ω,

$$\begin{aligned}
\hat{\beta}^{MLE} &= \left((I_P \otimes X)^T \left[(L \otimes Q)\,\Lambda^{-1}\,(L \otimes Q)^T\right](I_P \otimes X)\right)^{-1} (I_P \otimes X)^T \left[(L \otimes Q)\,\Lambda^{-1}\,(L \otimes Q)^T\right] y \\
&= L^{-T} \underbrace{\left(\left(I \otimes [Q^T X]^T\right)\Lambda^{-1}\left(I \otimes [Q^T X]\right)\right)^{-1}}_{\Omega_X} \left(I \otimes \underbrace{[Q^T X]}_{X'}{}^T\right) \text{vec}\left(\underbrace{\left[\text{mat}(\Lambda^{-1}) * \left(Q^T Y L\right)\right]}_{Z}\right) \\
&= L^{-T} \Omega_X \text{vec}\left(X'^T Z\right) \\
&= X'^T Z \Omega_X L^{-1}
\end{aligned}$$

Because we only test one covariate at a time, Ω_X is just a $P \times P$ matrix (if, instead, $D > 1$ covariates are used, this becomes a $DP \times DP$ matrix and requires partial trace operations). In

fact, Ω_X is diagonal with

$$\left(\left(\Omega_X\right)_{pp}\right)^{-1} = X^T Q \left[\Lambda^{-1}\right]_{(pp)} Q^T X = || \left[\Lambda^{-1/2}\right]_{(pp)} X' ||_2^2$$

which is manageable since Λ is diagonal.

Once $\hat{\beta}$ is evaluated, the likelihood can be compactly evaluated for both Y and $Y - X\hat{\beta}$ using previous results [3].

3.9 Calibrating the imputation metric r

To assess the calibration of our imputation metric r, we simulated from our baseline model and compared the true and estimated imputation correlations. We averaged over 1,000 independently simulated datasets. The results are shown in Supplementary Figure 12. The black lines in the top row show the true imputation correlation using our oracle knowledge of the heldout, simulated data, and are essentially identical to the red lines in Figure 1 (we only consider PHENIX in these assessments).

The brown and purple lines show two different estimators for r, which in practice is unknown since the missing data is truly unobserved. Both estimators are formed by first hiding some of the entries of Y^o, the observed part of Y, to form $\tilde{Y}^o$. This new phenotype matrix is then imputed, returning a fully-observed matrix $\hat{Y}$. Finally, r is estimated as the correlation between $\hat{Y}$ and Y^o at the entries hidden from Y^o to create $\tilde{Y}^o$.

The brown and purple lines differ by f, the fraction of Y^o masked to create $\tilde{Y}^o$. As $f \to 1$, $\tilde{Y}^o$ becomes a completely blank matrix and phenotype imputation becomes impossible, yielding estimates of r near 0; conversely, as $f \to 0$, a vanishingly small number of entries of Y^o are masked, resulting in highly variable estimates of r.

We have plotted two choices for f that compromise between this bias at $f = 1$ and variance at $f = 0$. The additional bias from choosing the larger f explains the gap between the purple and brown lines in the top row of Supplementary Figure 12, though even the brown lines are slightly downwardly biased. The additional variance coming from the smaller choice of f is evident but mitigated by our averaging over many simulated datasets. Ultimately, despite this bias and variance, the bottom row of Supplementary Figure 12 shows that our estimates of r are very close and, at worst, conservative.

In practice it is possible to average these r estimates across many replicates of the masking process to create $\tilde{Y}^o$ from Y^o, leading to estimates with lower variance (and thus making choices of small f feasible). In our GWAS, for example, we repeated this sub-sampling 10,000 times with $f = .05$ to remove essentially all sub-sampling variance.

Though this procedure is involved, it is easy to implement in our `R` package. Moreover, this procedure can be performed phenotype-wise, computing imputation correlations within-phenotype and returning a vector of r's. This vector can be used to inform downstream analyses, as we did in our rat GWAS analysis and can be seen in Figure 3.

3.10 Runtimes on simulated and real datasets

Most (method, dataset) pairs were run on 64 2.30 GHz processors (AMD Opteron 6276) in parallel for 12 hours or until all 3,000 simulated missingness patterns had run (100 for each of 30 levels of

added missingness). We made exceptions for the particularly computationally expensive (method, dataset) pairs.

First, MPMM and TRCMA were dramatically more costly than other methods, and so were only run on NSPHS and wheat, two of the smaller datasets (on 64 2.30 GHz processors (AMD Opteron 6276) and 16 3.30GHz processors (Intel Xeon E5-2667) in parallel, respectively). For both these datasets, we ran MPMM on all 3,000 simulated missingness patterns (though it's case-wise deletion approach discarded all data and could not run for 75% and 50% of the patterns in NSPHS and wheat, respectively).

Next, for (TRCMA, NSPHS), by far the most expensive situation studied, we ran on five missingness patterns for each level of missingness below 20%; above this cutoff, one missingness pattern was run for each missingness level. For (TRCMA,wheat) we ran 6 or 7 missingness patterns for each missingness level.

Finally, the chicken dataset had far greater N than any other dataset, which caused LMM and PHENIX–the methods using relatedness–to become far more expensive; for example, a full-rank eigendecomposition of K costs roughly a half hour. We run both these methods on 16 3.30GHz processors (Intel Xeon E5-2667) in parallel for 20 independent missingness patterns at 15 missingness levels (giving 300, rather than 3,000, simulated datasets) without any time constraints.

We note that we could have pre-computed the eigendecomposition of K for PHENIX but not for LMM; the former does not drop samples and thus always works with the same K while the latter drops a different set of samples for each phenotype and thus performs P unique eigendecompositions. For sufficiently large N, this means that performing P LMMs will be P times more expensive than PHENIX, meaning our new method would be both more powerful and much faster.

	N	P	phenix	MVN	LMM	softI	KNN	mice	MPMM	TRCMA
UK BS	1,500	6	0.8	0.1	0.9	0.3	0	0.1		
NSPHS	1,021	15	1.2	0.1	1	0.4	0	0.1	100.8	144 (h)
Wheat	720	7	0.2	0	0.1	0.2	0	0	0.5	8 (h)
Rats	1,407	140	131.2	3.5	16.3	22.9	0	9.7		
Yeast	1,008	46	5.1	0.2	2.6	2.4	0	0.7		
Chickens	11,575	14	89.5	0.8	154.2	4.2	0	4		
Fig 1	300	15	0.1	0	0.1	0.1	0	0.1	7	41
Fig S3	1,000	50	3.9	0.1	9.3	2.2	0	0.9		

Average runtimes for each method on each dataset. Times are in minutes by default, but (h) means the time is in hours. Except TRCMA, MPMM and, on the chicken dataset only, phenix and LMM, all running times were recorded in identical computing environments.

4 Appendix: Jeffreys' prior for matrix factorization

We use a matrix factorization model as our prior on the genetic contribution U:

$$U = S\beta;\ S \sim \mathcal{MN}(0, K, I);\ \beta \sim \mathcal{MN}\left(0, I, \tau^{-1}I\right)$$

As $\tau \to 0$, the prior on β becomes flat (also called objective, or non-informative, because such priors typically deliver unregularized estimates). In contrast, as $\tau \to 0$, the implied prior on U

does become flatter, but *does not* become flat. This means that even in the improper limit of $\tau = 0$–which we use as a default–our prior still encourages U to shrink toward the prior mean of 0.

[17] shows this using the invariance property of Jeffreys priors. First, the Jeffreys prior on U is flat, and therefore the Jeffreys prior on (S, β) induces a flat prior on $S\beta$. But the (improper) Jeffreys prior on (S, β) is, when $N = M = P = 1$,

$$p(S, \beta) \propto \sqrt{S^2 + \beta^2}$$

As $\tau \to 0$, the concave, normal priors that we uses to model S and β become flatter and thus closer to this strictly convex, quadratic Jeffreys prior. This explains why choosing small τ minimizes shrinkage, but it also explains why even $\tau = 0$ cannot eliminate shrinkage.

We derive the Jeffreys prior for general N, M and P below.

Proposition 1. *Let*

$$Y \sim \mathcal{MN}(S\beta, I, I) \tag{25}$$

Then the prior on (S, β) which induces a flat prior on $S\beta$ is

$$p(S, \beta) \propto \sqrt{|S^T S|^{P-M} |\beta\beta^T|^{N-M} |(S^T S) \oplus (\beta\beta^T)|}$$

Proof. Following [17], we first show that the flat prior on U is the Jeffreys prior on U; then, since the Jeffreys prior is invariant under reparameterization, the Jeffreys prior on U is equivalent to the Jeffreys prior on (S, β). This shows the Jeffreys prior on (S, β) induces a flat prior on U.

First, reparameterize the likelihood in terms of $U := S\beta$, so that

$$\ell(Y|U) \equiv -\frac{1}{2}||(Y - U)||_F^2$$

Since this log likelihood is quadratic, the Hessian with respect to U is constant, thus so is its expectation, the Fisher information. Because the Jeffreys prior on U depends only on the Fisher information, it, too, must be constant. Then, since the Jeffreys prior on (S, β) necessarily induces the Jeffreys prior on U, the Jeffreys prior on (S, β) induces the flat prior on U.

Finding the Fisher information requires the log-likelihood derivatives:

$$\begin{aligned}
\frac{\partial \ell(Y|S, \beta)}{\partial S} &= -Y\beta^T + S\beta\beta^T \\
\frac{\partial \ell(Y|S, \beta)}{\partial \beta} &= -S^T Y + S^T S\beta
\end{aligned}$$

This leads to expected second derivatives

$$\begin{aligned}
\frac{\partial}{\partial S_{im}} \frac{\partial \ell(Y|S, \beta)}{\partial S} &= I_{im}\beta\beta^T \implies \nabla_S^2 \ell(Y|S, \beta) = (\beta\beta^T) \otimes I_N \\
\frac{\partial}{\partial \beta_{mp}} \frac{\partial \ell(Y|S, \beta)}{\partial \beta} &= S^T S I_{mp} \implies \nabla_\beta^2 \ell(Y|S, \beta) = I_P \otimes (S^T S) \\
\frac{\partial}{\partial \beta_{mp}} \frac{\partial \ell(Y|S, \beta)}{\partial S} &= -Y I_{mp}^T + S\beta I_{mp}^T + S I_{mp}\beta^T \\
\implies \mathbb{E}\left(\frac{\partial}{\partial \beta_{mp}} \frac{\partial \ell(Y|S, \beta)}{\partial S} | S, \beta \right) &= S I_{mp}\beta^T \implies \mathbb{E}\left(\nabla_S \nabla_\beta \ell(Y|S, \beta)\right) = \beta \otimes S
\end{aligned}$$

and so the Fisher information is

$$\mathcal{I}\left(\mathrm{vec}\left(S\right),\mathrm{vec}\left(\beta\right)\right)=\left(\begin{array}{cc}\left(\beta\beta^{T}\right)\otimes I_{N} & \beta\otimes S\\ \beta^{T}\otimes S^{T} & I_{P}\otimes\left(S^{T}S\right)\end{array}\right) \tag{26}$$

Now the goal is to find the eigenvalues of $\mathcal{I}$. Let $\beta=U_{\beta}D_{\beta}V_{\beta}^{T}$ and $S=U_{S}D_{S}V_{S}^{T}$ be SVDs and whiten $\mathcal{I}$ by conjugating with the orthogonal matrix $U:=\left(U_{\beta}\otimes U_{S}\right)\times\left(V_{\beta}\otimes V_{S}\right)$, where $\times$ is the Cartesian product (or direct sum; we use non-standard notation because we reserve $\oplus$ for the Kronecker sum in this paper):

$$U^{T}\mathcal{I}U=\left(\begin{array}{cc}D_{\beta}D_{\beta}^{T}\otimes I_{N} & D_{\beta}\otimes D_{S}\\ D_{\beta}^{T}\otimes D_{S}^{T} & I_{P}\otimes D_{S}^{T}D_{S}\end{array}\right)=:\mathcal{I}'$$

Define $\Lambda_{\beta}=D_{\beta}D_{\beta}^{T}$ and $\Lambda_{S}=D_{S}^{T}D_{S}$ and let $\lambda_{i}^{S}=\left(\Lambda_{S}\right)_{ii}$, $\lambda_{i}^{\beta}=\left(\Lambda_{\beta}\right)_{ii}$. Then the eigenvalues of $\mathcal{I}$ are roots of the characteristic polynomial:

$$\begin{aligned}
&\left|\mathcal{I}-\lambda I_{M^{2}NP}\right|=\left|\mathcal{I}'-\lambda I_{M^{2}NP}\right|\\
&=\left|\left(\begin{array}{cc}\left(\Lambda_{\beta}-\lambda I_{M}\right)\otimes I_{N} & D_{\beta}\otimes D_{S}\\ D_{\beta}^{T}\otimes D_{S}^{T} & I_{P}\otimes\left(\Lambda_{S}-\lambda I_{M}\right)\end{array}\right)\right|\\
&=\left|\left(\Lambda_{\beta}-\lambda I_{M}\right)\otimes I_{N}\right|\left|I_{P}\otimes\left(\Lambda_{S}-\lambda I_{M}\right)-\left(D_{\beta}^{T}\otimes D_{S}^{T}\right)\left(\left(\Lambda_{\beta}-\lambda I_{M}\right)\otimes I_{N}\right)^{-1}\left(D_{\beta}\otimes D_{S}\right)\right|\\
&=\left(\prod_{m}(\lambda_{m}^{\beta}-\lambda)\right)^{N}\left|I_{P}\otimes\left(\Lambda_{S}-\lambda I_{M}\right)-\left(\left[\Lambda_{\beta}\left(\Lambda_{\beta}-\lambda I\right)^{-1}\right]\times 0_{P-M,P-M}\right)\otimes\Lambda_{S}\right|\\
&=\left(\prod_{m}(\lambda_{m}^{\beta}-\lambda)\right)^{N}\prod_{m=1}^{M}\prod_{p=1}^{P}\left[\left(\lambda_{m}^{S}-\lambda\right)-I\{p\leq M\}\left(\frac{\lambda_{p}^{\beta}}{\lambda_{p}^{\beta}-\lambda}\right)\lambda_{m}^{S}\right]\\
&=\left(\prod_{m}(\lambda_{m}^{\beta}-\lambda)\right)^{N}\left(\prod_{m}\left(\lambda_{m}^{S}-\lambda\right)\right)^{P-M}\prod_{m,m'=1}^{M}\left[\left(\lambda_{m}^{S}-\lambda\right)-\lambda_{m}^{S}\left(\frac{\lambda_{m'}^{\beta}}{\lambda_{m'}^{\beta}-\lambda}\right)\right]\\
&=\left(\prod_{m}(\lambda_{m}^{\beta}-\lambda)\right)^{N-M}\left(\prod_{m}\left(\lambda_{m}^{S}-\lambda\right)\right)^{P-M}\prod_{m,m'=1}^{M}\left[\left(\lambda_{m}^{S}-\lambda\right)\left(\lambda_{m'}^{\beta}-\lambda\right)-\lambda_{m}^{S}\lambda_{m'}^{\beta}\right]\\
&=\left(\prod_{m}(\lambda_{m}^{\beta}-\lambda)\right)^{N-M}\left(\prod_{m}\left(\lambda_{m}^{S}-\lambda\right)\right)^{P-M}\prod_{m,m'=1}^{M}\left[\left(\lambda-(\lambda_{m}^{S}+\lambda_{m'}^{\beta})\right)\lambda\right]\\
&=\left|\Lambda_{\beta}-\lambda I\right|^{N-M}\left|\Lambda_{S}-\lambda I\right|^{P-M}\left|\lambda I-\Lambda_{\beta}\oplus\Lambda_{S}\right|\lambda^{M^{2}}
\end{aligned}$$

As in Appendix 1 of [17], I take the Jeffreys prior proportional to the square root of the product of non-zero eigenvalues of the Fisher information.

□

5 Appendix: Useful Linear Algebra Idenitities

Lemma 1. *Let $A\in\mathbb{R}^{P\times P}$ and $X\in\mathbb{R}^{N\times N}$. Then $tr_{P}\left(A\oplus X\right)^{-1}$ can be computed in $O(NP+P^{3})$ given the matrix of eigenvalues of X, Λ_{X}.*

Proof. First,

$$\mathrm{tr}_P\left(A \oplus X\right)^{-1} = \mathrm{tr}_P\left[\left(U_A \otimes U_X\right)\left(\Lambda_A \oplus \Lambda_X\right)^{-1}\left(U_A \otimes U_X\right)^T\right] = U_A\left[\mathrm{tr}_P\left(\left(\Lambda_A \oplus \Lambda_X\right)^{-1}\right)\right]U_A^T$$

To compute the right hand side, the eigendecomposition of A must be performed ($O(P^3)$), an NP diagonal matrix must be inverted ($O(NP)$) and partial-traced out ($O(NP)$), and finally $P \times P$ matrix multiplications are performed ($O(P^3)$).

□

Lemma 2. *Let $A \in \mathbb{R}^{P\times P}$, $X \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times P}$ and let $X = Q_X\Lambda_X Q_X^T$ be a known eigendecomposition of X. Then $\left[A \oplus X\right]^{-1}$ vec(B) can be computed in:*

- $O(P^3 + N^2P)$ *in general*
- $O(P^3 + NP^2)$ *if X is diagonal*
- $O(P^3 + RP^2 + RNP)$ *if X has rank R*
- $O(P^3 + RP^2)$ *if X is diagonal and has rank R*

Proof. First,

$$\left(\left[A \oplus X\right]^{-1}\right)\mathrm{vec}\left(B\right) = \left(U_A \otimes Q_X\right)\underbrace{\left[\Lambda_A \oplus D\right]^{-1}\mathrm{vec}\left(Q_X^T B U_A\right)}_{\mathrm{vec}(Z)} = \mathrm{vec}\left(Q_X Z U_A^T\right)$$

There are four types of operations above

1. eigendecomposition of A
2. multiplication of an $N \times P$ matrix with a $P \times P$ matrix (BU_A and ZU_A^T)
3. matrix multiplication an $N \times N$ matrix with an $N \times P$ ($Q_X^T B$ and $Q_X Z$)
4. diagonal $NP \times NP$ matrix operations

In general, 1 costs $O(P^3)$; 2 costs $O(NP^2)$; 3 costs $O(N^2P)$; and 4 costs $O(NP)$. When X is diagonal, $Q_X = I$ and 3 can be elided. If X is low-rank, B and Z can be compressed to $\mathbb{R}^{R\times P}$ and the cost of 2 becomes $O(RP^2)$; analogously, 3 becomes $O(RNP)$ and 4 becomes $O(RP)$. Finally, if additionally X is diagonal, 3 can again be skipped.

□

References

[1] Genevera I Allen and Robert J Tibshirani. Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics*, 4(2):764–790, June 2010.

[2] Emamnuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[3] Andy Dahl, Victoria Hore, Valentina Iotchkova, and Jonathan Marchini. Network inference in matrix-variate Gaussian models with non-independent noise. *arXiv:1312.1622v1*, pages 1–17, December 2013.

[4] D.S. Falconer and Trudy Mackay. *Introduction to Quantitative Genetics*.

[5] Nicholas A Furlotte and Eleazar Eskin. Efficient multiple trait association and estimation of genetic correlation using the matrix-variate linear mixed-model. *Genetics*, pages genetics–114, 2015.

[6] Trevor Hastie, Robert Tibshirani, and Gavin Sherlock. Imputing missing data for gene expression arrays. *Technical Report, Division of Biostatistics, Stanford University*, pages 1–9, 1999.

[7] Alexander Ilin and Tapani Raiko. Practical approaches to principal component analysis in the presence of missing values. *The Journal of Machine Learning Research*, 11:1957–2000, 2010.

[8] Hyun Min Kang, Noah A. Zaitlen, Claire M. Wade, Andrew Kirby, David Heckerman, Mark J. Daly, and Eleazar Eskin. Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709–23, March 2008.

[9] Yong-deok Kim and Seungjin Choi. Variational Bayesian view of weighted trace norm regularization for matrix factorization. *IEEE Signal Processing Letters*, 20:261–264, 2013.

[10] Arthur Korte, Bjarni J Vilhjálmsson, Vincent Segura, Alexander Platt, Quan Long, and Magnus Nordborg. A mixed-model approach for genome-wide association studies of correlated traits in structured populations. *Nature genetics*, 44(9):1066–1071, 2012.

[11] Gordan Lauc, Abdelkader Essafi, Jennifer E. Huffman, Caroline Hayward, Ana Knežević, Jayesh J. Kattla, Ozren Polašek, Olga Gornik, Veronique Vitart, Jodie L. Abrahams, Maja Pučić, Mislav Novokmet, Irma Redžić, Susan Campbell, Sarah H. Wild, Fran Borovečki, Wei Wang, Ivana Kolčić, Lina Zgaga, Ulf Gyllensten, James F. Wilson, Alan F. Wright, Nicholas D. Hastie, Harry Campbell, Pauline M. Rudd, and Igor Rudan. Genomics meets glycomics-the first gwas study of human N-glycome identifies HNF1A as a master regulator of plasma protein fucosylation. *PLoS Genetics*, 6(12):1–14, 2010.

[12] Yew Jin Lim and Yee Whye Teh. Variational Bayesian approach to movie rating prediction. *Proceedings of KDD Cup and Workshop*, 7:15–21, 2007.

[13] Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M CM Kadie, Robert I Davidson, and David Heckerman. FaST linear mixed models for genome-wide association studies. *Nature Methods*, 8(10):833–835, January 2011.

[14] Roderick J A Little and Donald B Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[15] Dehua Liu, Tengfei Zhou, Hui Qian, Congfu Xu, and Zhihua Zhang. A nearly unbiased matrix completion approach. In *Learning and Knowledge Discovery in Databases*, pages 210–225. Springer Berlin Heidelberg, 2013.

[16] Rahul Mazumder, Trevor Hastie, and Robert J Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, March 2010.

[17] Shinichi Nakajima and Masashi Sugiyama. Theoretical analysis of Bayesian matrix factorization. *The Journal of Machine Learning Research*, 12:2583–2648, 2011.

[18] Shinichi Nakajima, Masashi Sugiyama, S Derin Babacan, and Ryota Tomioka. Global analytic solution of fully-observed variational Bayesian matrix factorization. *The Journal of Machine Learning Research*, 14(1):1–37, 2013.

[19] Shinichi Nakajima, R Tomioka, M Sugiyama, and S Derin Babacan. Perfect Dimensionality Recovery by Variational Bayesian PCA. *Advances in Neural Information Processing Systems*, pages 971–979, 2012.

[20] Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt, and Oliver Stegle. It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. In *Advances in Neural Information Processing Systems*, pages 1466–1474, 2013.

[21] Benjamin Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, 2011.

[22] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. *International Conference on Machine Learning*, pages 880–887, 2008.

[23] Vincent Segura, Bjarni J Vilhjálmsson, Alexander Platt, Arthur Korte, and Magnus Nordborg. An efficient multi-locus mixed-model approach for genome-wide association studies in structured populations. *Nature genetics*, 44(7):825–830, July 2012.

[24] Oliver Stegle, Leopold Parts, Richard Durbin, and John Winn. A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies. *PLoS computational biology*, 6(5):e1000770, May 2010.

[25] Matthew Stephens and David J Balding. Bayesian statistical methods for genetic association studies. *Nature Reviews Genetics*, 10(10):681–690, October 2009.

[26] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, Trevor Hastie, Robert J Tibshirani, David Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–5, June 2001.

[27] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate Imputation by Chained Equations in R. *Journal Of Statistical Software*, 45(3):1–67, 2011.

[28] Zheng Wang, Ming-Jun Lai, Zhaosong Lu, Wei Fan, Hasan Davulcu, and Jieping Ye. Orthogonal Rank-One Matrix Pursuit for Low Rank Matrix Completion. *Proceedings of the 31st International Conference on Machine Learning*, pages 91–99., 2014.

[29] BS Weir, AD Anderson, and AB Hepler. Genetic relatedness analysis: modern data and new challenges. *Nature Reviews Genetics*, 7(10):771–80, October 2006.

[30] Jianming Yu, Gael Pressoir, WH William H Briggs, Irie Vroh Bi, Masanori Yamasaki, John F Doebley, Michael D McMullen, Brandon S Gaut, Dahlia M Nielsen, James B Holland, Stephen Kresovich, Edward S Buckler, and IV Bi. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature genetics*, 38(2):203–208, February 2006.

[31] Keyan Zhao, MJ Aranzana, Sung Kim, Clare Lister, Chikako Shindo, Chunlao Tang, Christopher Toomajian, Honggang Zheng, Caroline Dean, Paul Marjoram, and Magnus Nordborg. An Arabidopsis example of association mapping in structured samples. *PLoS genetics*, 3(1):e4, January 2007.

[32] Xiang Zhou and Matthew Stephens. Genome-wide efficient mixed-model analysis for association studies. *Nature genetics*, 44(7):821–4, July 2012.

[33] Xiang Zhou and Matthew Stephens. Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nature methods*, 11(4):407–409, February 2014.