Methods to jointly analyze multiple phenotypes

Andrew Dahl

Merton College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2016

Acknowledgements

First, I would like to thank my supervisor Jonathan Marchini for suggesting and advising the phenix project and comments on early versions of the GLMM project. I would also like to thank my course director Julian Knight for invaluable career and life advice. The GMS community has given me broad and consistent support, from Jonathan Flint encouraging me to come to Oxford to stimulating pub conversations with Alex, Charly, Matthias and Patrick. I thank the Wellcome Trust for funding my research.

I wish to thank Victoria Hore and Lloyd Elliot for their contributions to the GLMM project and extensive research discussions, and Amelie Baud for extensive help with rat GWAS data. I also thank Winni, Kevin and Robbie for always interesting lunch conversations ranging from programming to basic biology to politics.

Finally, I want to thank my extensive personal support network. I would not possibly have pursued this degree without lifelong guidance and encouragement from my parents, and I likely would have quit many times without the academic and emotional support of Jenni, my partner in everything.

Methods to jointly analyze multiple phenotypes

Andrew Dahl, Merton College, University of Oxford

Thesis submitted for the degree of Doctor of Philosophy

Trinity 2016

Linear mixed models (LMMs) have re-emerged as a central tool in statistical genetics. Fixed effects capture genetic variants tested for association. Random effects leverage aggregate relatedness while remaining agnostic to specific genetic mechanisms, naturally modeling heritability and controlling for polygenic background and confounding from population or family structure in genome-wide association studies (GWAS). Multiple random effects can partition heritability amongst many biologically meaningful variance components (VCs).

Concurrently, genetic studies have begun to analyze multiple traits. This can improve power by adding data and can inform the path from genotype to phenotype, e.g. with graphical models, pleiotropy detection or endophenotyping. Multi-trait analyses are natural for biobanks and high-throughput phenotypic measurements like gene expression, medical images or metabolites.

Following these advances, this thesis develops three multi-trait mixed models. phenix imputes missing phenotype data by modifying probabilistic matrix factorization to incorporate genetic relatedness. Gen-

eral linear mixed models (GLMMs) generalize and unify multi-VC, multi-trait and likelihood-penalized mixed models. Finally, compressive mixed models (CMMs) combine the two, obtaining the imputation and computational benefits of phenix and the heritability estimation and multi-VC capabilities of GLMMs.

phenix essentially always outperforms all competitors in imputation and can improve GWAS power. GLMMs accurately estimate heritability despite (measured) confounders, can improve phenotype prediction, and increase gene-based, multi-trait association signal. CMMs regularly improve prediction, scale to thousands of phenotypes, and can uncover plausible GWAS hits entirely missed by LMMs. Altogether, multi-trait mixed models are invaluable for intrinsically multi-trait tasks, like phenotype imputation and low-rank decomposition, and, surprisingly, can be much faster than LMMs; however, I find only small, and inconsistent, benefits for single-trait-oriented objectives like heritability estimation and out-of-sample prediction.

The challenges in this thesis are primarily computational. Naively, multi-trait approaches model an $N \times P$ matrix of P phenotypes measured on N samples as a long Gaussian vector, inducing prohibitive $O(N^3P^3)$ computations. Fortunately, the parsimonious matrix normals underlying mixed models enable simpler $O(N^3 + P^3)$ expressions. This is summarized by a new decomposition for positive semidefinite tensor products that, under a crucial assumption, facilitates cheap evaluation of ubiquitous low-level operations like multiplication.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

$\mathbb{R}$	The real numbers
$\mathbb{R}^{n \times m}$	The $n \times m$ matrices of real numbers
I_n	The $n \times n$ identity matrix
1_n	The $n \times 1$ vector of 1s
J_n	The $n \times n$ matrix of 1s
e_i	The i-th canonical basis vector
I_{ij}	The matrix with entry ij=1 and all other entries 0
PSD	Positive semidefinite
$\mathbb{S}_+$	The $P \times P$ PSD matrices
N	The number of genotyped samples
P	The number of measured phenotypes
Y	The partially-observed matrix of phenotypes in $\mathbb{R}^{N \times P}$
K	The kinship matrix in $\mathbb{S}_+^N$ representing genome-wide genetic similarity
G	A genotype matrix, either genome-wide or within some region
h^2	The heritability of a trait
$Q_X \Lambda_X Q_X^T$	The eigendecomposition of a matrix X
λ_X	The eigenvalues of a matrix X

x The column-wise vectorization of a matrix X

X The column-wise matricization of a vector x

$\otimes$ The Kronecker product of two matrices

$\oplus$ The direct sum of two matrices

$*$ The Hadamard product of two matrices

$\mathrm{tr}_P(X)$ The partial trace of a matrix X

$X_{,j}$, $X_{i,}$ The j-th column or i-th row of a matrix X

$X_{,-j}$, $X_{-i,}$ All but the j-th column or i-th row of a matrix X

$I\{\omega\}$ The 0/1 indicator of the event ω

$\equiv$ Formal equivalence (context dependent)

$\propto$ Proportionality

$\stackrel{d}{=}$ Equal in distribution

i.i.d. Independent and identically distributed

$\mathbb{E}(X)$ The expected value of the random variable X

$\mathbb{V}(X)$ The variance of the random variable X

$\mathcal{N}(\mu, \Sigma)$ A multivariate Gaussian random variable with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{S}_+^n$

$\mathcal{MN}(M, R, C)$ A matrix-variate Gaussian random variable with mean $M \in \mathbb{R}^{n \times m}$ and row- and columb-covariance $R \in \mathbb{S}_+^n$ and $C \in \mathbb{S}_+^m$

PCA Principal component analysis

CCA Canonical component analysis

RRR Reduced rank regression

MF Matrix factorization

GWAS Genome-wide association study

LD Linkage disequilibrium

Chapter 1

Introduction

Aiming to increase power to find genetic associations with complex traits and to enrich the interpretation of these associations, this thesis develops new tools for multiple trait analyses that incorporate genetic similarity between samples. The former aspect–multiple trait modelling–has been approached many ways, typically by repurposing standard methods from statistics/machine learning, and will be reviewed in Section 1.1. As is standard in the field, the latter aspect–incorporating genetic similarity–will be addressed with mixed models, which posit that samples are correlated due to a genetically determined Gaussian effect; these are reviewed in Section 1.2 below and in greater detail, where relevant, throughout the thesis.

Modern single-trait genetic analyses, which aim for thousands or, ideally, hundreds of thousands of samples, demand so-called efficient inference, or lower-complexity variants of standard algorithms. Multi-trait models require additional manipulations to extend beyond a few traits. Further, multiple random effects, or variance components (VCs)–to represent multiple levels of relatedness–and likelihood penalization–to computationally, statistically and intuitively simplify estimates–have become useful tools in genetics that present additional algorithmic challenges. A primary goal of this thesis is to merge these various (albeit simi-

lar) computational ideas into algorithms that are more general, more efficient or, ideally, address novel questions and produce novel biological insights.

1.1 Multitrait analysis

Multiple related phenotypes can be analyzed simultaneously to boost power to detect association with genetic variation, estimate coheritability–the shared genetic basis between traits–and learn graphical models relating the phenotypes to each other and to statistically significant genetic variants. Such analyses are motivated by pleiotropy, the notion that causal genetic variants affect clusters of related traits either by distinct modes of action or, more simply, by affecting a low-level endophenotype that, in turn, affects many high-level observed traits. Of course, if a variant truly affects only one observed trait, univariate analysis of that trait alone is optimal–adding in irrelevant phenotypes cannot aid estimation.

Many studies have found evidence of pleiotropy, some going so far as to call it ubiquitous in complex traits [Sivakumaran et al., 2011; Cotsapas et al., 2011; Yang et al., 2013; Solovieff et al., 2013; Visscher and Yang, 2016; Pickrell et al., 2016]. Pleiotropic markers are naturally studied with multiple traits, not only to increase association power but also to elucidate the mechanisms underlying multitrait associations [Stephens, 2013]. As genetic markers in a region are typically highly correlated (inter-variant correlation is known as linkage disequilibrium, or LD), it can be difficult to distinguish one pleiotropic locus from multiple nearby and non-pleiotropic loci [Gianola et al., 2015], though some methods aim to resolve this ambiguity by incorporating prior information [Pickrell et al., 2016].

A natural first approximation to multitrait analysis is to aggregate single trait analysis results *post hoc*. The simplest way to do this is Bonferroni correction,

which makes the worst-case assumption that the tests for each trait are independent, but more powerful approaches have been suggested. For example, [Yang et al., 2010b] takes a decorrelated average of single-trait statistics; [Sluis, Posthuma, and Dolan, 2013] additionally injects sparsity into this averaging process, aiming to improve power and detect which markers affect which traits. These approaches are appealing because they model genotypes only through single-trait GWAS methods, which generally are more mature, more diverse and faster than their multitrait analogues.

Another way to repurpose single-trait GWAS pipelines for multiple traits is to test univariate combinations of traits. Ideally, these transformations would be chosen to combine traits with shared genetic bases to maximize power to detect pleiotropic loci. In practice, such transformations are rarely known *a priori* (though many observed traits, like BMI, are implicitly functionals of other traits), motivating the application of generic, unsupervised statistical methods to define interesting axes of trait covariation.

By far the commonest of such approaches is principal component analysis (PCA). PCs decompose variation among the phenotypes into orthogonal directions explaining successively less overall trait variation. More formally, the k-th PC explains as much variation in the phenotypes as possible subject to the restriction that the it is orthogonal to PCs 1 through $k-1$. For this reason, the top k PCs provide an ℓ_2-optimal rank-k representation of the data. Further, PCs have analytic solutions: they are simply the singular vectors of the phenotype matrix and, thus, can be conveniently computed via singular value decomposition (SVD).

As an example, consider a matrix of gene expression traits, and imagine expression variation is driven only by three effects: a large experimental batch effect

with no biological significance, a trans-eQTL modestly affecting the expression of 10% of the genome, and white noise. Typically, the first PC will capture the batch effect, the second PC will capture the trans effect, and remaining PCs will simply be noise. Performing GWAS on all PCs (as it is generally not known *a priori* which, or how many, PCs are biologically significant) will powerfully detect the trans effect for two reasons. First, the second PC will primarily weight the 10% of genes that are affected by the eQTL, approximating a natural (but *a priori* infeasible) test of only the relevant genes. Less obviously, the first PC–capturing a batch effect–will also aid power: as the second PC must be orthogonal to the first (by construction), it is automatically corrected for the batch effect.

Whether GWAS on PCs is powerful–and which PCs correspond to genetic effects [Aschard et al., 2014]–depends entirely on the genetic architecture of the traits at hand. The above example is a best-case scenario as the genetic signal is confined to a single PC–rather than spread over many–and the batch effect is entirely absorbed by the first PC. The first attribute is particularly unlikely to hold in general because PCs are defined without any knowledge of genotype data. In the example, there was only one trans effect, and it was strong; typically, though, trans effects will be dwarfed by batch and cis effects. In such cases, PCA has no particular incentive to dedicate each genetic mechanism to a distinct PC, and pleiotropic signal will not necessarily be more parsimoniously represented with PCs than with observed traits.

This problem would be solved if PCs could be defined to directly partition genetic variance. This is the goal of principal components of heritability (PCH). Under a monogenic model without background relatedness or population structure, [Klei et al., 2008] shows how to choose a linear trait contrast to maximize

association power. Unlike PC testing, which first learns trait transformations without supervision and then performs testing, PCH marries the two steps to uncover trait combinations with greater genetic meaning, obtaining calibrated p-values by sample splitting.

PCH is a domain-specific approach to supervised dimensionality reduction, but many generic approaches exist and have been applied in genetics more or less out-of-the-box. Two prominent examples are canonical correlation analysis (CCA) and reduced-rank regression (RRR). Both seek axes explaining large covariation between two matrices, with these matrices comprising genotypes and phenotypes in genetics; this contrasts with PCA, which aims only to explain phenotypic variation. CCA is similar to PCA, but with variance maximization replaced by correlation maximization: genotype-phenotype correlations are partitioned into orthogonal axes of decreasing importance. Unlike CCA, RRR treats the two matrices asymmetrically, regressing phenotype on genotype under a rank constraint on the (matrix of) regression coefficients.

Both approaches render a low-rank representation of the map between genotype and phenotype. The statistical significance of this map can be directly assessed to test overall genotype-phenotype correlation, fully integrating dimensionality reduction and testing, in both CCA [Ferreira and Purcell, 2008] and RRR [Marttinen et al., 2014] frameworks (the latter a Bayesian RRR with confounder correction). Further, CCA variants with sparse [Chen, Liu, and Carbonell, 2012; Chi et al., 2013] or group-sparse [Lin, Calhoun, and Wang, 2014; Du et al., 2016] phenotype loadings have been used to refine interpretation by pairing genetic associations with the (parsimonious) set of traits with nonzero loadings. As CCA and RRR model phenotypes exchangeably, they are most appropriate for datasets with ho-

mogeneous phenotypes and have been commonly used for brain imaging, gene expression, and metabolite traits.

An intrinsically multi-trait task is to learn graphical models between traits and genetic variants. Such graphs can be used to maximize association power and interpretability by differentiating between traits that are genetically determined and those that are only indirectly affected by genetics [Stephens, 2013; Aschard et al., 2016]. Roughly speaking, conditioning on confounders improves power and reduces false positives, but conditioning on other phenotypes sharing underlying causal genetic structure will mask the causal effect and potentially induce false positives.

For small datasets, a closely related method can be used to include multiple SNPs in the graph [Scutari et al., 2014]. Such a model can in principle delineate the entire causal path from all causal genotypes to all phenotypes. A real example of such a graph (albeit one learned using a different method) is given below in Figure 1.1 (reproduced from [Hore et al., 2015]). Directed arrows indicate which nodes directly affect others. Nodes correspond either to a single SNP or the expression of various genes, where the SNP was GWAS-significant and the genes were hand-selected based on correlations with this SNP. Overall, the directed graph in this picture gives a vaguely informative but very simplistic picture of gene expression regulation: the SNP affects CIITA (in blue), a known trans-regulator, which in turn affects the other genes. It is unclear whether the link from the SNP to HLA-DOA is genuine or merely reflects unmodelled confounding or higher-order relationships in the expression pathway that are not sufficiently captured by the linear model. Of course, any directed graph required to be acyclic cannot realistically represent gene expression–unless there are no regulatory feedback mechanisms

whatsoever.

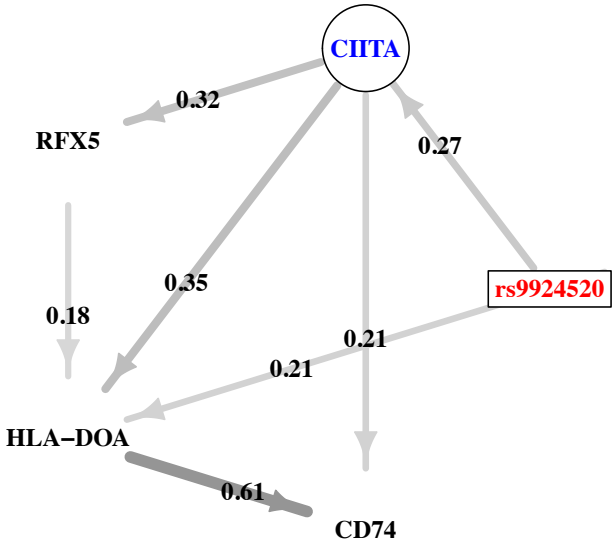


Figure 1.1: Example acyclic digraph for a SNP and genes identified in MHC class II gene regulation. Causal orderings are estimated using DirectLiNGAM [Shimizu et al., 2011] subject to the requirement no edge is directed into the SNP. All displayed edge strengths are significant ($p < 0.01$) conditional on the estimated causal ordering. Variables are standardized to mean zero and unit norm so that effect sizes can be interpreted as the fraction of variance explained.

The main strength of these approaches is their generality and flexibility: CCA, RRR, Bayesian networks, high-dimensional regression and their variants can all be applied to any data matrices. While this makes their application to genetic data simple and robust, it also means these applications cannot harness the vagaries of genetic data. In particular, population structure, often an overwhelmingly strong contributor to phenotypic variance, is entirely ignored[1].

Quite generally, correcting for genetic PCs *a priori* or, when possible, including them as covariates may mitigate these problems, and such steps make generic

[1]An exception is [Marttinen et al., 2014], which models confounders (albeit without genetic relatedness), though the approach was applied to at most 200 SNPs.

approaches palatable. However, more flexible models for confounding are often needed (see below mixed model discussion), and PC adjustment is not state-of-the-art for GWAS. Certainly, sparse methods attenuate this problem–as population structure, and other typical confounders, affect the whole genome–but local ancestry is sparse and non-trivial. Moreover, there is no reason to think a truly sparse, small causal component would be preferred by a generic model over a sparsified version of a large confounding component.

A final type of multitrait analysis reverses the ordinary approach by asking which phenotypes associate with a given variant (dubbed PheWAS) [Denny et al., 2010; Denny et al., 2011; Denny et al., 2013; Glicksberg et al., 2016]. Such approaches are attractive for large datasets [Marx, 2015], e.g. biobanks or electronic health record databases. A related analysis attempts to uncover parsimonious endophenotypes or further sub-classify known phenotypes using subtle variations in a large number of proxy phenotypes. Typically, such approaches restrict to a gene known to be causal: for example, autism and *CDH8* [Bernier et al., 2014]; autism and six genes known to associate with autism-related diseases [Bruining et al., 2014]; or the monogenic Huntington's disorder [Alexandrov et al., 2016]. Additionally, the order can be reversed, first detecting endophenotypes and then finding endophenotype-specific variants, with the aim of improving power and interpretability [Li et al., 2015]. These approaches differ from the generic RRR and CCA by leveraging *a priori* biological knowledge to improve power, confidence in biological meaningfulness and, crucially, computational and statistical tractability.

1.2 Mixed models in genetics

Mixed models generally decompose a response variable into fixed and random effects, typically representing foreground signal and background correlations between samples, respectively. This structured background noise violates standard independent-and-identically-distributed (i.i.d.) assumptions and, if ignored, can attenuate power and induce spurious associations. Yet such structure commonly arises from time series measurements, spatial structure or, more generally, any unobserved confounders.

In particular, mixed models are well-suited for genetics, where confounding–due to unmodelled causal genetic variants, population structure and cryptic relatedness–is ubiquitous and typically dramatically stronger than the causal signals of primary interest. Moreover, mixed models are theoretically attractive to geneticists because they convey genuine uncertainty about the complex causal genetic mechanisms underlying phenotypic variation. Despite this agnosticism to the particular form of confounding effects, it remains crucial to specify appropriate random effect models to maximize power and minimize spurious findings [Yang et al., 2014].

Mixed models originated to describe aggregate genetic effects with Fisher and were further developed, largely for breeding purposes, by Henderson. More recently, mixed models in genetics have been used in linkage analysis [Abecasis et al., 2001; Almasy and Blangero, 1998], the randomness representing uncertainty in causal locus location in addition to polygenic effects. Now, with genome-wide genotypes readily available, mixed models are re-emerging as a state-of-the-art approach to control for relatedness, population structure and genome-wide polygenicity in human GWAS. Moreover, as statistical genetics moves beyond vanilla

GWAS, mixed models are growing in prominence as tools to broadly decompose phenotypic variation into biologically meaningful components.

1.2.1 Heritability estimation with mixed models

Heritability (h^2) is defined as the fraction of phenotypic variation that can be explained by a linear function of genotype data (this thesis does not discuss "broad-sense" heritability). Mixed model based heritability estimation in humans is a young field, originating in the mid-2000s and coming to prominence a few years later. See [Zaitlen and Kraft, 2012] and [Campos, Sorensen, and Gianola, 2015] for reviews from the human- and animal breeding-perspectives, respectively.

The approach, at least in human genetics, originated in pedigree based studies. Heritability analysis is standard for such data, but the innovation was to use observed, rather than expected, kinship [Visscher et al., 2006]; this required genome-wide genetic data and allowed for deconvolution of shared family background and genetic similarity, and as the latter represents exogenous random variation it can be used to estimate the (aggregate) causal genetic effect sizes. This approach was then extended to unrelated humans [Yang et al., 2010a; Yang et al., 2011], where explicit control for family structure is replaced by *a priori* QC that is assumed to eliminate non-causal relationships between genotype and phenotype; similar assumptions were implicit in [Visscher et al., 2006], which required families to be independent.

This approach was introduced as a generalization of pedigree-based mixed models, but it also has a connection with Bayesian linear regression. Assume the standard genome-wide linear model relating the genotype matrix $G \in \mathbb{R}^{N \times L}$ to the

phenotype vector $y \in \mathbb{R}^{N \times 1}$:

$$y = G\beta + \epsilon$$

A standard prior on β is a spherical Gaussian, i.e. $\beta \sim \mathcal{N}\left(0, \frac{1}{L}\sigma_g^2 I_L\right)$. This is often called a polygenic, or infinitesimal, model, as every SNP (column of G) contributes an equal and small fraction of the overall genetic variance σ_g^2. (Using the central limit theorem, similar results to those given below can be derived using weaker assumptions on the distribution of β and a limit where $L \to \infty$.)

The homoscedastic prior on G is less restrictive than it appears as G may have columns with different variance, and thus any heteroscedasticity among SNP effects can be absorbed into the column sizes of G. In particular, using a G matrix with columns normalized to unit variance–as is nearly ubiquitous–combined with the homoscedastic assumption on β becomes an assumption that the effect size of the unnormalized SNPs is inversely proportional to allele frequency; at least qualitatively, this aligns with the intuition that selection will typically drive highly penetrant alleles to lower frequencies in the population [Wakefield, 2009]. Nonetheless, this genotype normalization is not necessary, and other choices for genotype normalization have been studied and shown to perform better in certain cases [Speed et al., 2012].

Motivation aside, given the linear model regressing y on G and the homoscedastic prior on β it is easy to integrate out β:

$$y = G\beta + \epsilon \overset{d}{=} \mathcal{N}\left(G0, \frac{1}{L}\sigma_g^2 G I_L G^T\right) + \mathcal{N}\left(0, \sigma_e^2 I_N\right) \overset{d}{=} \mathcal{N}\left(0, \sigma_g^2 K + \sigma_e^2 I_N\right)$$

defining $K = \frac{1}{L}\sum_{l=1}^{L} G_{,l} G_{,l}^T$ to be the realized/genome-wide relatedness matrix, also known as the Gram matrix or a linear kernel. Note that K simply averages the correlation between samples that is observed at each SNP.

This easily generalizes to more than one phenotype by standard properties of the matrix variate normal [Dawid, 1981], which will be formally introduced in Section 2.1. Assuming $\beta \sim \mathcal{MN}\left(0_{L\times P}, I_L, \frac{1}{L}B\right)$, where $B \in \mathbb{S}_+^P$ gives overall genetic covariance between traits,

$$\begin{aligned} y = G\beta + \epsilon &\overset{d}{=} \mathcal{MN}\left(G0, \frac{1}{L}\sigma_g^2 G I_L G^T, B\right) + \mathcal{MN}\left(0, I_N, E\right) \\ &\overset{d}{=} \mathcal{MN}\left(0, K, B\right) + \mathcal{MN}\left(0, I_N, E\right) \end{aligned}$$

where E is the environmental counterpart to B. This simple calculation is only meant to illustrate that the equivalence between Bayesian regression and (co)variance decomposition is not limited to the $P = 1$ case. In fact, a near-identical calculation worked for general $P > 1$, requiring only the replacement of multivariate normals by matrix variate normals. This is a common theme of this thesis, and the primary means of generalizing single trait results and methods to multiple traits will often involve replacing multivariate normals with matrix variate normals.

The normalization of G discussed above ultimately results in reweighting each SNP's contribution to overall relatedness. Fundamentally, the above equation relates regression on y to a variance decomposition of y, linked through the prior that allows marginalization of the specific manifestation of the regression coefficient β, instead focusing the analysis on the size of β. The right hand side of this equation is the same heritability estimation problem that was originally motivated through an analogy to pedigree based studies, and thus uncovers and elucidates implicit assumptions in the approach of [Visscher et al., 2006; Yang et al., 2010a].

Critics quickly uncovered subtle problems with this method, most significantly the agnostic approach to causal genetic variants [Golan and Rosset, 2011; Speed et al., 2012]; the literature offers no clear consensus whether this is an intrinsic

problem or one that can be overcome by LD-smoothing [Speed et al., 2012] or MAF-stratification [Lee et al., 2013]. Recent work [Kumar et al., 2016a], based partially on an incorrect application of Sylvester's identity (performing the matrix analogue of dividing by 0) has argued the approach is intrinsically biased and unstable, leading only to further confusion in the field [Yang et al., 2016; Kumar et al., 2016b]. In forthcoming work based on approximate characterization of the heritability MLE [Steinsaltz, Dahl, and Wachter, 2016], we argue that bias and inflated variance do not intrinsically pose a substantial problem. Nonetheless, real data may badly violate the assumptions of the model, in which case heritability estimates can be arbitrarily biased. Overall, it seems clear mixed model heritability estimation has some merit, though it may be wise to take results with a grain of salt.

Some work has been done on estimating the variance of heritability estimators, with (delta method approximate) frequentist and Bayesian approaches compared in [Furlotte, Heckerman, and Lippert, 2014]. The asymptotically accurate information matrix can also be used [Visscher et al., 2014; Visscher and Goddard, 2015], and a more detailed calculation shows this asymptotic is quite generally applicable [Steinsaltz, Dahl, and Wachter, 2016].

Finally, some recent progress has been made on approximating mixed models. LD score regression [Bulik-Sullivan et al., 2015b] infers heritability as the correlation between LD score–a measure of the genome-wide, infinitesimal signal captured by each individual marker simply through LD with other markers–and a measure of phenotype association. This method is extremely fast, extensible [Bulik-Sullivan et al., 2015a; Finucane et al., 2015], simple and requires only on GWAS summary data.

A related approximation is Haseman-Elston regression. This approach also equates sample and theoretical moments, this time using (vectorized) genotypic and phenotypic similarity, and shares the parsimony and speed benefits of LD score regression. The approach is also more robust to various model-misspecification than more complex (and, by assumption, misspecified) mixed model methods [Golan, Lander, and Rosset, 2014].

Another approach first approximates the p-value for the hypothesis that $h^2 > 0$ (a null hypothesis rarely taken seriously) and inverts it (using an approximate distribution function for h^2) to estimate h^2 [Ge et al., 2015]. This is shown to approximate $-\log_{10}$ p-values very well, though the approach has clear inaccuracies in estimating h^2, and the authors prefer to use Haseman-Elston regression for hundreds of phenotypes [Ge et al., 2016]. Nonetheless, the approach is extremely fast and may be invaluable when only a rough estimate of heritability is needed–e.g. for initializing another algorithm–or when other methods are intractable.

Heritability partitioning with multi-component mixed models

Mixed models can incorporate arbitrarily many random effects, which in genetics can be used to incorporate multiple notions of relatedness: for example, close and distant to address different definitions of heritability [Zaitlen et al., 2013] or improve prediction [Tucker et al., 2015]; or ordinary and dominant genome-wide similarity to assess genetic architecture [Zhu et al., 2015; Muñoz et al., 2014]. These other types of relatedness need not be genome-wide, and by modelling multiple regions of the genome simultaneously–defined, for example, by functional annotation [Cross-Disorder Group of the Psychiatric Genomics Consortium et al., 2013; Loh et al., 2015b; Finucane et al., 2015] or chromosome [Yang et al., 2011; Lee

et al., 2012b; Kostem and Eskin, 2013]–heritability can be partitioned amongst meaningful components of genetic variation. Further, these additional components can be defined adaptively, which complicates testing and computation but improves predictive performance [Speed and Balding, 2014]. Next, random effects can themselves be tested, enabling, for example, gene-set tests [Listgarten et al., 2013]. Finally, known environmental confounders can be included as random effects [Heckerman et al., 2016], as is standard in mixed model applications outside genetics.

Coheritability estimation with multitrait mixed models

As discussed above, multi-trait methods are valuable for their ability to improve power [Zhou and Stephens, 2012; Lippert et al., 2014] and uncover genetic structure underlying trait covariation. The latter is naturally accomplished with multi-trait mixed models, which give estimates of coheritability: the proportion of trait covariance explained by genetics. Assuming heritable trait covariation is due to pleiotropic loci affecting the covarying traits, coheritability estimates loosely inform the degree of pleiotropy.

Multi-trait random effects can also be used to improve out-of-sample prediction by sharing information across traits [Rakitsch et al., 2013]. Another application learns heritable graphs [Stegle et al., 2011]; these differ from the multi-trait graphical models above by explicitly incorporating aggregate genetic independence between traits rather than by building a graph between specific variants and traits.

Criticisms of genome-wide heritability estimates

Above, I have described how mixed models can be used to estimate heritability for polygenic traits, as the random effect captures small contributions from each SNP. Below, in Section 1.2.2, I will describe how mixed models use the random effect to control for confounding in GWAS. How can the genetic random effect simultaneously describe causal genetic effects and confounders like population structure and relatedness? I have not heard an answer. In unpublished work [Steinsaltz, Dahl, and Wachter, 2016], we make a boring and obvious claim: the random effect describes the "correlation" between genomic and phenotypic similarity, and thus incorporates whatever combination of population structure, family structure and polygenic effects that has induced phenotypic similarity in the dataset at hand.

And yet [Yang et al., 2010a] attributes the entire size of the random effect to heritability and the approach has exploded in popularity, at least partially, because it increases heritability estimates. But of course the estimate increases if population structure is contributing to phenotypic variation. [Yang et al., 2010a] argues population structure (and other relatedness) is absent (after QC) because there is no inter-chromosomal LD, presenting a table of p-values that are all Bonferroni-insignificant (after correcting for $231 = \binom{22}{2}$ tests). These p-values are represented in Figure 1.2; it is obvious to the naked eye that these p-values are not null, and a simple Kolmogorov-Smirnov test returns $p < 10^{-6}$ that these p-values are drawn from a uniform distribution.

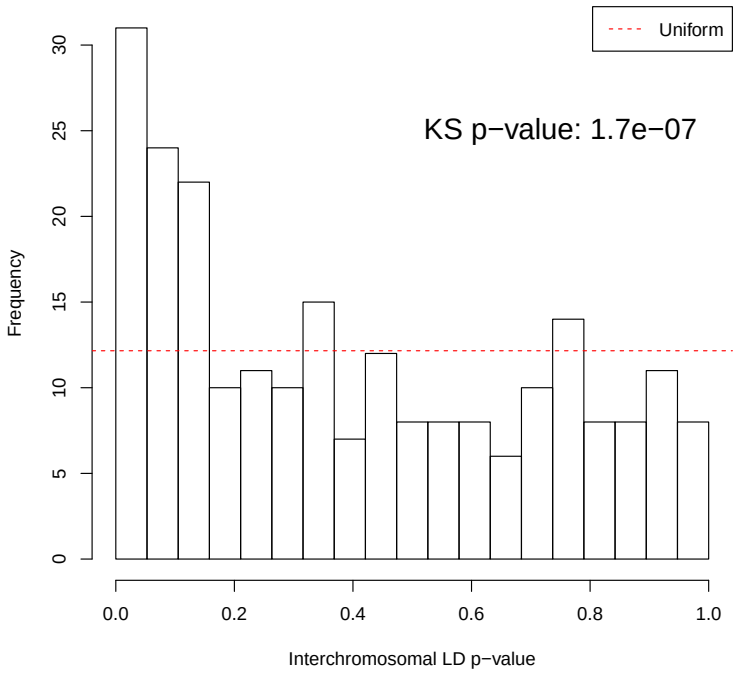


Figure 1.2: Inter-chromosomal LD p-values reported in Supplementary Table 2 of [Yang et al., 2010a]. The original conclusion was that there is no inter-chromosomal LD, and thus GCTA truly estimates heritability. However, the p-values clearly deviate from the null, shaking the foundation on which all subsequent mixed model heritability methods have been built. p-values come from regression of one chromosomal-relatedness matrix on another, in "3,925 unrelated individuals."

This observation inspires skepticism of many mixed model results. Where possible, I will attempt to make claims that are robust to this issue; for example, the incursion of confounders is irrelevant in simulated data or when assessing accuracy of phenotype prediction or imputation, and the latent CMM factors in Section 5.5 can be validly tested regardless their genetic or confounding nature. However, some applications, particularly the heritable networks in 4.4.3, would be hard to take too seriously in real data.

On the other hand, any available proxies for these offending confounders can enable (at least partial) correction. Such an analysis, with simulated data, is performed in Section 4.4.1, and requires multi-trait mixed models that can harness multiple VCs (like those developed in Sections 4 and 5). Ultimately, extensive correction for potential structure is advised, and even then the biological interpretation of VC sizes may be suspect.

1.2.2 GWAS with mixed models

The fundamental idea in LMM-based GWAS is to use a genome-wide random effect to capture confounding signal. This confounding is often interpreted as coming from genome-wide polygenicity, though the random effect can also model population or family structure. After controlling for this confounding background, fixed effect tests for large effect loci can be performed more powerfully and robustly. (Detailed algorithm descriptions will be given later.)

Many additions to this basic framework have been proposed. LMM-select modifies the above Bayesian linear regression model based on the assumption that many, but far from all, SNPs are causal, i.e. $y = G_{,S}\beta_S + \epsilon$ for some set S of causal SNPs [Lippert et al., 2013]. Further assuming that β_S remains homoscedastic–which may be less tenable as a model on causal effects, which can vary over orders of magnitude–the above mixed model framework can be exactly reused, modifying only the definition of the kinship matrix to only include contributions from the active SNPs in S, i.e. $K_S := \frac{1}{|S|}\sum_{l\in S} G_{,s}G_{,s}^T$. But this requires already knowing S, the set of causal SNPs–the original goal of the analysis. Nonetheless, it may be that K_S is relatively robust to mis-identification of S, in which case an approximation $K_{\hat{S}}$ would still perform well. In practice, S is defined by sorting SNPs via

linear regression and then defining $|S|$ by optimizing some out-of-sample criterion (though, to my knowledge, no sample-splitting is used to separate the detection of S and the use of S to perform GWAS testing). Broadly, LMM-select is expected to perform well when it's motivating assumption–that a small fraction but large number of SNPs are causal–is satisfied, though it may underperform when family or population structure is present [Widmer et al., 2014].

PC-select combines LMM-select with principal component regression, aiming retain the benefits of LMM-select while simultaneously correcting for family/population structure by incorporating PCs [Tucker, Price, and Berger, 2014]. This is closely related to the original approach in [Yu et al., 2006], which separately controlled for polygenic similarity and population structure. If the family/population structure is strong enough, top PCs will capture these confounders [Patterson, Price, and Reich, 2006] and thus including them in a regression will correct for the confounding [Price et al., 2006].

Further, the GRM can be replaced by a rank-one approximation to control for structure in some situations where PCs work better than an ordinary GRM [Sul and Eskin, 2013]. This illustrates the flexibility of mixed models, as many problems can be rectified simply by using a different definition of relatedness. The Bayesian linear regression duality, in particular, can be used to guide appropriate definition of the relatedness matrix K, as it directly corresponds to covariates implicitly regressed upon.

Finally, step-wise approaches have been developed to model multiple large-effect loci as fixed effects in addition to the random effect for polygenic background [Segura et al., 2012]. This is an alternative approach to the same problem motivating LMM-select: in both cases, the new approach aims to distinguish be-

tween large-effect SNPs and background SNPs. While LMM-select ignores the background SNPs, as they contribute nothing in the absence of confounding, the approach in [Segura et al., 2012] retains them in the background kinship matrix with the aim of absorbing confounders.

As GWAS typically focus on disease traits, methods to accurately apply mixed models to case-control studies have been invaluable in practice [Lee et al., 2011; Lee et al., 2012a]. However, repurposing standard machinery developed for continuous traits may be biased [Golan, Lander, and Rosset, 2014], and though this effect depends upon parameters of the data [Lee, 2015] it seems that simple moment methods are likely to extend better to large datasets [Golan, Rosset, and Lander, 2015], not to mention they obtain improved runtime[2]. Another binary trait problem, preferential case and family ascertainment, leads to biased effect estimates but can be corrected [Hayeck et al., 2015; Hayeck et al., 2016].

1.2.3 Out-of-sample phenotype prediction

Additionally, estimated VCs can be used for out-of-sample predictions [Campos, Gianola, and Allison, 2010], though the prediction performance may rapidly decay with genomic distance between the training and testing samples as the underlying linearity approximation decays [Wray et al., 2013].

Genomic selection, or animal breeding using genetic information, is closely related to prediction, and many tools developed in the two fields overlap. The former uses best linear unbiased predictions (BLUPs) in-sample as a proxy for fitness, while the latter uses BLUPs out-of-sample. Originating in the days of linkage,

[2]The authors cite a roughly four fold runtime improvement; more importantly, though, the method requires only the inner product of two sample covariance matrices, which is $O(N^2)$ rather than the standard $O(N^3)$ mixed model cost

mixed models were argued to be more effective than predictions based on only loci of large effect, and this effect was further improved with a Bayesian approach [Meuwissen, Hayes, and Goddard, 2001]. Various computational approaches to handle genotype data were studied in [Legarra and Misztal, 2008]. Around the same time, the value of the realized relationship matrix–what I call the GRM–over the traditional pedigree-based kinship was becoming clear [Hayes, Visscher, and Goddard, 2009]. Further, variants of the mixed model Gaussian priors were shown to add value in some settings [Goddard, 2009; Gianola et al., 2009], and choice of effect size prior remains an area of research [Gianola, 2013].

1.3 Computation of mixed models in genetics

Mixed models have long been used to test locus effects while controlling for polygenic background in humans, at least since 2002 [Abney, Ober, and McPeek, 2002]. This paper preceded dense genotyping, and additionally had to cope with genotype uncertainty within a locus. Further, this paper, to my knowledge, first identified an invaluable approximation: assuming individual genetic effects are negligible, the random effect part of the model need only be fit once. (This paper also derived appropriate permutation procedures for non-i.i.d. data.)

To my knowledge, the next step was a similar approach rediscovered by [Yu et al., 2006], now applied to genome-wide SNP data. This paper incorporated population structure by regressing on STRUCTURE [Pritchard, Stephens, and Donnelly, 2000; Pritchard et al., 2000] admixture components as fixed effects (which is very similar to regressing on PCs [Price et al., 2006]: see [Patterson, Price, and Reich, 2006; Zhao et al., 2007; Engelhardt and Stephens, 2010] and, especially, Proposition 2 in Text S4 of [Lawson et al., 2012]).

Evidence then emerged that a single random effect, appropriately constructed to convey genome-wide relatedness, can accurately control for many levels and types of relatedness in highly-structured model organisms [Zhao et al., 2007]. This incited the first "efficient" optimization algorithms to fit mixed models [Aulchenko, Koning, and Haley, 2007; Kang et al., 2008], which were then applied to humans [Kang et al., 2010] and further expedited using the (rediscovered) small-SNP-effect approximation from [Abney, Ober, and McPeek, 2002]. This was quickly followed by further algorithmic improvements, all in a similar vein [Zhou and Stephens, 2012; Lippert et al., 2011; Svishcheva et al., 2012; Pirinen, Donnelly, and Spencer, 2013].

Subsequent work has focused on generalizing the efficient computations to models with multiple random effects, multiple related phenotypes or likelihood penalties; these are reviewed in detail in Section 4, which aims to unify these disparate contributions. Another significant direction involves new optimization algorithms–rather than faster implementations of standard algorithms–for example using conjugate gradient descent to circumvent matrix inversions [Legarra and Misztal, 2008; Loh et al., 2015a; Loh et al., 2015b]; such approaches seriously threaten the standard eigendecomposition-based approaches used in the above references and this thesis.

Finally, more generic computational advances have been proposed to leverage low-level algebra libraries [Gray, Stewart, and Tenesa, 2012] and GPU architectures for regional heritability [Cebamanos et al., 2014]; orthogonal to algorithmic advances, these implementation ideas could presumably be incorporated within any mixed model algorithm.

1.4 The type of data used in this thesis

I always denote the number of samples by N, the number of phenotypes by P, and the matrix of phenotypes by $Y \in \mathbb{R}^{N \times P}$. Y may or may not be fully observed, and may have entirely blank rows for the case of out-of-sample prediction. I always assume columns of Y are standardized to mean zero and variance 1, and I additionally quantile-normalize the columns in real data analyses. This last step is not generally applicable if the observed scale matters, but the aim is only to put methods on an even, standardized footing for comparison; when performing GWAS downstream, however, I do not quantile normalize for imputation (though phenix can internally quantile normalize and then "invert" the output to return imputations on the original scale).

I will also always use $K \in \mathbb{S}_+^N$ to represent the kinship, or genome-wide relatedness. All algorithms will allow any definition of K (of which there are many), but I construct K in simulations and examples using the "realized" relationship matrix

$$K = \frac{1}{L} X X^T$$

where $X \in \mathbb{R}^{N \times L}$ is the matrix of genotypes at L markers. The columns of X are standardized to mean-zero and variance 1: the former implies K has row-, column- and grand-mean 0; the latter implies $\mathrm{tr}\,(K) = \frac{1}{L}\|X\|_F^2 = \frac{1}{L}\sum_l \|X_{,l}\|^2 = N$. K is the central element in ordinary random effect models in genetics [Kang et al., 2008; Yang et al., 2010a].

In Sections 4 and 5, I will also use $M - 1$ incidence matrices for additional random effects, $\{G_i\}_{i=2,\ldots,M}$ (reserving $G_{i=1} = X$ for the GRM), each with rank r_i and with linear kernel matrix $K_i := \frac{1}{r_i} G_i G_i^T$. These, too, are standardized:

columns are demeaned and, if G_i has L_i columns, $\frac{1}{L_i}\|G_i\|_F^2$ is re-scaled to N; this is equivalent to rescaling $\mathrm{tr}\,(K_i) = N$. The reason I discuss the GRM as a PSD matrix and all other effects as the square-root of a PSD matrix is notational: the former is anticipated to be full-rank, while the latter will be required to be low-rank.

The rescaling is innocuous as a scalar can freely pass between the incidence matrix and the VC. The demeaning, used in practice for simplicity, can also be described as projecting out the intercept in a REML analysis.

1.5 Thesis outline

Chapter 2 reviews notation and useful linear algebra results. In particular, I derive computations for partial traces and Kronecker products that are used throughout the thesis to reduce computational costs. I then show how to efficiently compute what I call the GLMM decomposition, which is central to Sections 4 and 5. I then define a slightly modified algorithm for Cholesky decomposition that is necessary for the GLMM decomposition to obtain minimal computational complexity.

Chapter 3 uses a probablistic matrix factorization model to impute missing entries in a phenotype matrix. This standard approach from machine learning is modified to incorporate genetic relatedness in a modified mixed model framework, thus combining state-of-the art approaches from genetics and machine learning. The approach, phenix, is fit with variational Bayes and scales to thousands of individuals and a few hundred traits. I show phenix improves imputation accuracy over generic matrix completion and imputation methods as well as standard mixed models and then apply phenix to a published GWAS dataset and find novel, biologically plausible hits.

Chapter 4 presents the general linear mixed model (GLMM), which generalizes and unifies multi-trait, multi-VC, and penalized linear mixed models into one simple framework. Each of these generalizations uses a unique computational trick, and my primary result is efficiently combining these tricks to obtain all advantages simultaneously. I then derive computationally efficient expressions useful for various GLMM applications, like imputation, prediction and likelihood ratio testing. Finally, I evaluate a few of these applications: first, showing the utility of GLMMs in simulated multi-trait, multi-VC data; then boosting association signal with a multi-trait gene-test in a positive control from real data; then estimating simulated graphical models with a generalization of the graphical lasso; and finally predicting traits out-of-sample using multiple phenotypes and likelihood penalization.

Chapter 5 aims to combine the phenix model from Chapter 3 and the GLMMs from Chapter 4 to create compressed linear mixed models (CMMs). CMMs use variational Bayes machinery, similar to phenix but with richer priors. Unlike phenix, CMMs extend to thousands of traits and can estimate coheritability and incorporate multiple random effects. Unlike GLMMs, CMMs consistently improve out-of-sample prediction (in five real datasets), though by a small margin. More interestingly, CMMs are applied to the same GWAS data as in the phenix section, and the latent CMM factors uncover associations not remotely significant in univariate analyses. Moreover, these associations naturally correspond to phenotype sets that suggest biologically plausible stories underlying the novel associations.

Chapter 6 concludes the thesis with a brief discussion of future extensions of multitrait mixed models and, more generally, multi-trait models of causal genetic variation in the presence of unobserved confounding. An appendix contains technical proofs, most of which are well known.

Chapter 2

Useful linear algebra notation and computations

2.1 Definitions

Throughout the thesis, I often use relative change when monitoring an algorithm's convergence. This is always defined with respect to some norm, ℓ_2 by default, and the relative change between some generic X and Y is

$$\frac{\|X - Y\|}{\frac{1}{2}\|X\| + \frac{1}{2}\|Y\|}$$

The Kronecker product (KP), denoted $\otimes$, of matrices $A \in \mathbb{R}^{N \times N}$ and $B \in \mathbb{R}^{P \times P}$ is defined by

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1N}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1}B & A_{N2}B & \cdots & A_{NN}B \end{bmatrix}$$

Equivalently, the KP of A and B can be defined as the tensor product of the associated linear maps. KPs are central to matrix variate normals (introduced below) and to nearly all the algebraic manipulations in this thesis.

Unlike the full tensor product space $\mathbb{R}^{N \times N} \otimes \mathbb{R}^{P \times P}$, KPs (or pure tensors) are not closed under addition. Thus, extending well-known results for KPs to sums

of KPs is not trivial or even generally possible. But these KP identities are key to performing efficient computations, and similar lemmas for sums of KPs will be essential for efficiently fitting mixed models. Thus the primary mathematical objective in this thesis is to derive simplifying identities for sums of KPs, at least in special cases (as formalized by Assumption 1 below).

Matrix normals

Before turning to the general matrix normal, which can be written parsimoniously using KPs, it may be helpful to first write out the special case where the matrix X has only two columns. The goal of the matrix normal is to generalize the multivariate normal distribution, and thus the marginal distribution of each column of X will multivariate normal. Specially, and for now assuming X has mean zero for simplicity, the column marginals of X will be

$$X_{,1} \sim \mathcal{N}\left(0, \sigma_{11}^2 R\right); X_{,2} \sim \mathcal{N}\left(0, \sigma_{22}^2 R\right)$$

These distributions require the columns of X have identical correlation (defined through R) but possibly different variances (defined through σ_{11}^2 and σ_{22}^2). In other words, I have assumed that both columns of X are identically distributed modulo scalar multiplication.

The key novelty of the matrix normal is in the model for correlations between the two columns of X. In typical i.i.d. analyses, these columns would simply be assumed independent. A fully general model, on the other hand, would not constrain the correlation between $X_{i,1}$ and $X_{j,2}$ in any way. In many settings, estimating the N^2 inter-column correlation parameters–one for each (i, j) pair–introduced by the fully general model may be infeasible, motivating a more parsimonious model for

inter-column correlations. Matrix normals provide one such option by requiring:

$$\mathrm{Cov}\left(X_{i,1}, X_{j,2}\right) = R_{ij}\sigma_{12}$$

That is, the inter-column correlation between samples i and j is the same as the within-column correlations but multiplied by an attenuation factor σ_{12}. When $\sigma_{12} = \sigma_{11}^2 = \sigma_{22}^2$, the two columns are perfectly correlated; at the other extreme, when $\sigma_{12} = 0$, the two columns are independent. In this way, σ_{12} provides a continuous relaxation of the independence constraint between these two columns, flexibly modelling the entire spectrum of inter-column correlation.

Under all the above conditions, the matrix X has mean 0, row-covariance R and column covariance $\Sigma := \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{12} & \sigma_{22}^2 \end{pmatrix}$.

More generally, $X \in \mathbb{R}^{N\times P}$ has a matrix normal distribution with mean $M \in \mathbb{R}^{N\times P}$, between-row covariance $R \in \mathbb{S}_+^N$ and between-column covariance $C \in \mathbb{S}_+^P$, written $X \sim \mathcal{MN}\left(M, R, C\right)$, if [Dawid, 1981]

$$x \sim \mathcal{N}\left(m, C \otimes R\right)$$

using the convention that lower-case letters refer to the column-wise vectorization of a matrix, i.e. $x = \mathrm{vec}\left(X\right)$ (and vice versa). Note the sum of matrix normals is not (generally) a matrix normal as the sum of KPs is not (generally) a KP.

As indicated in the two-column case, R and C have the row- and column-interpretations because:

$$X_{i,} \sim \mathcal{N}\left(M_{i,}, R_{ii}C\right); \quad X_{,j} \sim \mathcal{N}\left(M_{,j}, C_{jj}R\right)$$

That is, each row of X has covariance proportional to C, the across-column, within-row covariance matrix. (An analogous statement holds for the columns of X; the

transposability of the matrix normal is a notable feature.) In this way, matrix normals generalize the standard multivariate Gaussian: when $R = I$, rows of X are simply i.i.d. Gaussians with covariance C (possibly with different means); but a general R allows correlation between rows, giving a richer dependence structure across entries of the matrix.

Matrix normals are the key ingredient in our formulation of multiphenotype mixed models. Specifically, both genetic and environmental contributions (and, potentially, additional random effects) will be modeled as matrix normal random variables. Matrix normals are well suited for this purpose as each random effect will have unique covariance structures across both its rows and columns–for example, the genetic effect will have correlation across samples due to genetic relatedness and "genetic correlations" across traits, while the environmental effect will be independent across different samples but will have "environmental correlations" across traits. Similar decompositions of variance are the central idea of univariate random effect models, and matrix normals naturally lift this idea to the decomposition of covariance.

The partial trace

The partial trace $\mathrm{tr}_P : \mathbb{R}^{NP \times NP} \mapsto \mathbb{R}^{P \times P}$ can be defined as the unique linear functional such that $\mathrm{tr}_P(A \otimes B) = A\mathrm{tr}(B)$ for all $A \in \mathbb{R}^{P \times P}, B \in \mathbb{R}^{N \times N}$.

There is also a more concrete definition (that is equivalent by Lemma 5 in the Appendix). Write $M \in \mathbb{R}^{NP \times NP}$ as a $P \times P$ matrix of $N \times N$ blocks:

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

Here and elsewhere, (11) is a multi-index referring to a block of the matrix M.

The partial trace applies the trace to each $N \times N$ block, giving a $P \times P$ matrix of traces:

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

$\mathrm{tr}_N(\cdot)$ is defined analagously (I never consider cases where $N = P$, so this is never ambiguous).

KPs and partial traces in quantum mechanics

The partial trace is a natural way to restrict covariance across a system to covariance across a subsystem; it was first used in quantum physics to describe a measurement operator that collapses a two-particle system to a one-particle system. The distribution of non-interacting particles is the tensor product of their marginal distributions [Aerts and Daubechies, 1978], and so it is appropriate that

$$\mathrm{tr}_P(X \otimes Y) \propto X\mathrm{tr}(Y) \tag{2.1}$$

(See Lemma 2 in the appendix for proof.) That is, combining X with Y via the tensor product and then reducing back simply to X gives back the original answer. The partial trace is a natural opposite of the tensor product in this sense: a tensor product unites particles into a many-particle system and the partial trace recovers a single particle from the many.

In this thesis, the 'particles' will be the sample and trait dimensions of a phenotype matrix, and the non-interacting tensor product is simply the Kronecker product. Analagously, I create (sample,trait) covariance by combining sample and trait covariances with a tensor product–using standard matrix normal priors–and recover trait covariances using the partial trace. This use of the partial trace is

inspired by [Kalaitzis et al., 2013], which used the partial trace for efficient inference when a Gaussian vector had precision given by a (sparse) Kronecker sum. However, unlike [Kalaitzis et al., 2013], which treated the matrix dimensions symmetrically, I will fix (a basis for) row-covariance *a priori* and apply the partial trace only along the phenotype dimension.

2.2 Partial trace identities

The partial trace has many useful properties that often eliminate the need for element-wise operations, simplifying interpretation and notation. Primarily, I will use Lemma 2, which generalizes (2.1), to simplify computation via

$$\mathrm{tr}_P\left((C \otimes R)\, X\right) = C\mathrm{tr}_P\left((I \otimes R)\, X\right) \tag{2.2}$$

This identity is useful when X and R have simple structure–which I will require throughout the thesis with the below Assumption 1–as generic $NP \times NP$ computations are simplified to a $P \times P$ computation and 'simple' $NP \times NP$ computations.

A natural dual to (2.2), which identifies a property for $\mathrm{tr}_P(\cdot)$ along the $P \times P$ dimension, is the partial trace equivalent of the cyclic property, which occurs along the $N \times N$ dimension:

$$\mathrm{tr}_P\left((I \otimes X)R\right) = \mathrm{tr}_P\left(R(I \otimes X)\right) \tag{2.3}$$

This follows directly from the definition of the partial trace and cyclic property of

the trace:

$$\begin{aligned}
\mathrm{tr}_P\left((I \otimes X)R\right)_{pq} &= \mathrm{tr}\left(((I \otimes X)R)_{(pq)}\right) \\
&= \mathrm{tr}\left((I \otimes X)R\left(I_{pq} \otimes I\right)\right) \\
&= \mathrm{tr}\left(R(I \otimes X)\left(I_{pq} \otimes I\right)\right) \\
&= \mathrm{tr}_P\left(R(I \otimes X)\right)_{pq}
\end{aligned}$$

A final identity allows extraction of a rank-one matrix from the partial trace (see Lemma 3 in the appendix for proof):

$$\mathrm{tr}_P\left(uv^T\left[I \otimes X\right]\right) = U^T X^T V \tag{2.4}$$

As all matrices are sums of rank-one matrices (and the partial trace is linear), this can be applied to extract any matrix from a partial trace. However, this only simplifies computation for low-rank matrices; this is essentially the algorithmic reason for imposing Assumption 1, which ensures the relevant quantities will be low rank.

In Chapter 4, the partial trace serves to project $NP \times NP$, element-wise covariance across a matrix to $P \times P$ covariance across traits (see Section 4.3.1 for details). In particular, disparate projections of the sample covariance matrix yy^T will repeatedly be evaluated when fitting GLMMs, essentially using $u = v = y$ and $X = K_i$ for some relatedness matrix K_i. These frequent computations are much faster using the right hand side of (2.4), which costs $O(N^2P)$, than the left hand side, which costs $O(N^2P^2)$. (In fact, the $O(N^2P)$ cost can be further reduced using problem-specific structure, but the identity in (2.4) is general.)

2.3 The GLMM decomposition

The GLMM decomposition applies to tensor products of the form

$$\sum_{m=0}^{M} C_m \otimes K_m$$

This covariance structure arises in the general linear mixed models in Section 4 and 5 and generalizes many related structures used in this thesis. Here, M is general but, in later mixed model sections, will correspond to the number of random effects (other than pure environmental noise).

Informally, all difficult GLMM computations (and the closely related computations for phenix and CMMs) boil down to manipulations of the overall trait precision matrix Λ_y. Λ_y is related to the above tensor product because of the matrix normal random effects:

$$Y \sim \sum_{m=0}^{M} \mathcal{MN}(0, K_m, C_m) \implies \mathbb{V}(y)^{-1} := \Lambda_y = \left(\sum_m C_m \otimes K_m \right)^{-1}$$

The compact GLMM decomposition–performed efficiently using the below Proposition–enables a lower-complexity representation of this Λ_y matrix which, in turn, expedites inference in the crucial steps in fitting the GLMM. Previous computational advances in fitting mixed models have revolved around efficient representations of the Λ_y matrix in special cases, i.e. the univariate case where the C_m are simply scalars and the Kronecker product is simply a scalar product, or the single-random-effect case where VCs other than C_0 and C_1 are omitted.

The GLMM decomposition is essential for the GLMMs discussed in Chapter 4 and, thus, the GLMM prior used by CMMs in Chapter 5. Further, special cases of this result are used in Chapter 3, though the simplified phenix model does not partition heritability and does not need the full generality of this result.

An example of this decomposition's utility can be seen in Proposition 2, which performs simple manipulations of the GLMM decomposition to derive an algorithm to efficiently multiply any long vector (i.e. length NP) with Λ_y.

2.3.1 Key assumption

The motivating assumption is that the $K_i \in \mathbb{S}_+^N$, called relatedness or kinship matrices, are all known. Typically, $K_0 = I$, giving idiosyncratic noise, and K_1, the genomic relatedness matrix (GRM), represents aggregate, full-rank relationships among the N samples. The GRM can capture either close family relationships, population structure, subtle and small variations in similarity amongst nominally unrelated samples or, typically, some combination of these factors. See Section 5.2.4 for discussion of modified GRMs.

For $m > 1$, K_m typically represents an interesting set of genetic variants, sharing some *a priori* characteristic, like location in a gene or annotation, or grouped adaptively via some model selection. More generally, if G_m is the incidence matrix of a random effect, $K_m \propto G_m G_m^T$ represents the relatedness between samples induced by G_m. This factorization, assumed known *a priori*, decreases memory and runtime burdens (under Assumption 1).

Formally, motivated by the above considerations, we assume that all but K_0 and K_1 are collectively low-rank.

Assumption 1. *Define* $r_m := rank(K_m)$. *Then I assume*

$$r := \sum_{m>1} r_m < N$$

If each K_m is built from single nucleotide polymorphisms (SNPs), r is simply the total number of SNPs used (modulo perfect LD); in particular, r does not

depend on M, the number of components into which the r SNPs are divided.

Much work has been done to derive efficient implementations for various GLMM sub-models, and these implementations typically require a similar assumption. This began with [Listgarten et al., 2013], which studied $M = 2$ and $P = 1$ and derived $O(r^2N)$ inference by working in r-dimensional space; the same idea is used in [Casale et al., 2015], which generalizes the $M = 2$ approach to arbitrary P at a cost of $O(r^2NP^3)$. Generalizing to $M > 2$ (while fixing $P = 1$) and defining r to be the sum of all low-rank G matrices, [Speed and Balding, 2014] derived an algorithm again obtaining $O(r^2N)$ cost–as cheap as if all r SNPs were combined into one random effect. As in this prior work, we make Assumption 1 to ensure that low-rank matrix operations provide a genuine speedup –i.e. are truly low rank–as otherwise the above complexities collapse to the naive $O(N^3P^3)$ cost.

While our setting is strictly more general–allowing generic M and P–it may be that Assumption 1 is less palatable for large P. In [Speed and Balding, 2014], for example, the goal is to detect the r SNPs scattered across M regions that contribute to trait variation–by this definition, r and M can only grow with the number of considered traits. In this sense, the appropriate choices for r and M increase implicitly with P, arguably belying many of the complexities stated throughout this thesis.

Nonetheless, correlated traits will likely share many active genomic regions, especially for high-throughput phenotypes like gene expression or medical images. Moreover, restricting to computationally feasible r may present a bias-variance tradeoff–the added precision from using multitrait models to test pleiotropic loci may at least partially outweigh the underfitting cost of fitting only small r.

In another application, regions are chosen not adaptively but based on *a priori*

segmentation of the genome. Specifically, [Casale et al., 2015] uses $M = 2$ and tests the r SNPs in a given gene. This choice of r is clearly independent of P and, thus, Assumption 1 is no less tenable for large P. More generally, if the M genomic regions are defined *a priori*–e.g. by genes or annotations–or the low-rank VCs correspond to non-genetic confounders–like the family and population structure considered in Section 4.4.1–Assumption 1 remains reasonable as P grows.

2.3.2 Derivation

This section combines a known simultaneous diagonalization idea from multi-trait methods [Meyer and Kirkpatrick, 2010; Zhou and Stephens, 2014; Lippert et al., 2014; Rakitsch et al., 2013; Dahl et al., 2013] with a known low-rank idea from multi-effect methods [Speed and Balding, 2014]. The simultaneous diagonalization is applied to the sum of two KPs (the polygenic and environmental VCs), yielding the product of KPs and a diagonal matrix; simultaneously diagonalizing more than two KPs is not generally possible. The low-rank idea aggregates low-rank matrices and then applies Woodbury (as in [Speed and Balding, 2014], generalizing the single low-rank matrix appearing in [Lippert et al., 2011; Listgarten et al., 2013; Casale et al., 2015]).

The goal of the below proposition is to unify these two key linear algebra results, and thus unify multi-effect and multi-trait methods.

Proposition 1. *Let* $\Lambda_y := \left(\sum_{m=0}^{M} C_m \otimes K_m\right)^{-1}$. *Let* r *be as in Assumption 1, and let* R *be the rank of the matrix* $\sum_{m>1} C_m \otimes K_m$. *Then the GLMM decomposition can be performed efficiently: that is, it costs* $O(N^3 + P^3 + NPR^2)$ *time to compute* (T, Ξ, Ψ) *such that*

$$\Lambda_y = (T \otimes Q_{K_1}) \left[\Xi - \Psi\Psi^T\right] (T \otimes Q_{K_1})^T \tag{2.5}$$

Further, if K_1 is omitted or Q_{K_1} is given, the $O(N^3)$ term drops out, and if $\{G_m\}_m > 1$ are omitted (so $M = 1$), setting $R = 0$ gives the correct complexity. Finally, if all C_m are full rank, then $R = rP$ and the complexity becomes $O(N^3 + Nr^2P^3)$.

Proof. Let $T \in \mathbb{R}^{P\times P}$ be invertible, and define $K'_m := Q_{K_1}^T K_m Q_{K_1}$ for $m > 1$. Since $\sum_{m>1} C_m \otimes K_m$ has rank R by Assumption 1, there exists some $U \in \mathbb{R}^{NP\times R}$

$$\sum_{m>1} [T^T C_m T] \otimes K'_m = UU^T$$

Without Assumption 1, this representation achieves no compression and subsequent steps are $O(N^3P^3)$, the same cost as directly computing Λ_y.

This square-root can be used to simplify Λ_y:

$$\begin{aligned}
\Lambda_y &:= \left(C_0 \otimes I + C_1 \otimes K_1 + \sum_{m>1} C_m \otimes K_m\right)^{-1} \\
&= (T \otimes Q_{K_1}) \left(\underbrace{[T^T C_0 T] \otimes I + [T^T C_1 T] \otimes \Lambda_{K_1}}_{\Xi^{-1}} + \underbrace{\sum_{m>1} [T^T C_m T] \otimes K'_m}_{UU^T} \right)^{-1} (T \otimes Q_{K_1})^T \\
&= (T \otimes Q_{K_1}) \left[\Xi - \Psi\Psi^T\right] (T \otimes Q_{K_1})^T
\end{aligned}$$

using Woodbury on $\left(\Xi^{-1} + UIU^T\right)^{-1}$ in the last line and defining $\Psi = \Xi U^T \left(I + U^T \Xi U\right)^{-1/2}$.

To obtain diagonal Ξ and simplify computation, I use a factorization from [Meyer and Kirkpatrick, 2010]: let $T := (C_0 + C_1)^{-1/2} Q_\Delta$, where $\Delta := (C_0 +$

$C_1)^{-1/2}C_1(C_0+C_1)^{-1/2}$, so that

$$\begin{aligned}
\Xi^{-1} &= [T^T C_0 T] \otimes I + [T^T C_1 T] \otimes \Lambda_{K_1} \\
&= [T^T (C_0 + C_1) T] \otimes I + [T^T C_1 T] \otimes (\Lambda_{K_1} - I) \\
&= [Q_\Delta^T Q_\Delta] \otimes I + [Q_\Delta^T \underbrace{(C_0 + C_1)^{-1/2} C_1 (C_0 + C_1)^{-1/2}}_{\Delta} Q_\Delta] \otimes (\Lambda_{K_1} - I) \\
&= I + \Lambda_\Delta \otimes [\Lambda_{K_1} - I]
\end{aligned}$$

Despite the minus sign, this matrix is PSD–as it must be–because $I \succeq \Delta$.

This choice for T requires that $C_0 + C_1$ is full-rank; since both are assumed PSD, this automatically holds if either is full-rank. T can also be defined replacing $C_0 + C_1$ with either C_0 or C_1, as in [Dahl et al., 2013; Rakitsch et al., 2013; Zhou and Stephens, 2014], but this is sub-optimal as the resulting inverse exists less generally. Q_Λ, and hence T, is non-unique for low-rank C_1, but any choice suffices.

Overall, T and the eigendecomposition of Δ can be evaluated in $O(P^3)$. The eigendecomposition of K_1 costs $O(N^3)$, and forming the whitened incidence matrices $\{G'_m := Q_{K_1}^T G_m\}_{m>1}$ costs $O(N^2 r)$, which is negligible (evaluating K'_m for $m > 1$ is not necessary if U is computed by the lazy Cholesky decomposition described below). If the eigendecomposition of K_1 and all $\{G'_m\}_{m>1}$ matrices are given (this latter assumption is implicit in the proposition statement, but, in practice, will hold when the former does), these steps can be skipped. The matrix multiplications to form Ψ from U and, by Section 2.4, the Cholesky decomposition to find U both cost $O(NPR^2)$, which can also be skipped if no low-rank incidence matrices are provided (i.e. $M = 1$).

□

This decomposition enables efficient low-level operations like multiplication and

partial-traces–the vital ingredients for the mixed models discussed in Chapters 4 and 5. To illustrate the first, define $\psi_0 := \left(\Xi - \Psi\Psi^T\right) \text{vec}\left(Q_{K_1}^T Y T\right)$, so that

$$\begin{aligned}
\Lambda_y y &= (T \otimes Q_{K_1}) \left[\Xi - \Psi\Psi^T\right] (T \otimes Q_{K_1})^T y \\
&= (T \otimes Q_{K_1}) \left[\Xi - \Psi\Psi^T\right] \text{vec}\left(Q_{K_1}^T Y T\right) \\
&= (T \otimes Q_{K_1}) \psi_0 \implies \\
\text{mat}\left(\Lambda_y y\right) &= Q_{K_1} \text{mat}\left(\psi_0\right) T^T
\end{aligned}$$

ψ_0 can be efficiently computed by iteratively performing simple matrix multiplications:

- $X \leftarrow Q_{K_1}^T Y T$ costing $O(N^2 P + N P^2)$
- $\text{mat}\left(\psi_0\right) \leftarrow \text{mat}\left(\Xi\right) * X - \text{mat}\left(\Psi(\Psi^T x)\right)$ costing $O(NPR)$

This proves:

Proposition 2. *The cost of evaluating ψ_0 for any $Y \in \mathbb{R}^{N \times P}$, given the GLMM decomposition of Λ_y and $Y' := Q_{K_1}^T Y$ (taking $Q_{K_1} = I$ if K_1 is omitted), is $O(NP^2 + NPR)$).*

This is crucial for likelihood computations in Chapters 4 and 5 as a means of computing the quadratic term, i.e. $y^T \Lambda_y y^T$. More generally, this Proposition allows efficient computations of the form $\Lambda_y x$, which arise often when fitting and manipulating GLMMs, e.g. when predicting or imputing phenotypes, estimating fixed effects, or simply computing likelihood gradients. This Proposition is one of the basic tools used in this thesis to rewrite equations in a form facilitating efficient computation.

2.4 Lazy Cholesky decomposition

The GLMM decomposition requires the Cholesky decomposition of an $NP \times NP$ matrix with rank R:

$$X := \sum_{m>1} C_m \otimes K_m$$

Standard software for Cholesky decomposition (with pivoting) solves this problem in $O(R^2NP)$; however, this elides the cost of forming X, which is $O(N^2P^2)$ memory and time.

To circumvent this overhead, I implemented a Cholesky decomposition which lazily evaluates entries of X as needed. The `R/Rcpp` implementation [Eddelbuettel and François, 2011; Eddelbuettel, 2013] simply performs a standard Cholesky decomposition with pivoting [Harbrecht, Peters, and Schneider, 2012] with one exception: a standard implementation computes X upfront and reads off entries as needed, while mine computes each entry X_{ij} on-the-fly. Because far fewer than all N^2P^2 entries of X are needed for the incomplete Cholesky decomposition, this saves considerable time and memory. Other efficient decomposition ideas–especially the Nystrom method, which uses columns of X as a basis for low-rank representation–could presumably incorporate the same lazy-evaluation trick. Further, this lazy Cholesky decomposition applies to any low-rank element of the tensor product $\mathbb{S}_+^P \otimes \mathbb{S}_+^N$ (though I am unaware of other useful applications).

This basic approach has long been advocated as an easy way to feed problem specific structure into extremely low-level and highly-optimized operations [Wright, 1999]. This idea has been used to scale to large-N learning problems [Fine and Scheinberg, 2001; Cristianini, Shawe-Taylor, and Lodhi, 2002; Bach and Jordan, 2003; Zeileis et al., 2004], though previous focus has been on approximat-

ing nearly low-rank kernel matrices. The present situation is even more structured as X is exactly low-rank, but the approximate approach suggests Assumption 1 could be relaxed at the cost of inexact, but still cheap and perhaps sometimes approximately correct, operations.

Figure 2.1 shows the computational savings of the lazy Cholesky decomposition compared to `R`'s highly optimized, general-purpose Cholesky decomposition[1]. As expected, for small parameter sizes my simplistic implementation is slower than the sophisticated default implementation in `R`. For larger parameter sizes, though, the lower complexity of my approach becomes evident and, eventually, lazy Cholesky becomes many orders of magnitude cheaper than the `R` default.

More specifically, when $M = 2$, $r_2 = r = 10$ SNPs are modeled, $P = 10$ and full-rank VCs are used (as in the left panel of Figure 2.1), the two Cholesky approaches are comparable–and cheap–for N less than roughly 1,000. But even once N is 2,000, the lazy Cholesky approach is ten times faster, and the generic `R` Cholesky decomposition becomes prohibitively expensive before N reaches 10,000.

Conversely, the lazy Cholesky decomposition scales poorly with respect to R, which is appropriate as the lazy approach only benefits by exploiting low-rankness, i.e. the fact that $R < NP$. When $N = 2,500$ and $P = 10$ (as in the right panel of Figure 2.1), the lazy Cholesky decomposition provides a substantial benefit only for less than 50 SNPs (i.e. $R < 500$) and becomes more expensive than the `R` default for as few as 100 SNPs, even though 100 SNPs still gives a very low-rank model.

[1]This was performed on a server with roughly 400 GB RAM. Many of the $NP \times NP$ matrices passed to `R`'s Cholesky function to generate Figure 2.1 could not be stored loaded into memory on typical computers; the lazy Cholesky, however, creates matrices no larger than $NP \times R$.

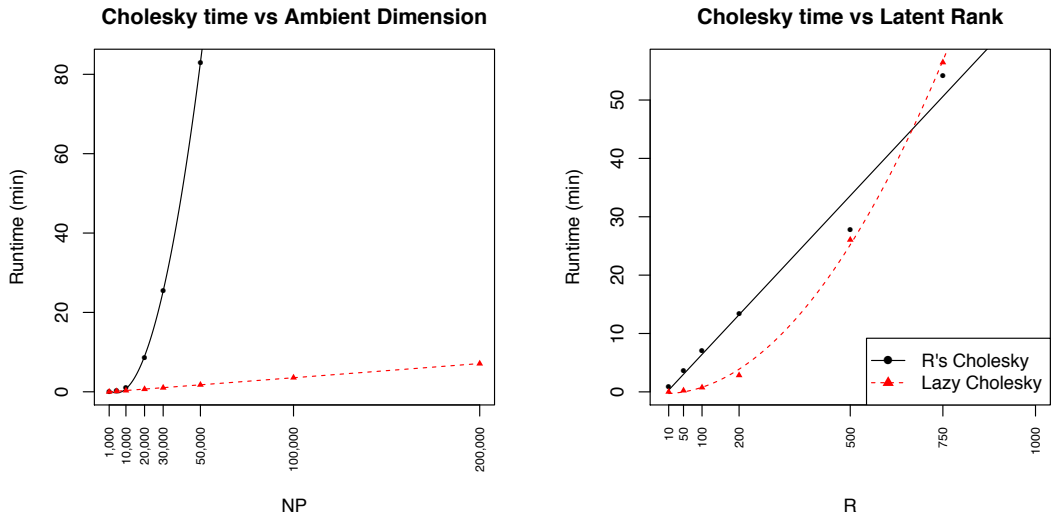


Figure 2.1: Comparison of Cholesky implementations. Solid lines build X then use `R`'s `chol` function; dotted lines use new code that builds X lazily. The left panel varies N and fixes $P = 10$, $r = 10$ and $R = rP$; the right panel varies r and, implicitly, $R := rP$ and fixes $N = 2,500$ and $P = 10$. As runtime depends on N, P and r only through NP and R, the specific choices for N, P and r are irrelevant.

The lazy Cholesky decomposition is essential to deriving the claimed computational complexity of the GLMM decomposition. If a base Cholesky algorithm were used instead, the GLMM decomposition (and thus fitting the GLMM) would increase in cost to $O(N^2P^2)$ simply by forming the input to the base Cholesky decomposition. Except as an (crucial) ingredient in the GLMM decomposition, however, the lazy Cholesky is not directly used in this thesis.

Chapter 3

Phenotype imputation with phenix

This section closely follows the supplement from [Dahl et al., 2016].

3.1 Phenotype matrices and missing data

Missing data is prevalent for high-throughput, multiple-phenotype measurements, which can sporadically fail and/or require QC to remove "bad" observations. A common answer to such missing observations is to use imputed data downstream [Little and Rubin, 1987]. This approach entirely separates the imputation and analysis tasks, which is a simplification with both upsides and downsides. The main benefit is ease for the statistician, as standard analyses–which typically require fully observed data–can be performed without modification. But simply analyzing the imputed data as if it were observed fails to account for the added uncertainty in imputed values, which can result in inaccurate downstream conclusions.

Genotype imputation [Marchini et al., 2007; Howie, Donnelly, and Marchini, 2009; Marchini and Howie, 2010], in particular, has been an invaluable imputation

tool for the genetics community. Leveraging extremely rich reference data and highly constrained, HMM-based likelihood functions, genotype imputation can provide dramatic boosts of power and refine signals to narrower genomic windows. Nonetheless, the standard single-imputation approach is not generally calibrated and imputed values must be reasonably accurate (as defined by some metric; see Section 3.6.4).

We intend to develop a similar imputation approach for phenotypes: given a phenotype matrix with missing entries, we will return a full phenotype matrix for downstream applications. The modelling approach differs drastically from genotype imputation: genotypes have known linear marker ordering, ternary SNP values, massive reference data and a thoroughly characterized recombination model; phenotypes, by contrast, can in principle be anything. Thus, genotype imputation algorithms are extremely problem-specific and informative, but phenotype imputation algorithms must be more generic and, typically, will have worse imputation accuracy.

Nonetheless, there are many situations where phenotype imputation is useful. For example, tightly correlated traits can be imputed well, which seems to drive (at least) one of the new GWAS hits obtained with phenotype imputation in Section 3.6.5. Related, expensive traits can be probed through many cheap proxy traits, and this may dramatically improve power for a fixed budget.

In fact, in simple settings, the power of such proxy phenotype studies can be computed analytically. Specifically, [Hormozdiari et al., 2016] studies a model with a single missing phenotype, a single causal genetic marker (that directly affects only the primary, unphenotyped trait) and no population structure/relatedness. Further, they assume traits are jointly multivariate normal and impute with what

we call MVN in Section 3.4. Under these assumptions, an optimal imputation and association testing approach can be developed, and the power of the resulting test can be calculated analytically, greatly informing study design considerations.

Though phenotype imputation may be valuable for downstream single-trait analyses, it is invaluable for multi-trait analyses. To our knowledge, all multi-phenotype approaches require fully observed data matrices [Huffman et al., 2011; Zhou and Stephens, 2012; Lauc et al., 2010; O'Reilly et al., 2012; Dahl et al., 2013; Lippert et al., 2014; Loh et al., 2015b]. When faced with missing data, most approaches perform case-wise deletion (CWD)–every person with even one missing trait is removed. While potentially acceptable for a small number of traits, the CWD cost grows exponentially in the number of traits: if each trait is independently observed with probability θ, the probability a person is observed on all P traits is θ^P. In three of the six real datasets studied in Section 3.5.3, for example, CWD throws away the entire dataset.

Missing phenotypes have been discussed before. Though the standard approach is CWD [Hai et al., 2012; Piccolo et al., 2009; Choi, Chowdhury, and Swaminathan, 2014; Scutari et al., 2014; Park, Lee, and Kim, 2011], some have applied generic imputation methods [Shin et al., 2014; Suhre et al., 2011; Meuwissen et al., 2014], sometimes combined with CWD [Cui et al., 2009]. While generic imputation is typically better than throwing away data, it is unclear which generic algorithms work best. Moreover, no generic algorithm can be optimal: we have genetic data that we know is useful. Although the resulting inter-sample relatedness is typically less informative than inter-trait correlations–especially for nominally unrelated humans–incorporating it respects the genetic structure of the data, is always at least slightly helpful and can be crucial.

This chapter introduces phenix, a phenotype imputation method leveraging both genetic and phenotype relationships. phenix is based on a Bayesian, low-rank variant of the standard multi-phenotype mixed model (MPMM), introduced in Section 3.2. The model is fit with variational Bayes (reviewed in generality in A.6), and the algorithm is derived in Section 3.3. Section 3.4 reviews alternative methods for matrix completion, including single-trait models from genetics and genetics-ignorant matrix completion approaches. Section Section 3.5 then shows that phenix outperforms these alternatives in imputation accuracy in extensive simulations and six real datasets spanning multiple organisms, numbers of traits, and missingness levels. We then use phenix for GWAS in Section 3.6, validating the approach in simulations and then applying it to a rat dataset with 140 traits [Sequencing and Consortium, 2013] to find novel, plausible GWAS loci. Finally, Section 3.7 concludes by discussing possible future extensions.

The `R` package is available at

`https://mathgen.stats.ox.ac.uk/genetics_software/phenix/phenix.html`

Definitions and notation

We use m and o to denote the missing and observed entries of a vector, respectively. When discussing a row n or a column p, we also refer to m_n/o_n and to m_p/o_p, the missing and observed parts of the n-th row and p-th column, respectively.

3.2 The phenix model

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 standard-

ized to unit variance. We start with the additive model

$$Y = U + \epsilon \tag{3.1}$$

where U represents the aggregate genetic contribution to phenotypic variance and ϵ is idiosyncratic noise.

One standard parameterization of this additive model is the multiphenotype mixed model (MPMM), described in detail in Section 3.4 and repeated here for reference:

$$\begin{array}{rcl} Y & = & U + \epsilon \\ U & \sim & \mathcal{MN}(0, K, B) \\ \epsilon & \sim & \mathcal{MN}(0, I, E) \end{array} \tag{3.2}$$

$K \in \mathbb{S}_+^N$ is the kinship matrix between individuals in the sample (see Section 1.4), which we assume is known from pedigree or genotype data [Segura et al., 2012; Kang et al., 2008; Zhao et al., 2007; Weir, Anderson, and Hepler, 2006]; B and E are the unknown genetic and environmental covariance matrices in $\mathbb{S}_+^P$. Many have recently developed "efficient" inference for these models applied to genetic data [Zhou and Stephens, 2014; Korte et al., 2012; Dahl et al., 2013; Stegle et al., 2010]. MPMMs arise naturally as a multiphenotype generalization of the typical univariate linear mixed model (LMM): when B and E are diagonal in (3.2), the MPMM reduces to P independent LMMs. MPMMs are a special case of the GLMMs discussed in Section 4, which incorporate penalization and additional random effects.

Unfortunately, MPMMs (and GLMMs) can handle only a few phenotypes, roughly dozens if the model is fit once (Section 4.3.6) or 10 in GWAS [Zhou and Stephens, 2014]: as P grows, covariance MLEs quickly become statistically and computationally intractable. Moreover, missing observations plague MPMMs, as

inference relies on KP covariance structures not (generally) inherited by the observed part of the phenotypes. Nonetheless, MPMMs can naturally incorporate entirely-missing samples, as the observed part of the matrix has no missing data–such missingness patterns occur when predicting phenotypes out-of-sample [Rakitsch et al., 2013] (see Section 4.3.4) or, more generally, after performing CWD [Zhou and Stephens, 2014], though this latter approach incurs costs of CWD.

To simultaneously address these limitations, we develop an alternative multi-phenotype generalization of LMMs[1] by using a Bayesian, low-rank matrix factorization model for the genetic term U. Such low rank models are computationally tractable and often biologically plausible: U (approximately) has low-rank M when the P observed phenotypes share a simple biological structure summarized (mostly) by M latent factors. Below, the model is first introduced in the case where only two traits are observed, and then the model is subsequently generalized to an arbitrary number of traits.

The case with only two traits

For now, assume there are $P = 2$ traits and that we are fitting a rank-one approximation, so that $M = 1$. Then the bivariate phenix model becomes

$$\begin{array}{rcc} Y|S,\beta,\epsilon & \sim & U+\epsilon \\ U_{ip} & = & S_i\beta_p \\ S & \sim & \mathcal{N}(0,K) \\ \beta_p & \overset{\text{iid}}{\sim} & \mathcal{N}(0,\tau^{-1}) \\ \epsilon_{ip}|\Sigma & \overset{\text{ind}}{\sim} & \mathcal{N}\left(0,\sigma^2_{pp}\right) \\ (\epsilon_{i1},\epsilon_{j2})\,|\Sigma & \sim & \mathcal{N}(0,I\{i=j\}\Sigma) \\ \Sigma = \begin{pmatrix} \sigma^2_{11} & \sigma_{12} \\ \sigma_{12} & \sigma^2_{22} \end{pmatrix} & \sim & \text{Inverse-Wishart}(e,I_2) \end{array} \tag{3.3}$$

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

The model for Y is a standard additive model, where genetics (U) and environment (ϵ) sum to define the phenotype; this rules out gene-environment interactions. The model forces U to be low-rank matrix (in this special bivariate case, a rank 1 matrix), corresponding to the assumption that the heritable phenotype component comes (mostly) from a single low-dimensional, latent biological process. The model on S dictates the belief that this latent biological factors is entirely genetically determined, and the generic distribution on β flexibly models how the latent factor S propagate into the observed traits. Finally, ϵ captures all environmental noise, which we anticipate to be unstructured–motivating the full-rank matrix normal prior–but with potentially complicated correlations across traits–motivating the inverse-Wishart prior on Σ that allows the model to learn environmental correlations.

Our additive decomposition of Y and matrix-normal prior on ϵ are standard in the field [Stegle et al., 2010; Korte et al., 2012; Zhou and Stephens, 2014], though the conjugate Wishart prior on the environmental precision matrix is a simple Bayesian modification of the typical frequentist approach that learns environmental correlations with (RE)ML. The low-rank model on U is newer, although others have used approximately-low-rank decompositions to model the genetic [Campos and Gianola, 2007] or the genetic and environmental [Runcie and Mukherjee, 2013] random effects. Another approximately low-rank approach takes the standard MPMM but fits low-rank VCs (plus a spherical offset) [Stegle et al., 2010; Rakitsch et al., 2013]; the distinction between these notions of random effect low-rankness is discussed further in Section 5.2.4.

The general case

The phenix model can now be generalized to arbitrary M and P. The major changes are replacing the multivariate normal on S–as before it had only $M = 1$ column–with a matrix-variate normal, and also that each component of β is now allowed to have it's own variance, dictated by diagonal entries of D:

$$\begin{array}{rcl} Y|S,\beta,\epsilon & \sim & U+\epsilon \\ U & = & S\beta \\ S & \sim & \mathcal{MN}\left(0,K,I_M\right) \\ \beta & \sim & \mathcal{MN}\left(0,D,B\right), \\ \epsilon|\Lambda_\epsilon & \sim & \mathcal{MN}\left(0,I,\Lambda_\epsilon^{-1}\right) \\ \Lambda_\epsilon & \sim & \text{Wishart}(e,E) \end{array} \tag{3.4}$$

If D is allowed to be an arbitrary diagonal matrix[2], then the matrix factorization model in (3.4) is equivalent to reduced-rank regression in the same sense that MPMM and LMM are equivalent to genome-wide linear regression (proof not shown). In this sense, the model is as natural an LMM generalization as the more traditional MPMM.

For simplicity, we here set $D = I_M$, $B = (\tau I_P)^{-1}$, $e = P + 5$ and $E = e^{-1}I_P$ (so that $\mathbb{E}\left(\Lambda_\epsilon\right) = I_P$), even though more general choices are possible. Similarly, although τ can be given a prior, chosen to maximize the (approximate) marginal likelihood, or tuned by cross-validation, we simply use the improper (or objective) $\tau = 0$. As a final simplification, we set $M = \min(N, P)$. All analyses in this thesis use these choices (though the package allows others), and phenix is tuning-free.

These minimally regularizing choices for τ and M may seem prone to overfit, but two surprising facts (discussed in Section 3.2.2 and 3.2.3) conspire to ensure that underfitting is the more significant concern. Moreover, cross-validation ap-

[2]Due to scaling and rotation non-identifiability of the product $S\beta$, D can be assumed diagonal without loss of generality.

plied to the NSPHS dataset (described below) consistently chose $\tau = 0$. More generally, I have not seen phenix overfitting (and consistently see it underfit).

3.2.1 Relation to matrix factorization models

A closely related method is probabilistic matrix factorization (PMF). PMF is the special case of our phenix model (3.4) obtained by setting $K = I_N$ and $E \propto I_P$. In such models, VB is an established alternative to computationally expensive MCMC and overfitting-prone maximum *a posteriori* despite its approximate nature [Salakhutdinov and Mnih, 2008; Ilin and Raiko, 2010; Lim and Teh, 2007].

Another set of competitive matrix completion methods, starting from [Candès and Recht, 2009], are based on spectral regularization [Recht, 2011; Candès and Tao, 2010; Recht, Xu, and Hassibi, 2008]; improving recovery guarantees, deriving new optimization algorithms and generalizing to different penalties are all active areas of research. PMF is more closely related to spectral regularization than it may appear, as[3]

$$||X||_* = \frac{1}{2}\left(\min_{AB'=X} ||A||_F^2 + ||B||_F^2\right) \tag{3.5}$$

The left side of (3.5) is the nuclear norm, and the right side represents (an optimized solution to) a combination of ℓ_2-penalties. The ℓ_2 penalties are, roughly, those induced by the PMF matrix normal priors, and (3.5) suggests that maximization approaches (e.g. MAP and VB) to PMF will be similar to nuclear-norm-penalized approaches. This is formalized in [Nakajima et al., 2013] (see also [Kim and Choi, 2013]) and detailed in Section 3.2.3.

Note that none of these established methods incorporates non-spherical priors.

[3] I have not seen a proof of this fact–known since at least [Srebro, 2004]–and so give one in Section A.3.

But K is the central element of LMMs in genetics (and random effect models generally). Moreover, by comparing to a competitive spectral-regularization algorithm [Mazumder, Hastie, and Tibshirani, 2010; Hastie et al., 2015] (see Section 3.4), our simulations and real data analyses suggest incorporating K is always beneficial–and sometimes vital.

3.2.2 Model induced regularization

An unusual property of the model (3.4), unrelated to variational approximation, is so-called model induced regularization (MIR) [Nakajima and Sugiyama, 2011]. MIR appears because of non-identifiability in the priors and likelihood for (S, β). MIR means even when each (S, β) pair is equally probable–as achieved with $\tau = 0$–it is not true that each $U := S\beta$ is equally probable: if $f(S, \beta) := S\beta$, the probability of U becomes the measure of the pre-image $f^{-1}(U)$, which shrinks as $\|U\|$ grows.

Roughly, this means flat priors on S and β induce regularizing priors on U; equivalently, a flat prior on U can be obtained only from a *convex* prior on (S, β). As $\tau \to 0$, the concave, normal priors on S and β in our model become flatter and thus closer to the concave priors required for unregularized inference. This explains why choosing small τ minimizes shrinkage, but also why even $\tau = 0$ cannot eliminate shrinkage.

More formally:

Proposition 3. *Let*

$$Y \sim \mathcal{MN}(S\beta, I, I) \tag{3.6}$$

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

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

[Nakajima and Sugiyama, 2011] proved this for $N = M = P = 1$ using a clever Jeffreys prior argument: Jeffreys priors are invariant under transformation; the Jeffrey prior on Y is uniform; therefore the Jeffreys prior on (S, β) induces a flat prior on Y. Our proof expands exactly this argument to general N, M and P; the computations are straightforward but lengthy and left to Section A.2.

3.2.3 Automatic rank selection

The VB implementation of PMF, called VBMF, has yet another surprising property: the rank of the fitted $U := S\beta$ often has rank strictly lower than M, the almost-sure rank of U under both the prior and the (exact) posterior [Nakajima and Sugiyama, 2011; Nakajima et al., 2013]. The proof–which assumes the simple PMF model, known noise level and fully-observed data–relies on analytically characterizing the solution to VBMF. (As those authors note, these equations do not easily generalize to our setting.) The derivation shows that $\hat{U}$, the mean and mode for U under the VB posterior, is a spectrally-regularized version on Y (as mentioned above).

In particular, with $M = P$ and spherical noise level σ^2,

$$\sigma_i(\hat{U}) \xrightarrow{\tau \to 0} \max\left(\left(1 - \frac{N\sigma^2}{\sigma_i(Y)^2}\right)\sigma_i(Y), 0\right)$$

where $\sigma_i(\cdot)$ is the i-th singular value. That is, $\hat{U}$ keeps the singular vectors of Y but shrinks the singular values to their positive part James-Stein estimators. Because this shrinkage maps nonzero singular values to 0 (any smaller than $\sqrt{N\sigma^2}$), it automatically induces low-rankness.

Moreover, this rank determination has desirable asymptotic properties, so long as (the analogue of) τ is chosen properly [Nakajima et al., 2012; Nakajima, Tomioka, and Sugiyama, 2015]. Thus, continuous choice of τ reliably relaxes rank selection in much the way ℓ_1 penalization relaxes support selection.

Though these properties have not been proven in our more general context, we assume analogues apply as phenix consistently fits automatically-low-rank $\hat{U}$. We thus set $M = \min(N, P)$ and allow phenix to determine the rank of the putatively low-rank U. Though [Nakajima et al., 2012; Nakajima, Tomioka, and Sugiyama, 2015] suggests choice of τ is crucial, this is for an asymptotic setting with both ambient dimensions growing, which starkly contrasts with typical phenix applications where $N \gg P$ and $P \ll 1,000$.

Setting $M = \min(N, P)$ is infeasible for large P–the full-rank first VB iteration, alone, will be prohibitive–but is not problematic in the below applications. Moreover, phenix appears to avoid too-low-rank local optima that generally plague PMF [Ilin and Raiko, 2010], presumably by initializing with full-rank.

3.3 A variational Bayes implementation

3.3.1 Our variational approximation to the posterior

Our standard variational approximation is that S, β, Λ_ϵ and Y^m are independent in the posterior. The goal is then to find the best approximation to the posterior among these functions (in Kullback-Leibler divergence). Section 3.3.2 shows that

the resulting approximate marginals 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 distributions thus reduces to optimizing the above variational parameters.

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 the approximate posterior–they do have a simple structure that admits low-complexity computations. If N_m is the number of unique missingness patterns among samples, phenix costs $O(N_m P^3 + NP^2 + N^2 M)$ for each VB iteration; additionally, a one-off $O(N^3)$ eigendecomposition of K is performed at the outset.

3.3.2 The parametric forms of the approximate posterior marginals

In the next sections, I apply the standard VB identity (equation (A.12)) to each parameter block in turn:

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

I iterate through these updates until convergence, defined as the marginal likelihood relatively changing by less than 10^{-8} or 1,000 loops over all parameter updates (less stringent criteria can be used without noticeable loss in accuracy if runtime is important).

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

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

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:

$$Y_{i,m_i}|Y_{i,o_i} \overset{\text{ind}}{\sim} \mathcal{N}\left(\mu^Y_{i,m_i}, \Sigma^{Y_i}\right)$$

$$\mu^Y_{i,m_i} = (\mu_S\mu_\beta)_{i,m_i} + (\Omega^{-1})_{m_i,o_i}\left(\Omega^{-1}\right)_{o_i,o_i}(Y_{i,o_i} - (\mu_s\mu_\beta)_{i,o_i})^T \tag{3.7}$$

$$\Sigma^{Y_i} = (\Omega_{m_i,m_i})^{-1} \tag{3.8}$$

Updating μ^Y_{i,m_i} and Σ^{Y_i} for each i costs $O(NP^3)$. But, since the $O(P^3)$ operations depend on i only through o_i, the complexity can be reduced to $O(NP^2 + N_mP^3)$. In real datasets, where experimental and observational constraints often induce structured missingness patterns, N_m is often much smaller than N: for example, in the chicken data, $N = 11,575$ but $N_m = 36$. Nonetheless, this improvement is not crucial as this step typically comprises only a small fraction of overall runtime.

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

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

giving the updates

$$\Lambda_b = \Omega_\beta \otimes V_S + \tau I \qquad \text{(implicit)}$$

$$\mu_b = \Lambda_b^{-1}\mathrm{vec}\left(\mu_S^T\mu_Y\Omega_\beta\right) \qquad (3.9)$$

$$\Omega_\beta = \Omega \qquad (3.10)$$

$$V_S = \mu_S^T\mu_S + \mathrm{tr}_P\left(\Lambda_s^{-1}\right) \qquad (3.11)$$

The partial trace in (3.11) can be efficiently evaluated using the expression for Λ_s below: using (2.2) to pull eigenvectors of V_β out of the partial trace and (2.3) to whiten K, only $O(M^3)$ multiplications and the partial trace of a diagonal matrix remain, which costs $O(M^3 + NM)$ rather than $O(N^3M^3)$. This proves

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

Similarly, Proposition 2 (with $R = 0$) computes (3.9) in $O(P^3 + MNP)$ rather than $O(M^3P^3 + MNP)$.

In both these computations, Λ_b is never actually created, and in place of Λ_b we store only V_S and Ω; copying the value of Ω is necessary (in theory, to guarantee the marginal likelihood improves) as I evaluate functionals of Λ_b after Ω has moved.

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

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

where

$$\Lambda_s = V_\beta \oplus K^{-1} \qquad \text{(implicit)}$$

$$\mu_s = \Lambda_s^{-1}\mathrm{vec}\left(\mu_Y\Omega\mu_\beta^T\right) \qquad (3.12)$$

$$V_\beta = \mu_\beta\Omega\mu_\beta^T + \mathrm{tr}_P\left(\left(\Omega\otimes I\right)\Lambda_b^{-1}\right) \qquad (3.13)$$

Again, Proposition 2 computes (3.12) in $O(N^2M)$ instead of $O(N^3M^3)$ and V_β is stored in place of Λ_s.

Unfortunately, $\mathrm{tr}_P\left(\left(\Omega\otimes I\right)\Lambda_b^{-1}\right)$ does not simplify when $\Omega \neq \Omega_\beta$. So I simply update β immediately before S, giving $\Omega = \Omega_\beta$ and

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

Now, Lemma 1 can compute (3.13) in $O(P^3)$ rather than $O(M^3P^3)$.

Equation (3.12) is the reason phenix has $O(N^2M)$ iterations while most mixed models only have one such step: if Y is complete, it need be whitened only once. The analogue here is storing a whitened version of the observed parts of Y. Let $Y_{ij}^0 = Y_{ij}$ if observed, $Y_{ij}^0 = 0$ otherwise. Then store

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

where Q are the eigenvectors of K, which costs $O(N^2P)$. Then at each iteration, $Q^T\mu_Y$ can be computed by

$$Q^T\mu_Y = Y' + Q^TY^1 \tag{3.14}$$

where $Y^1_{ij} = \mu^Y_{ij}$ if Y_{ij} is missing and 0 otherwise. If Y^1 has n_{miss} nonzero entries, the multiplication in (3.14) is $O(Nn_{miss})$, recovering linear-time iterations in N. This may be cheaper than $O(N^2M)$ but, if $n_{miss} \propto NP$, the $O(Nn_{miss})$ cost is only superficially linear in N; in fact, this cost may be greater than $O(N^2M)$ when $M << P$.

Equation (3.12) is also where low-rank kinship pays off: if $\text{rk}(K) = r$, this step becomes $O(NrM+P^2M)$ and the overall complexity drops from $O(N_mP^3+NP^2+N^2M)$ to $O(N_mP^3+NP^2+NrM)$; a similar logic applies to the one-off, low-rank eigendecomposition of K, which can be sped up to $O(rN^2)$. Though a substantial advantage for small P, large N–where N is tens or hundreds of thousands and P is tens–this trick is not pursued as it is insubstantial for our moderate-N applications.

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

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

$$\begin{aligned}
\Omega_0 &:= \mathbb{E}\left((Y - S\beta)^T(Y - S\beta)\right) \\
&= \mathbb{E}\left((Y - \mathbb{E}\left(S\beta\right))^T(Y - \mathbb{E}\left(S\beta\right))\right) + \mathbb{E}\left(\left(S\beta - \mathbb{E}\left(S\beta\right)\right)^T\left(S\beta - \mathbb{E}\left(S\beta\right)\right)\right) \\
&= (\mu_Y - \mu_S\mu_\beta)^T(\mu_Y - \mu_S\mu_\beta) + \sum_n \tilde{\Sigma}^{Y_n} && (3.15) \\
&\quad + \mu_\beta^T \text{tr}_P\left(\Lambda_s^{-1}\right)\mu_\beta + \text{tr}_P\left((I \otimes \left[\mu_S^T\mu_S + \text{tr}_P\left(\Lambda_s^{-1}\right)\right])\Lambda_b^{-1}\right) && (3.16)
\end{aligned}$$

If β has been updated more recently than S–which is trivially enforced–$V_S = \mu_S^T \mu_S + \mathrm{tr}_P\left(\Lambda_s^{-1}\right)$ and

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

Now Lemma 1 computes (3.16) in $O(P^3 + NM)$; (3.15) costs $O(NP^2)$ as written.

Finally, let $e' = e + N$, so

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

where

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

Initialization

First, I initialize μ_Y by imputing Y with MVN (see Section 3.4). This strategy is computationally cheap: if P is too large for MVN, it is too large for phenix. Ω is also initialized with MVN, using the MLE $\hat{\Sigma}$ for Ω^{-1}.

μ_S and μ_β are then initialized from the SVD of $\mu_Y = U_Y D_Y V_Y^T$:

$$\mu_S \leftarrow \left(U_Y (D_Y)^{1/2}\right)_{,1:M}; \quad \mu_\beta \leftarrow \left((D_Y)^{1/2} V_Y^T\right)_{1:M,}$$

where $1:M = (1,\ldots,M)$. Finally, we initialize $V_S = V_\beta = I_M$. Though simplistic, this reflects the component-wise orthogonality induced by our SVD initialization of μ_S and μ_β.

3.3.3 The marginal likelihood lower bound

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

$$\begin{aligned}
F(Q) &= \mathbb{E}_{\theta\sim Q}\left(\log P(Y^o,\theta) - \log Q(\theta)\right) \\
&= \mathbb{E}_Q\left(\log P(Y^o, Y^m, \beta, S, \Lambda_\epsilon) - \log Q(Y^m,\beta,S,\Lambda_\epsilon)\right) \\
&= \mathbb{E}_Q\left(\log P(Y|\beta,S,\Lambda_\epsilon) + \log P(\Lambda_\epsilon) - \log Q_Y(Y^m) - \log Q_\epsilon(\Lambda_\epsilon)\right) \quad (3.17) \\
&\quad + \mathbb{E}_Q\left(\log P(\beta) - \log Q_\beta(\beta)\right) \quad (3.18) \\
&\quad + \mathbb{E}_Q\left(\log P(S) - \log Q_S(S)\right) \quad (3.19)
\end{aligned}$$

We now compute each part:

$$\begin{aligned}
(3.17) &= 2\mathbb{E}_Q\left(\log P(Y|\beta,S,\Lambda_\epsilon) + \log P(\Lambda_\epsilon) - \log Q_Y(Y^m) - \log Q_\epsilon(\Lambda_\epsilon)\right) \\
&\equiv \mathbb{E}_Q\Big(N\log|\Lambda_\epsilon| - \left|\left|Y - S\beta\right|\right|^2_{\Lambda_\epsilon} + (e-P-1)\log|\Lambda_\epsilon| - \mathrm{tr}\left(E^{-1}\Lambda_\epsilon\right) \\
&\qquad - \sum_n \left(-\log|\Sigma^{Y_n}| - (Y_{nm}-\mu^Y_{nm})\Sigma^{Y_n}{}^{-1}(Y_{nm}-\mu^Y_{nm})^T\right) \\
&\qquad - \left(-e'\log|\Omega| + (e'-P-1)\log|\Lambda_\epsilon| - \mathrm{tr}\left(\Omega^{-1}\Lambda_\epsilon\right)\right) \\
&\equiv \mathbb{E}_Q\left(-\mathrm{tr}\left(\left[(Y-S\beta)^T(Y-S\beta) + E^{-1}\right]\Lambda_\epsilon\right)\right) + \sum_n \log|\Sigma^{Y_n}| + e'\log|\Omega| \\
&\equiv -\mathrm{tr}\left(\left[\Omega_0' + E^{-1}\right]\Omega\right) + \sum_n \log|\Sigma^{Y_n}| + e'\log|\Omega|
\end{aligned}$$

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

$$\begin{aligned}
(3.18) &= 2\mathbb{E}_Q\left(\log P(\beta) - \log Q_\beta(\beta)\right) \\
&= \mathbb{E}_Q\left(-\tau||\beta||_F^2 - \log|\Lambda_b| + (b-\mu_b)^T\Lambda_b(b-\mu_b)\right) \\
&\equiv -\tau\mathbb{E}_Q\left(||\beta||_F^2\right) - \log|\Lambda_b| \\
&\equiv -\tau\left(||\mu_\beta||_F^2 + \operatorname{tr}\left(\Lambda_b^{-1}\right)\right) - \log|\Lambda_b| \\
(3.19) &= 2\mathbb{E}_Q\left(\log P(S) - \log Q_S(S)\right) \\
&= \mathbb{E}_Q\left(-||S||_{K^{-1}} - \log|\Lambda_s| + (s-\mu_s)^T\Lambda_s(s-\mu_s)\right) \\
&= -\operatorname{tr}\left(\mathbb{E}_Q\left(SS^T\right)K^{-1}\right) - \log|\Lambda_s| \\
&= -\operatorname{tr}\left(\mu_S^T K^{-1}\mu_S\right) - \operatorname{tr}\left((I\otimes K^{-1})\Lambda_s^{-1}\right) - \log|\Lambda_s|
\end{aligned}$$

All terms can be computed in $O(N_m P^3 + NP^2)$.

3.4 Other methods for phenotype imputation

MVN: an EM algorithm assuming unrelated samples

Rows of Y are correlated in the presence of genetic relatedness due either to confounding structure or causal variants. Nonetheless, a simple EM algorithm can be derived assuming

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

MVN is the resulting EM algorithm that infers μ and Σ in an M-step and the missing entries of Y (and their second moments) in an E-step; full derivations can be found in [Little and Rubin, 1987; Dahl et al., 2016]. As MVN ignores correlation across samples, it performs well when either relatedness or heritability is low.

LMM: univariate linear mixed models

I use LMM to refer to the model

$$y \sim \mathcal{N}\left(0, \sigma_g^2 K\right) + \mathcal{N}\left(0, \sigma_e^2 I\right)$$

The "fixed" part of the model is presumed handled via REML.

LMM is intrinsically univariate, and so imputing a matrix with LMM imputes each phenotype separately. For each phenotype p, I first run an LMM on $Y_{o_p,p}$, the observed samples, to fit the variance components, $\sigma_{g,p}^2$ and $\sigma_{e,p}^2$. Second, using these estimates, I compute the conditional expectation, or best linear unbiased predictor (BLUP), to impute missing samples:

$$\hat{Y}_{m_p,p} \leftarrow \left(\sigma_{g,p}^2 K_{m_p,o_p}\right)\left(\sigma_{g,p}^2 K_{o_p,o_p} + \sigma_{e,p}^2 I_{|o_p|}\right)^{-1} Y_{o_p,p}$$

This algorithm is embarrassingly parallel across traits but, to facilitate runtime comparisons, a single-thread implementation was used in all experiments.

Typically, ignoring inter-trait correlations means LMM is among the fastest imputation methods considered in this thesis. However, when N is large–as in the chicken dataset–LMM can in fact be slower than phenix. Because LMM models P different sample sets–$(o_1, \ldots, o_P)$–it performs P kinship eigendecompositions; by contrast, phenix models only the full set of samples and performs only one eigendecomposition. As eigendecomposition costs $O(N^3)$, for sufficiently large N LMM becomes P times more expensive than phenix.

TRCMA: transposable regularized covariance model

The transposable regularized covariance model (TRCM) uses a mean-restricted matrix normal [Allen and Tibshirani, 2010]:

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

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 imputation methods in this thesis, especially for large N: at every iteration, TRCMA performs $O(N^3)$ computations to learn an $N \times N$ covariance matrix; by contrast, phenix, LMM and MPMM use K and upfront computations to dramatically reduce per-iteration costs. Presumably, TRCMA could be modified to use K, or at least its eigenvectors, for similar computational advantages.

TRCMA is also expensive because of the search for regularization parameters: for R^{-1} and C^{-1}, a penalty magnitude and type (ℓ_1 or ℓ_2) must be chosen by cross-validation. We use two shortcuts to mitigate this cost. First, we use only ℓ_2 penalization: it is much faster (as analytic solutions replace calls to `glasso` [Friedman, Hastie, and Tibshirani, 2008]), it has a globally optimal solution, and [Allen and Tibshirani, 2010] found it worked well even for sparse true precision matrices. Second, we performed preliminary simulations to identify a small but reasonable set of regularization parameters to choose from: 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 analyses. TRCMA regularly chose ρ's in the interior of this grid, roughly suggesting these ranges are sufficiently wide. While these two speed-ups 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 $\mathcal{G}$–this compromise between runtime and accuracy is hopefully reasonable and representative.

kNN: k-nearest neighbors

We use the function `impute.knn` from the R package `impute` as a non-parametric benchmark [Troyanskaya et al., 2001; Hastie et al., 1999]. We use default settings including, in particular, $k = 10$, except we allow phenotypes with high missingness. The method finds the k nearest neighbors for each phenotype and then imputes missing values to the average of their observed neighbors.

mice: multiple imputation by chained equations

We implement this method with the R package `mice` [Buuren and Groothuis-Oudshoorn, 2011]. We use default parameters and average over 5 (the default) multiply-imputed datasets; this performs dramatically better than using only one but not noticeably worse than using 10, in line with arguments that the sampling variability of across-imputation averages asymptotes quickly [Little and Rubin, 1987].

`mice` implements a variety of methods, but we only used predictive mean matching (`pmm`), the default for numeric variables. For each trait p, the algorithm works in three steps: first, trait p is regressed on the others; second, this regression produces fitted values for trait p, and samples are grouped based on these predictions; finally, each missing sample is imputed to the observed–not predicted–value of one of its neighbors. This loop over p is performed many times, and, as is standard for MCMC, the first imputed phenotype matrix is returned after burn-in and subsequent imputations are returned after sufficiently many step sizes. The steps are explained in greater detail in [Buuren and Groothuis-Oudshoorn, 2011; Dahl et al., 2016].

softImpute

We use the softImpute method of [Mazumder, Hastie, and Tibshirani, 2010; Hastie et al., 2015] as a benchmark from the matrix completion literature. We consider this method roughly representative of the state-of-the-art [Liu et al., 2013], though reported comparisons suggest the relative performances of the many published methods depend heavily on dataset.

softImpute maximizes the penalized likelihood

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

where $||M||_*$ measures the complexity of M to discourage 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 closely follows the guide at

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

We note that softImpute, and most generic matrix completion methods, are motivated by far larger datasets than those considered in this chapter. These algorithms focus on combating overfitting and improving speed and, unsurprisingly, underfit in our moderate-N, small-P datasets. Conversely, most algorithms in this chapter would fail if applied to larger datasets.

MPMM: multiphenotype mixed models

We follow the approach of [Zhou and Stephens, 2014] to impute phenotype data with an MPMM. We fit the model as described in Section 4 to Y after CWD, and predict missing entries with BLUPs conditioning on the full vector of phenotype

observations y_o: defining $\Sigma := (B \otimes K + E \otimes I_N)$, the BLUP is

$$\begin{aligned}\mathbb{E}\left(y_m | y_o, B, E\right) &= \mathrm{Cov}\left(y_m, y_o | B, E\right) \mathbb{V}\left(y_o | B, E\right)^{-1} y_o \\ &= \Sigma_{m,o} \left[\Sigma_{o,o}\right]^{-1} y_o\end{aligned}$$

In general, these computations cost $O(|o|^3)$ (or $O(|m|^3)$ if a Schur complement identity is used). If $|o| \propto NP$, then, the cost of imputing is $O(N^3P^3)$; Section 4.3.5 explains how to reduce this cost to $O(NP^2)$, though we follow [Zhou and Stephens, 2014] and do not use the speedup here. (Further, in special cases the computations easily simplify; see Section 4.3.4.)

3.5 Improved imputation accuracy with phenix

3.5.1 Baseline simulations

Throughout this chapter, all simulated phenotypes are drawn from a standard MPMM. More formally, for any triple (B, E, h^2), where B and E are (possibly random) PSD inter-trait covariance matrices and h^2 is the (scalar) heritability of all traits, I draw

$$\begin{aligned}Y &= U + \epsilon \\ U &\sim \mathcal{MN}\left(0, K, h^2 \texttt{cov2cor}(B)\right) \\ \epsilon &\sim \mathcal{MN}\left(0, I, (1-h^2) \texttt{cov2cor}(E)\right)\end{aligned}$$

defining `cov2cor` to map covariance matrices to their respective correlation matrices by rescaling the diagonal entries to 1. Finally, after drawing Y, we mask 5% of its entries completely-at-random (CAR), retaining the values to assess imputation accuracy.

We use two choices for the kinship matrix K, which determines the amount of correlation between rows of the phenotype matrix. First, we construct K to represent an idealized family study ('Sibs'): each family is comprised of four siblings and families are independent. The resulting K is block diagonal, with blocks of size four for each family. This 'Sibs' kinship matrix represent the upper end of realistic relatedness for human datasets.

The second choice of kinship is created by subsampling (independently for each simulated phenotype matrix) the human NSPHS study [Lauc et al., 2010]. This 'NSPHS' kinship matrix is far less structured than the 'Sibs' kinship matrix, but, for nominally-unrelated humans, still has quite high levels of relatedness. (Additionally, there is some close relatedness in the NSPHS.)

Our baseline parameter choices are: $N = 300$; $P = 15$; B is an AR(1) matrix with autocorrelation parameter $\rho = .45$, i.e. $B_{ij} = \rho^{|i-j|}$; and $E \sim \text{Wi}\left(P, \frac{1}{P}I\right)$, with E redrawn (and scaled) for each simulated dataset. We vary the heritability between .05 and .95.

Figure 3.1 shows the imputation correlations for each method on this baseline dataset. Here and in the simulations generating the below figures, all methods were run on 250 independently simulated datasets for each of 11 values of h^2, and resulting averages over these 250 replicates are plotted. All methods ran all datasets in under two hours on a server, with two exceptions: TRCMA ran only ≈ 125 datasets and, for the larger datasets in Figure 3.2, methods were given four hours (during which LMM completed ≈ 1500 datasets and TRCMA ran none).

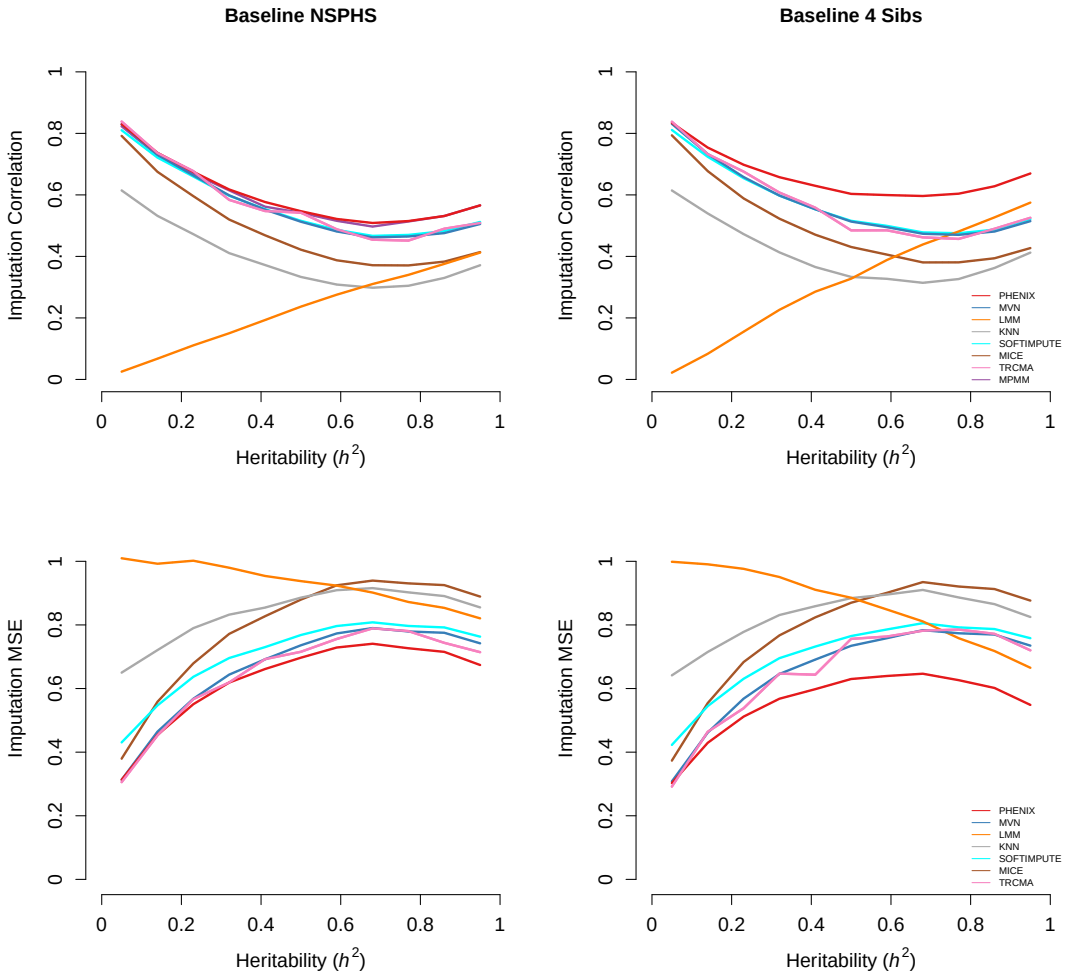


Figure 3.1: Simulation results measuring imputation accuracy with correlation (top) and MSE (bottom). The left panels simulate from an empirical kinship matrix derived from the human NSPHS study [Lauc et al., 2010]; the right panels use a theoretical kinship from 75 unrelated families of 4 sibs. Datasets were simulated at various levels of heritability (x-axis) for the traits. 300 individuals and 15 traits were simulated. 5% of phenotype values were set to missing before imputation. 7 different methods (legend) were applied to impute the missing values. Imputation accuracy is on the y-axis, computed using correlation or MSE. Perfect imputation has correlation 1 (MSE 0) and imputing all entries to 0 has correlation 0 (and, because phenotypes are centered and standardized, MSE 1). Note that qualitative method comparisons are the same using either correlation of MSE.

The primary result is that phenix is always optimal and there is no consistent second-best method. This is because phenix straddles two worlds: the generic matrix completion world, which ignores genetic relatedness; and the univariate genetics world, which ignores inter-trait correlations. Which of these approaches is preferable is entirely dependent on dataset but, regardless, phenix appears to get the best of both worlds.

The methods ignoring relatedness always do well, as the amount of inter-trait correlation is always large; moreover, when heritability is low, these methods are barely worse than phenix. The exceptions may be MICE and kNN, which consistently underperform relative to MVN (which has also been used in genetics [Hormozdiari et al., 2016]), softImpute and TRCMA, which consistently give similar results. LMM, however, has almost no predictive power when heritability is low. Even when heritability is nearly 1, LMM outperforms generic competitors only in the highly structured 'Sibs' scenario. More broadly, inter-trait correlations are generally expected to be large and useful, while inter-sample correlations are only substantial when both relatedness and heritability is high. Even when LMM performs poorly, though, it captures signal orthogonal to the signal in MVN, and phenix–which aims to combine these signals–outperforms MVN by an amount increasing in LMM's performance.

One method, MPMM, is not presented for these simulations. When run on the baseline simulation, it performs worse than phenix by an almost-imperceptible margin. For simplicity, MPMM hasn't been run on the other simulations or considered in depth for the baseline simulation: first, it is expensive (two orders of magnitude more than any method except TRCMA); second, MPMM is the true generative model, so this comparison is not particularly fair; and, most impor-

tantly, phenix is designed not to improve on problems where MPMM is tractable but, rather, to extend the reach of LMM-based methods to larger numbers of traits and higher levels of missingness (both of which are better tested by real datasets than these simulations).

Finally, note that MSE and correlation tell essentially identical stories about the various methods. While this is not generally true, it holds for every phenotype imputation experiment we have looked at and henceforth we assess only correlation.

3.5.2 Modifications to the baseline simulation

The above section uses relatively ideal parameters: low missingness, quite different variance components, quite genetically correlated traits, CAR missingness, no confounding environment, and Gaussian phenotypes. Below, we assess the importance of these factors in turn by modifying the the baseline simulation, describing the modifications in plot captions or, when necessary, in text. For reference, the baseline results are plotted as dotted lines in the background.

Overall, though some modifications noticeably make imputation harder, phenix always remains optimal.

Basic modifications: increased dimensions, missingness, and genetic correlation

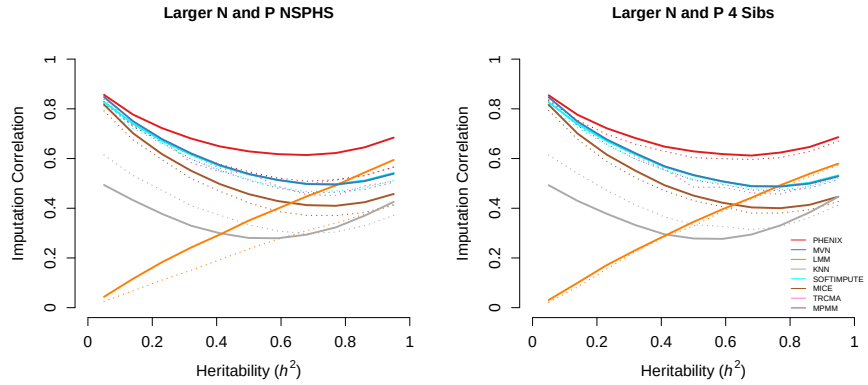


Figure 3.2: Simulation results using larger datasets. This figure uses $(N, P) = (1000, 50)$, while the dotted lines use $(N, P) = (300, 15)$. Increasing the data size nearly always improves imputation accuracy. Relatedness-based methods improve less in the "4 Sibs" model as family sizes are fixed and, thus, the amount of inter-sample correlation does not increase with N.

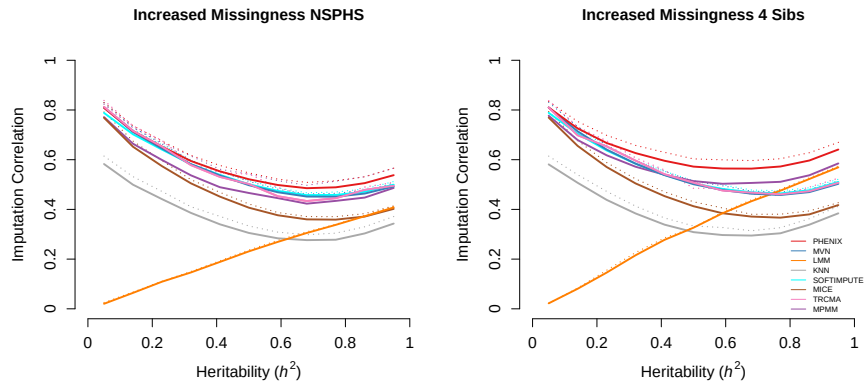


Figure 3.3: Simulation results with higher missingness. 10% of phenotype values were set to missing before imputation, rather than 5% as for the dotted lines. All methods perform slightly worse, though no qualitative comparisons change

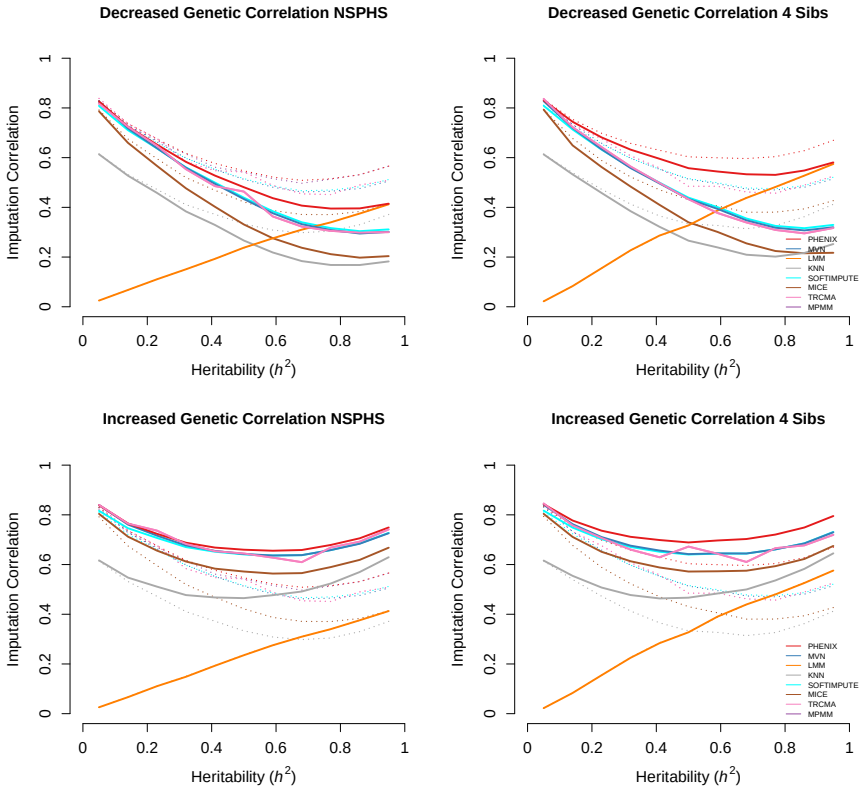


Figure 3.4: Simulation results varying the amount of genetic correlation. We vary the overall genetic correlation matrix B by changing ρ, the AR parameter. The top row shows simulations with $\rho = .275$, decreasing the average genetic correlation between traits compared to the dotted lines that use the baseline choice $\rho = .45$; the bottom row shows simulations with $\rho = 0.675$. Similar results were obtained using $\rho = -.275$ (not shown). Imputation accuracy increases with genetic correlation and this effect increases with h^2. The exception is LMM, which is agnostic to genetic correlations. In fact, LMM is can be optimal with small enough ρ–for $\rho = 0$ and $h^2 = 1$, LMM is exactly the generative model.

Cancellation of genetic and environmental covariances

Intuition Imputation correlation has a noticeably convex shape in Figure 3.1, being minimized near $h^2 \approx .6$ for multi-trait methods. This is most easily understood for MVN: at intermediate levels of heritability, genetic and environmental correlations cancel.

Because we simulate from an MPMM, the true covariance across-traits, within-samples is

$$\mathbb{V}(Y_{i,}) = h^2 B + (1 - h^2)E$$

(This assumes the diagonal of K is all 1s.) MVN implicitly approximates $\mathbb{V}(Y_{i,})$ with its overall-trait-covariance matrix Σ, the size of which (in a deliberately unspecified norm) dictates imputation performance. But, since norms are convex,

$$\|\Sigma\| \approx \|h^2 B + (1 - h^2)E\| \leq h^2\|B\| + (1 - h^2)\|E\|$$

This suggests performance at $h^2 = 0$ and $h^2 = 1$ depends on the size of B and E–in the baseline, the Wishart E is bigger than the AR B and thus MVN favors $h^2 = 0$ to $h^2 = 1$.

To a first approximation, phenix combines MVN and LMM. It is unsurprising, then, that phenix imputation correlation inherits convexity (with respect to h^2) from MVN but with an upward-tilt inherited from LMM.

Simulation To confirm this intuition, we modified the baseline simulation so the off-diagonal entries of B and E exactly cancel: we draw E from a Wishart as before, and then set

$$B_{ij} = -E_{ij} \quad \text{for } i \neq j$$

At $h^2 = .5$, then, $h^2 B + (1 - h^2)E = I_P$. This suggests that MVN has essentially no information and should have nearly 0 imputation correlation at $h^2 = .5$; Figure 3.5 confirms this. Moreover, LMM outperforms phenix near $h^2 = .5$, which is also expected as ignoring inter-trait correlation is an ideal form of shrinkage in this case. Nonetheless, LMM's optimality is short-lived, demonstrating the near-ubiquitous value of inter-trait methods: only in extreme and contrived settings

might LMM be optimal.

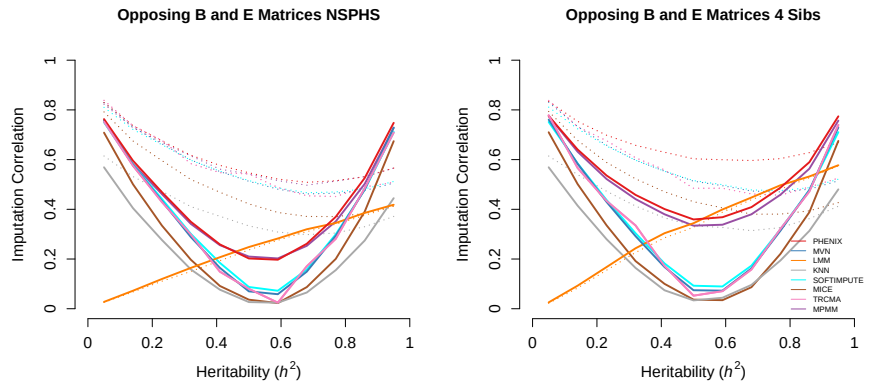


Figure 3.5: Simulation results with counteracting genetic and environmental correlations. Rather than an AR matrix, this plot chooses genetic correlation to cancel the environmental correlation, $B_{pq} = -E_{pq}$ for $p \neq q$. 5% of phenotype values were held out and the correlation between the true and imputed values is plotted on the y-axis for each method.

Non-random missingness

Our model implicitly assumes that missingness is ignorable when updating Y (equations (3.7) and (3.8)) and we simulate this in our baseline by removing 5% of entries CAR. We can simulate non-ignorable missingness, however, by removing entries of Y, independently, with probability depending on their values:

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

where Φ is the standard normal cdf. The proportionality constant is chosen so $\mathbb{E}\,(\text{ fraction of entries missing}) = 5\%$. This non-random missingness makes the problem harder but does not appear to change relative method performances.

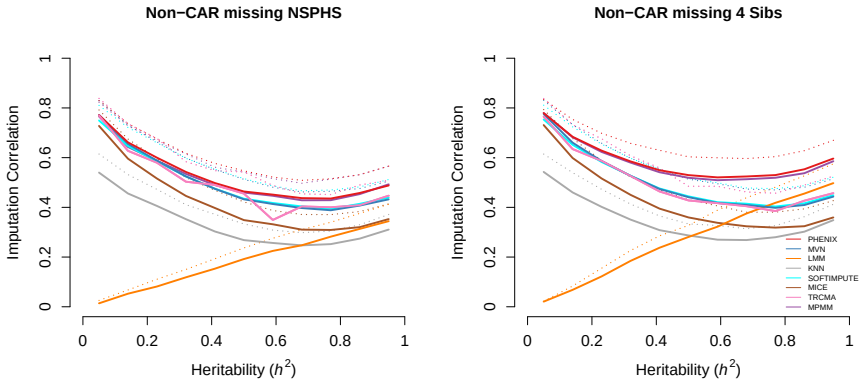


Figure 3.6: Simulation results with non-ignorable missingness. We hold out 5% of the entries of the phenotype matrix with probability increasing in their values and the correlation between the true and imputed values is plotted on the y-axis for each method.

Unmodelled shared environment

We added a random effect to the generating mixed model to represent shared environmental effects and investigate the impact of unmodelled confounders:

$$Y = a^2U + c^2C + e^2\epsilon$$

$$U \sim \mathcal{MN}\left(0, K, \texttt{cov2cor}(B)\right)$$

$$C \sim \mathcal{MN}\left(0, R, \texttt{cov2cor}(D)\right)$$

$$\epsilon \sim \mathcal{MN}\left(0, I, \texttt{cov2cor}(E)\right)$$

Such ACE models comprise an Additive effect (U), a Common environmental effect (C) and an independent Environmental contribution (ϵ) [Falconer and Mackay, 1996].

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 take R to be block diagonal with 10 independent blocks (representing environments) with each block

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 titles), the x-axis in Figure 3.7 determines the relative contributions of the unstructured ϵ and the structured U and C.

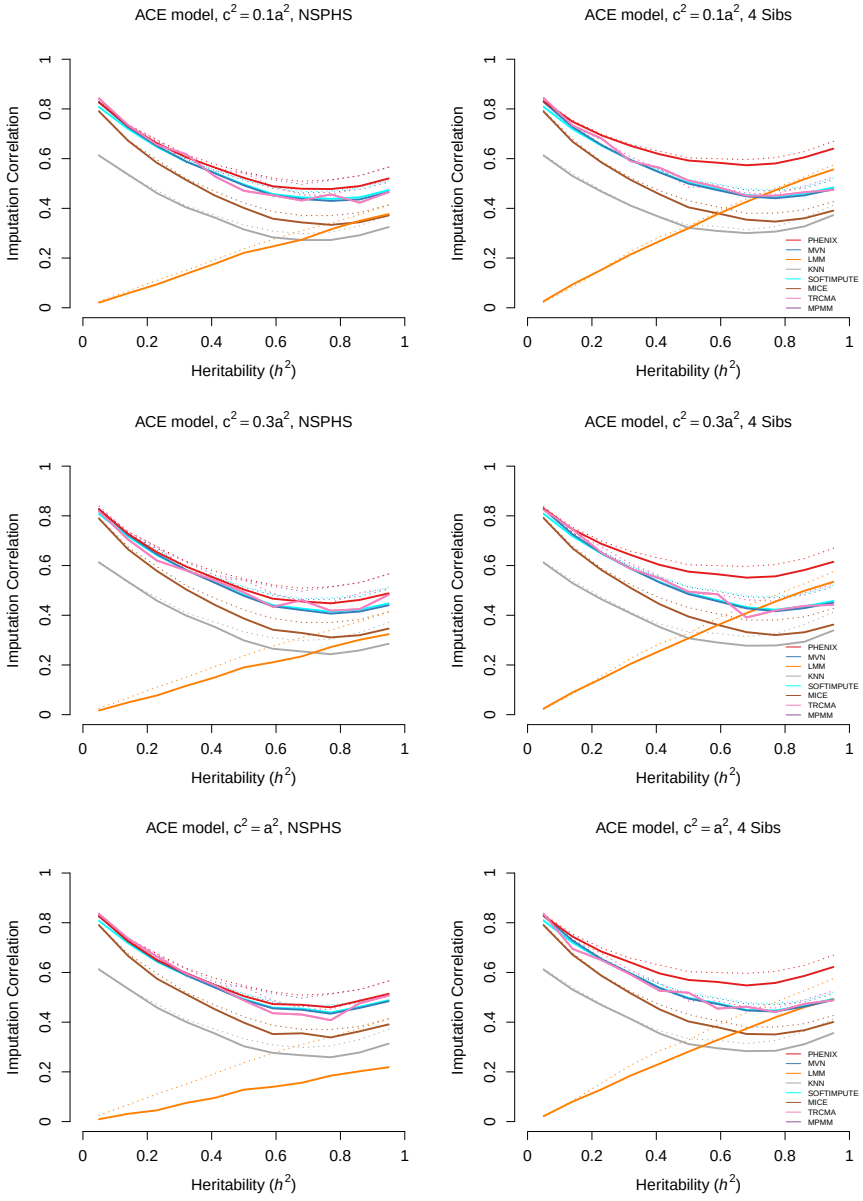


Figure 3.7: Simulation results with confounding cryptic relatedness. The contribution of the additive genetic term–U, in a typical MPMM–is a^2; each row increases the relative contribution of the contaminating shared environment, c^2, to the overall heritability, here defined as $h^2 := a^2 + c^2$.

Again, we see a similar story: increasing c^2, the amount of common environ-

ment, hurts the performance of all methods, but phenix remains optimal.

Non-normally distributed phenotypes

To create non-normal phenotypes, we transform the noise in the baseline MPMM model:

$$Y_{ij} = U_{ij} + e^{\epsilon_{ij}}$$

Phenotype imputation is then performed either on Y or on a quantile normalized version; quantile normalization is natural for most downstream analyses, including GWAS [Fusi et al., 2014].

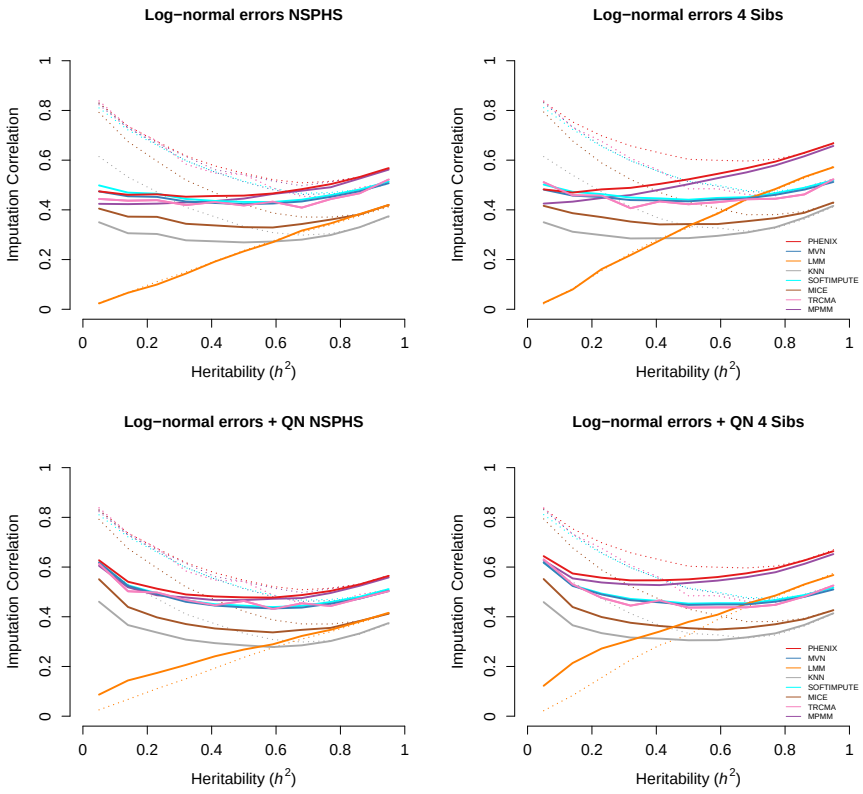


Figure 3.8: Simulation results with log-normal noise. The resulting phenotypes are imputed without (top) or with (bottom) quantile normalization.

Again, all methods suffer from the non-normal noise but phenix remains optimal. Note that softImpute–a flexible, non-parametric method–perhaps slightly outperforms phenix for extremely low heritability and un-normalized phenotypes. However, this low-heritability scenario is quite extreme: non-normality is at its strongest and the advantage of genetics is at its weakest. More importantly, though, quantile-normalization–which we broadly advocate–also makes phenix optimal.

3.5.3 Real data

Datasets

In this section, I test phenix against other methods on a variety of real datasets. The primary goal is to demonstrate that phenix consistently outperforms the state-of-the-art. A secondary goal is to characterize which methods work well in which situations: in particular, I will identify missingness, aspect ratio, and a combination of heritability and relatedness as the key determinants of relative imputation performance. These statistics, and others, are evaluated for our real datasets in Table 3.1. These datasets vary in relatedness from nearly-unrelated to partially clonal, in missingness from nearly 0 to over 50%, in sample size from a few hundred to over 10,000.

The two human datasets are hematological measurements in the UK Blood Services Common Control (UKBS) [Soranzo et al., 2009] collection and glycans phenotypes in the NSPHS study [Lauc et al., 2010]. These are relatively standard GWAS datasets, with low missingness and high $\frac{N}{P}$ aspect ratios. The UKBS contains nominally unrelated samples, the NSPHS comprises a relatively homogeneous population and includes a few close relatives. Human family studies likely

Dataset	N	P	% Missing	Ψ	Reference
UKBS	1,500	6	14.5	.03	[Soranzo et al., 2009]
NSPHS	1,021	15	.1	.05	[Lauc et al., 2010]
Chickens	11,575	12	57.1	.06	[Abdollahi Arpanahi et al., 2014]
Wheat	720	7	2.4	.09	[Mackay et al., 2014]
Yeast	1,008	46	5.2	.10	[Bloom et al., 2013]
Rats	1,407	140	15.8	.12	[Baud et al., 2013]

Table 3.1: Summary statistics for the 6 real datasets. The statistic Ψ, defined in the text, is one way to measure the amount of genetic relatedness in a dataset. Table created using `xtable` [Dahl, 2009].

have relatedness comparable to our non-human datasets.

The rat dataset includes hierarchically-structured phenotypes, with each of trait belonging to one of six disease models [Sequencing and Consortium, 2013]. I will return to this data to perform GWAS in Section 3.6.5.

The yeast dataset, downloaded from

`http://genomics-pubs.princeton.edu/YeastCross_BYxRM/data/cross.Rdata`

and winter-sown wheat dataset, downloaded from

`http://www.niab.com/pages/id/402/NIAB_MAGIC_population_resources`

are both public and measure growth-related traits, the former corresponding to different media and the latter to agriculturally meaningful indicators. The chicken dataset, from a genomic selection program in a multigenerational chicken pedigree, contains food-production-related traits.

It is useful, but difficult, to summarize inter-sample relatedness for each dataset. Nonetheless, we have found the (scaled) Frobenius norm of K useful:

$$\Psi(K) = \frac{1}{\text{tr}\,(K)}\|K\|_F$$

The main disadvantage of this measure is demonstrated by the chicken dataset:

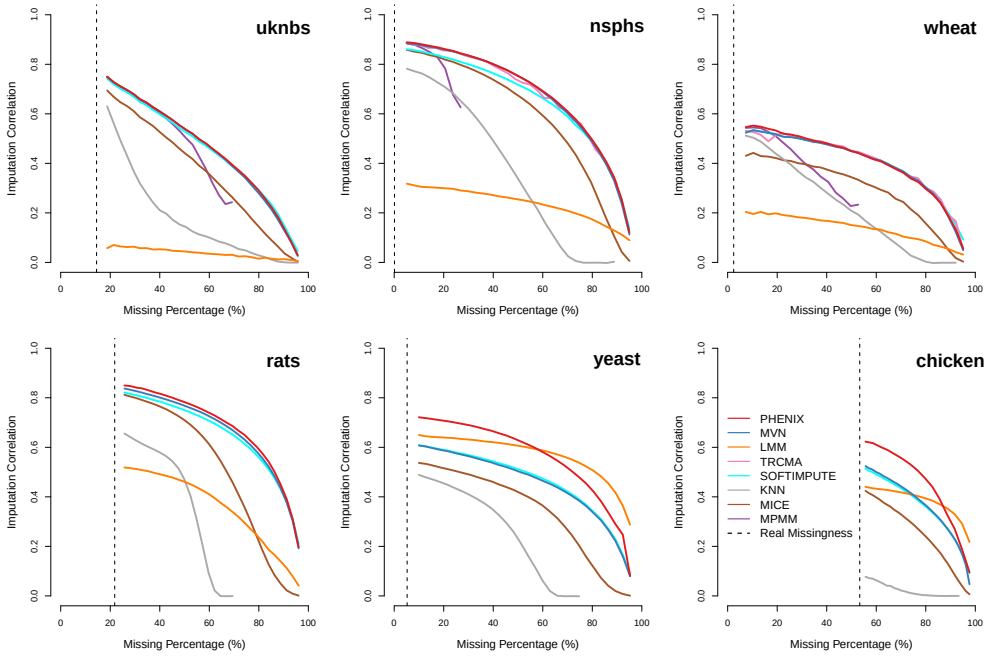


Figure 3.9: Real data imputation accuracy. phenix is essentially always optimal.

despite a dense pedigree and large sample size, the Ψ value is the smallest among non-humans. This is because Ψ is normalized by the trace of K. This is somewhat appropriate as $\|K\|_F$ is non-decreasing even if one adds entirely unrelated samples– but the chicken dataset shows the trace normalization goes too far. Nonetheless, Ψ seems a qualitatively reasonable measure, especially when considering datasets with similar N.

Results

We remove observed phenotype entries completely-at-random (CAR), impute and then assess recovery. We repeated this process 100 times (or 20 for the large chicken dataset) for 11 levels of missingness. The resulting imputation correlations are shown in Figure 3.9.

This experiment tells a story consistent with simulations. First, when inter-sample structure is weak–due to low relatedness and/or low heritability–phenix outperforms generic methods–MVN, softImpute, and TRCMA (when it runs)–by only an imperceptible margin and LMM performs far worse. The other generic methods (kNN and mice) always perform poorly; kNN would improve if k were chose carefully (see Section 3.4) and runs very fast, while mice has no obvious implementation suboptimalities and runs comparatively slowly. Overall, in these datasets where there is little row-structure, phenix is optimal but MVN and soft-Impute remain reasonable and fast choices.

The second key point is that for high-row-structure datasets–including the rats and, especially, the yeast and chickens–phenix outperforms the generic methods by a non-trivial margin and, concomitantly, LMM performs reasonably well. In fact, LMM outperforms the generic methods in yeast, showing that row-structure can be stronger than column-structure. Nonetheless, this yeast dataset with experimentally-controlled phenotypes is a near-ideal example of maximal row-structure and minimal column-structure, and we advocate MVN over LMM in all but extreme cases. However, this dilemma of method choice is obviated by phenix, which is essentially always optimal.

Finally, phenix is not optimal for sufficiently high levels of missingness. This is due, presumably, to the aggressively regularizing nature of VB: at high missingness, uncertainty leads VB to conservative models. This is not proven (or, perhaps, provable) but is consistently observed.

3.5.4 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 3,000 datasets 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 (on the 64 AMD Opteron 6276 and 16 3.30GHz [Intel Xeon E5-2667] processors in parallel, respectively). CWD, however, often removed the entire dataset causing MPMM to fail (for 75% and 50% of the patterns in NSPHS and wheat, respectively).

Next, for (TRCMA, NSPHS), by far the most expensive situation considered, 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 without time constraints. Note that we could have pre-computed the eigendecomposition of K for phenix (but not for LMM, which drops samples).

Method	Base Sim	Rat	Wheat	Chicken	Yeast	Human
phenix	0.1	32.3	0.1	5.8h	2.5	1.5
MVN	0.0	1.4	0.0	0.1	0.1	0.0
LMM	0.1	11.8	0.0	2.2h	1.3	0.5
softImpute	0.1	8.0	0.1	5.1	1.3	0.5
kNN	0.0	NA	1.8	15.2h	30.4	11.4
mice	0.1	40.8	0.0	2.5	0.9	0.2
TRCMA	19.4	NA	8.0h	NA	157.1h	126.0h

Table 3.2: Runtimes for each method on each dataset. Times are in minutes by default, but 'h' means the time is in hours. Timings are rounded to 1 decimal point, so 0.0 means below 0.05 minutes.

3.6 Phenotype imputation and GWAS power

In this section, I first show that p-value miscalibration is not a substantial problem via realistic simulations. Then I show phenix gives a modest power boost in univariate tests and a dramatic boost in multivariate tests. I then introduce a QC metric to avoid false positives and negatives in real data. Finally, I compare standard univariate rat GWAS results with and without phenotype imputation, resulting in new, plausible hits.

3.6.1 Phenotype imputation and Type 1 error rate: simulations

To assess the impact of phenotype imputation on the null distribution of GWAS p-values, we simulated phenotype data from the baseline MPMM model outlined in Section 3.5.1, except with 10% missingness. Note this is the genome-wide null for mixed models, as no SNP has a causal effect (beyond that absorbed in the polygenic term). Simulations varying the parameter of the AR(1) matrix, h^2, and the level of missingness did not qualitatively differ (not shown).

We use the same 'Sibs' and NSPHS kinship matrices as before. We simulated

100,000 unlinked SNPs–used only in testing–for the "Sibs" dataset by first drawing four parental alleles independently for each sibling set–using an across-parent allele frequency drawn from a uniform distribution on $(.05, .5)$–and then drawing sibling genotypes using Mendel's rules. We used 9,165,236 real, imputed SNPs for testing with the NSPHS kinship. We used `gemma` to perform the GWAS [Zhou and Stephens, 2012].

Calibration of GWAS p-values after single-imputation of phenotypes (or genotypes) is not guaranteed: in general, errors due to inexact imputation invalidate the assumptions of downstream analyses that fail to distinguish between observed and imputed data. The only general solution to this is multiple imputation (MI) [Little and Rubin, 1987; Rubin, 1996]. By re-drawing imputed values from a posterior and re-running downstream analyses for each draw, MI accurately propagates the error in imputed values to yield calibrated estimates. I have found MI difficult to implement for phenix as draws from the true, not variational, posterior over the missing data are required. Though tools exist to estimate second moments of the exact posterior from the variational approximation [Giordano, Broderick, and Jordan, 2015], they do not easily admit the KP computational simplifications necessary for tractable inference.

Regardless, the QQ-plots from our null simulation, shown in Figure 3.10, suggest that $-\log_{10}(p)$-value inflation is not a substantial problem. This is at least partially because the LMM used for association testing somewhat controls for the sample relatedness used in, and induced by, phenix. This result does not suggest phenotype imputation generally gives calibrated downstream results but, rather, that in well-behaved circumstances p-value miscalibration is negligible. Similar heuristic arguments underlie the common, and successful, practice of performing

ordinary GWAS on singly-imputed genotypes.

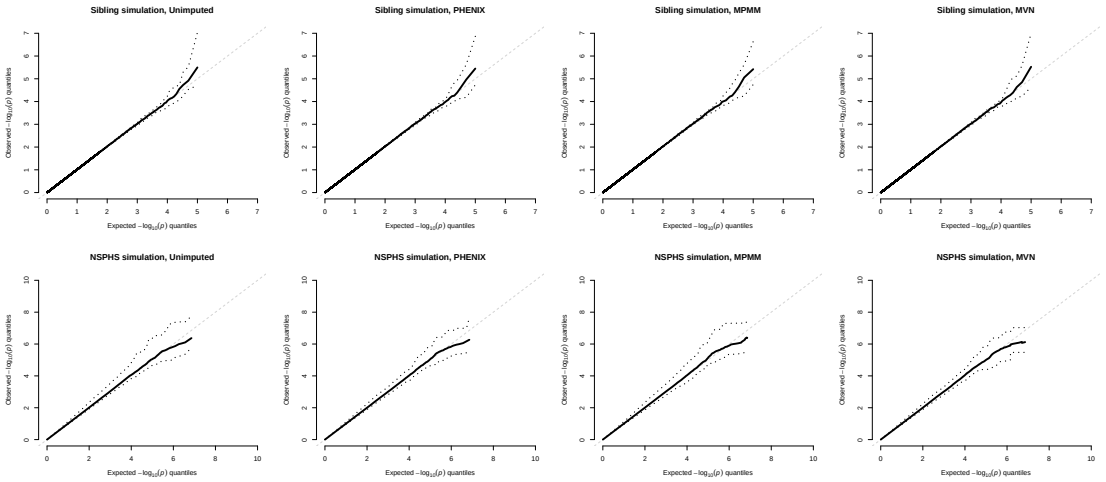


Figure 3.10: QQ plots from performing GWAS on 15 truly unassociated phenotypes with different imputation options (panel titles). Phenotypes are generated from our baseline simulation with either the 'Sibs' (top) or NSPHS (bottom) K matrix. Rather than represent each of the 15 GWAS for each panel, we plot the point-wise minimum and maximum (dotted lines) and median (solid line) of the 15 lines.

3.6.2 Power of single phenotype tests

Next, we simulated non-null data to assess power gains from phenotype imputation:

$$Y = X\beta + \mathcal{MN}\left(0, K, h^2 B\right) + \mathcal{MN}\left(0, I, (1-h^2)E\right) \tag{3.20}$$

$X \in \mathbb{R}^{N\times 1}$ is a common SNP that we draw independently for each dataset by $X_i \overset{\text{iid}}{\sim} \text{Binomial}\,(2, .2)$.

We choose $N = 5,000$, $P = 15$, 'Sibs' K and let B be AR(1) with autocorrelation $\rho = -.2$ (so there are positive and negative genetic correlations). We again draw Wishart E, but now we fix it at the outset (realizations of the matrix normals are still independently redrawn for each dataset). Finally, we set $h^2 = .3$

(this nominal heritability does not include the heritable contribution of X).

We choose a pleiotropic $\beta \in \mathbb{R}^{1\times P}$ so that the SNP X has a substantial effect on the first phenotype, which represents a phenotype of primary interest, and lesser effects on the other fourteen, which represent proxy phenotypes. In this section, we test only the first phenotype, using the other fourteen only for imputation.

Specifically, we choose β by the percent variance explained (PVE) of each phenotype: the PVE for phenotype 1 is 8%, and the other 14 PVEs were drawn randomly:

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

To introduce sparsity, the smallest 5 PVE were hard-thresholded to 0. The realized values used to create Figure 3.11 are displayed in the first two columns of Table 3.3.

We draw 1,000 independent realizations of model (3.20), each time masking 5% of Y's entries CAR. Each time, we run `gemma` on the resulting X_{o_1}, $Y_{o_1,1}$ and K_{o_1,o_1}, and we calculate power as the fraction of datasets with association p-value below 5×10^{-7}. We repeat this with phenix imputation: we impute Y to form $\hat{Y}$ and then run `gemma` on $\hat{Y}_{,1}$.

The results are shown in Figure 3.11. Unsurprisingly, more missing data always hurts power. But phenix can mitigate this loss by recovering some information about the unobserved phenotypes from correlated and observed phenotypes. This simulation suggests that phenix can modestly improve signal for univariate GWAS. We show in Section 3.6.5 that these small boosts can be sufficient to uncover novel associations.

	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

Table 3.3: Randomly generated percent-variance-explained (PVE) per phenotype and the corresponding regression coefficients used to generate Figures 3.11 (first 2 columns) and 3.12 (last six columns, each pair corresponding to a different line type in 3.12) for 15 simulated phenotypes. Univariate tests (columns 1 and 2) are performed on phenotype 1; multivariate 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. Table created using `xtable` [Dahl, 2009].

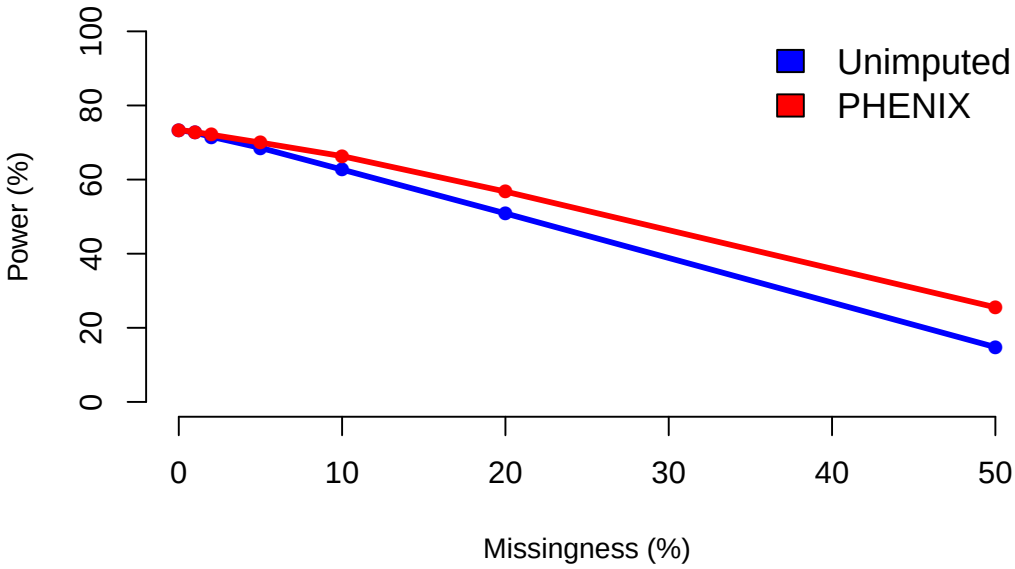


Figure 3.11: Power to detect a simulated, causal SNP using a univariate mixed model (LMM). 5,000 samples, comprising independent sets of 4 siblings, have 15 simulated phenotypes with a pleiotropic effect. 5% of phenotypes are deleted and then an LMM is run with `gemma` [Zhou and Stephens, 2012], either dropping missing data (Unimputed) or after imputing with phenix. Power is calculated by averaging over 1,000 independently simulated datasets using the standard GWAS p-value threshold 5×10^{-7}.

3.6.3 Power of multiple phenotype tests

This section modifies only β in the simulations from Section 3.6.2. Here, we use three different choices to represent varying levels of pleiotropy. In the first situation (UV), the causal SNP affects only the first phenotype; in the second (sparse), the SNP affects 10 of 15 traits; in the third (dense), the SNP affects all 15 phenotypes. All 15 traits are tested for association with the SNP X in all cases, representing the practical uncertainty about which traits a SNP affects. The three β were defined from the β in Section 3.6.2: the UV signal simply sets all but the first entry to 0; the second and third were set proportional to β before and after, respectively, the hard-thresholding step (and then scaled to yield power away from 0 and 1). The resulting PVEs and effect sizes are displayed in Table 3.3.

This section replaces the univariate test on $Y_{,1}$ with a multivariate test on Y, which is expected to be more powerful for detecting pleiotropic SNPs but less powerful when a SNP truly effects only trait 1. Specifically, using the model (3.20), I perform a likelihood ratio test (LRT) of the hypothesis $\beta = 0_P$:

$$\text{LRT} = -2\left(ll(\beta = 0, \hat{B}^{(0)}, \hat{E}^{(0)}) - ll(\beta = \hat{\beta}, \hat{B}^{(1)}, \hat{E}^{(1)})\right)$$

where ll is the MPMM log-likelihood and all estimated parameters are MLEs (under the null for the (0) superscript terms and under the alternative for the (1) superscript terms). I perform this test–which involves fitting VCs, computing β and evaluating the log-likelihood–using the tools developed in Section 4. In particular, I use CWD.

Due to the computational cost of fitting these models, I use the common approximation that the MLE VCs agree under the alternative and the null, i.e.

$$\left(\hat{B}^{(0)}, \hat{E}^{(0)}\right) = \left(\hat{B}^{(1)}, \hat{E}^{(1)}\right)$$

This means only one pair of VCs are computed rather than one per SNP. This approximation is exact when $\hat{\beta} = 0$ and, more generally, is good for SNPs of small effect [Abney, Ober, and McPeek, 2002; Aulchenko, Koning, and Haley, 2007; Kang et al., 2010]. Further, this approach is conservative as the approximate LRT lower-bounds the exact LRT. Nonetheless, this approximation can attenuate power when analyzing SNPs with very large effect sizes [Zhou and Stephens, 2012], and [Zhou and Stephens, 2014] fits the same model without this approximation.

The resulting power estimates are shown in Figure 3.12. They show two clear phenomena. First and foremost, phenotype imputation is crucial for missingness as low as 5%, and can retain nearly-100% power when the unimputed analysis has 0 power. The sharp contrast between these results and the modest power boost in Figure 3.11 is due to CWD: with 5% missingness, for example, sample size is reduced by 5% in the univariate tests but by more than 50% in the 15 trait test. In the univariate test, the power gain was due only to imputed data; here, phenotype imputation also enables the MPMM to use all observed data.

For multivariate analyses with sporadic, CAR missingness, phenotype imputation is essential. For other missingness patterns, however, this need not be true: for example, if the missingness in Y is due to entirely blank rows, CWD yields the same sample size for testing regardless whether one or all traits are considered. Similarly, non-ignorable missing can, in principle, induce arbitrarily poor imputations, in which case CWD would be preferable.

A second fact demonstrated by Figure 3.12 is that the advantage of phenix depends on the degree of pleiotropy. Though complicated by variation in initial power between β choices (line types in the figure), it is clear by comparing the dotted and solid lines that more-pleiotropic signals can be more reliably detected. In

these cases, phenix recapitulates not only high-level correlations but the pleiotropic signal itself.

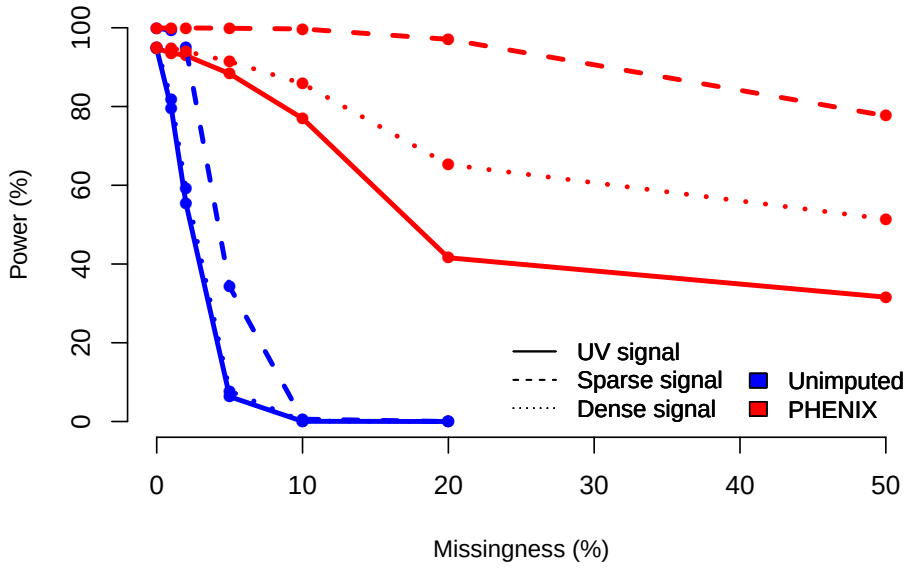


Figure 3.12: Power to detect a simulated, causal SNP using a multiphenotype mixed model (MPMM). 5,000 samples, comprising independent sets of 4 siblings, have 15 simulated phenotypes with three levels of pleiotropy (legend). 5% of phenotypes are deleted then an MPMM is run with our method by dropping 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}. Unimputed power drops more rapidly than for univariate tests because of case-wise deletion: testing more phenotypes requires removing more samples.

3.6.4 Phenotype imputation QC

When using genotype imputation before GWAS, it is standard to use QC metrics to exclude poorly imputed SNPs. The motivation is obvious: regressing on noisy genotypes will inflate error rates. Though there are many QC metrics for genotype imputation, they roughly share a property: testing a variant with a QC score α

and N samples has power similar to testing $N\alpha$ observed samples [Marchini and Howie, 2010]. Similar results can be proven for phenotype imputation using r, the imputation correlation, in simple settings [Hormozdiari et al., 2016]. We will also use r, though we have proven no relationships between r and power.

Regardless, better phenotype imputation will clearly, on balance, give better association tests (the converse, at least, seems undeniable). We attempt to avoid testing poorly-imputed traits by filtering on r: motivated by typical thresholds in genotype imputation, we only take forward traits with $r > .6$. I do not claim this is optimal, but it is clear from Figure 3.15 this choice removes obviously-suspect hits and retains plausible hits.

We follow a procedure similar to that in Section 3.5.3 to compute r: we mask data, impute, and then compute correlations (though we now compute correlations trait-wise, rather than aggregating over traits).

We hold out missing data differently. Before, we used CAR missingness, which put all methods on an equal footing. But now we care about the actual value of the imputation correlation, and so we attempt to hold out data more realistically, copying missingness patterns between samples: until $f\%$ of data is held out, we repeat:

1. Pick a missingness pattern m: for $i \in \{1, \dots N\}$ chosen CAR,

$$m \leftarrow \{p : Y_{ip} \text{ is missing}\}$$

   m is the set of phenotypes that rat i is missing.

2. Copy the missing pattern m: for $j \in \{1, \dots N\}$ chosen CAR,

$$Y_{jm} \leftarrow \text{NA}$$

Rat j is now additionally missing whichever phenotypes rat i is missing.

The resulting Y matrix has the property that each rat's missingness pattern is a union of observed missingness patterns: in this sense, the masked data is realistic.

The missingness patterns in the rat data, shown in Figure 3.13, emphasize the importance of this approach. No rat has been observed on a proper subset of the "Bone" phenotypes, for example, as these traits were procedurally measured together. A CAR missingness approach, however, would create rats observed on, say, all bones but one: given the other bone measurements, that one missing entry would be imputed with exquisite accuracy. But this accuracy is unrealistic, hence undesirable, as no rats have such a charitable missingness pattern. Admittedly, there are certainly similar artifacts unresolved by our approach, and our r estimates are primarily suggestive.

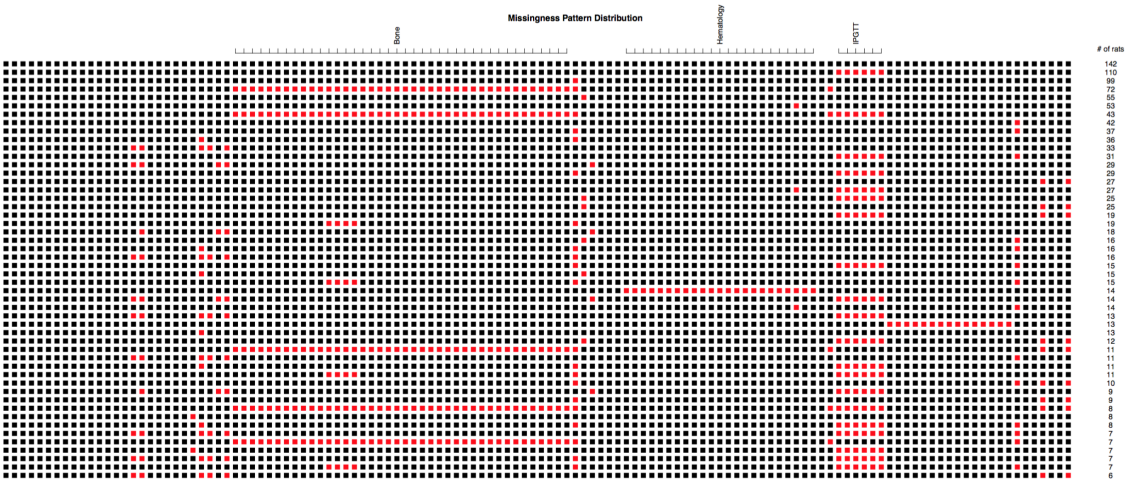


Figure 3.13: Missingness patterns in the rat data [Sequencing and Consortium, 2013]. Missing entries are represented in red; observed entries are in black. Traits are along the x-axis and sorted by phenotype class, some described at the top of the plot. Common missingness patterns are on the y-axis, sorted in descending order by the number of rats with that missingness pattern. Note that bone traits are either all missing or all observed.

Nonetheless, the r metric is calibrated when all modelling assumptions–including, crucially, ignorable missingness–are met. We return to our baseline simulation from Section 3.5.1, this time simulating 1,000 independent datasets and estimating $\hat{r}$. The resulting $\hat{r}$ are potted in the top row of Figure 3.14; as the data are simulated, we can also compute the true imputation correlation r, and these are compared to their estimates in the bottom row. Qualitatively, we clearly estimate r accurately. Our estimates are slightly biased downward, though, as we must estimate imputation accuracy from a masked dataset which, inherently, has less information than the full dataset. This bias can be mitigated by masking a smaller fraction of entries, but then correlation is estimated from a small number of holdout samples. Rather than optimize this classic bias-variance tradeoff, we simply average over many datasets with small added missingness, e.g. we use 1,000 different 5% holdout patterns in the rat GWAS.

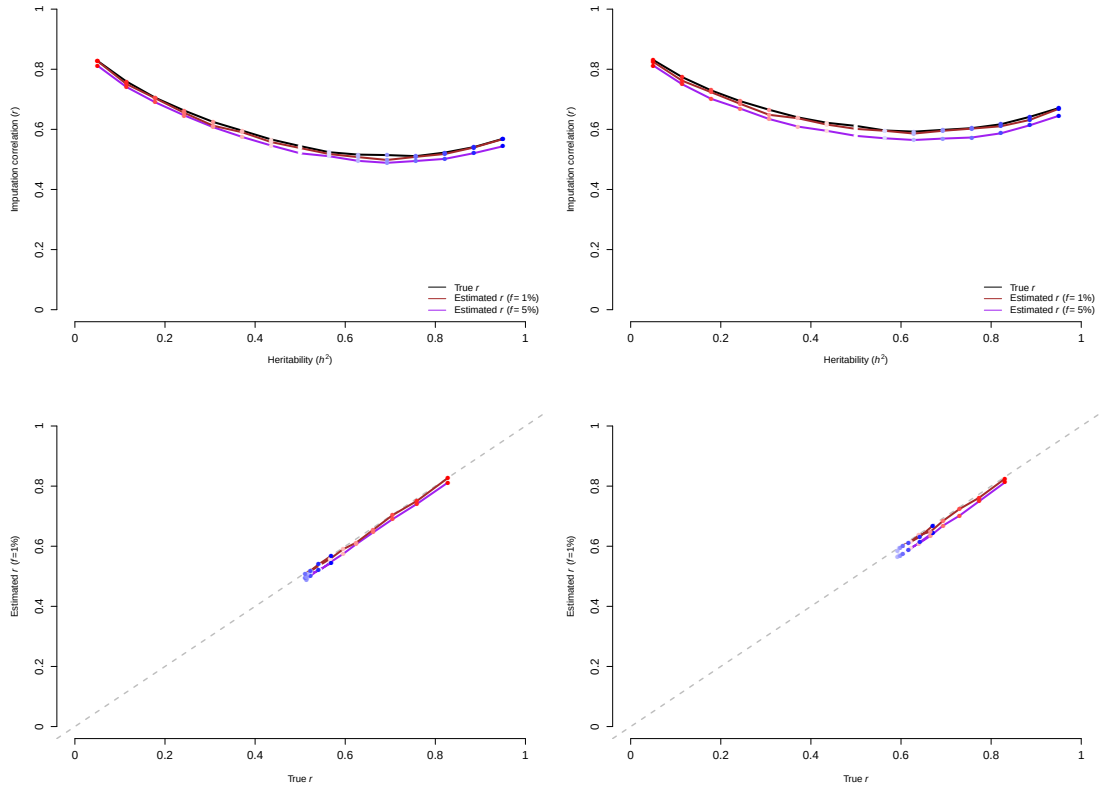


Figure 3.14: Calibration of $\hat{r}$, our estimate of imputation accuracy that we use as a QC metric. Data is simulated from the baseline model in Section 3.5.1, and both $\hat{r}$ and r are computed for 1,000 independent datasets Top row: imputation correlation is plotted against h^2. The black line is the true imputation accuracy and agrees with the phenix line (red) in Figure 3.1. We estimate r in two ways: by hiding 1% (brown line) or 5% (green 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.

3.6.5 phenix uncovers novel associations in outbred rat GWAS

We performed GWAS on the phenotypes used in the Rat Genome Sequencing and Mapping Consortium paper [Sequencing and Consortium, 2013]. In total, we used

317 phenotypes to carry out phenotype imputation; this was a superset of the "biologically meaningful" traits used for GWAS and in Section 3.5.3. Including the extra traits is akin to including extra covariates: a slight overfitting cost is likely small compared to the value of informative predictors. (Nonetheless, results did not change qualitatively when imputing from only the 140 GWAS traits.) Though the original study performed GWAS for 160 traits, we re-analyzed only the 140 that were originally analyzed with mixed models. We tested association as in the original paper: using their kinship matrix, their trait pre-processing, their covariate-to-trait pairings, their haplotype dosages (HAPPY [Mott et al., 2000] descent probabilities evaluated at 24,196 loci using the 8-founders structure of the data), and their mixed model F-test for testing fixed haplotype effects (thanks to Amelie Baud for this code).

As advocated in Section 3.6.4, we compute per-trait imputation r values and threshold at .6. 82 phenotypes pass out filter; ordinarily, GWAS would be run only on these 82 traits, though we analyze all traits to assess this filter. The missingness in these 140 traits ranged from roughly 1% to nearly 90%–unsurprisingly, highly-missing traits are filtered by the r metric.

After imputing phenotypes and analyzing both the unimputed and imputed phenotypes, we compared $-\log_{10}(p)$-values from the two GWAS in Figure 3.15. To be conservative, we apply a p-value threshold of 10^{-10} to account for the many tests performed. Each point in Figure 3.15 represents the minimum p-value in a 6 Mb window (rather than single markers) for a specific trait. The grey $y = x$ line represents associations where phenotype imputation has no effect; points above (below) the line are more (less) significant after phenix.

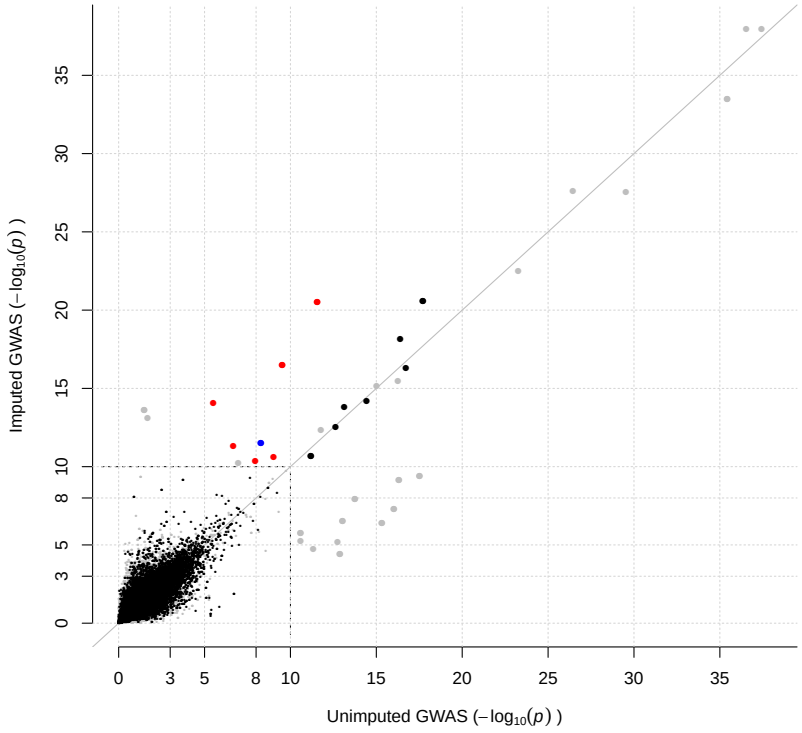


Figure 3.15: Comparison of GWAS with and without phenix. The x-axis show $-\log_{10}(p)$-values for the unimputed GWAS and the y-axis shows the same for the imputed GWAS. Each point corresponds to a (genomic region,trait) pair. Dashed black lines show our conservative p-value threshold. Points in grey fail our QC metric. Points in red and blue correspond to novel platelet and immune associations, respectively.

Points are colored to distinguish the two typical effect phenix has. First, phenotypes that fail the QC filter are represented with grey points, and these include both likely false positives (with unimputed $-\log_{10}(p) < 3$ and imputed $-\log_{10}(p) > 12$) and obvious false negatives (with unimputed p-values many orders of magnitude larger). The large cluster of grey points with unimputed $p < 10^{-10}$, for example, all correspond to immune traits measured (as a block) on a small

minority of rats–exactly the type of traits we hope to filter. Filtration based on estimated imputation correlation removes all obviously suspicious new positive and negative results.

Second, a few new hits are obtained. We focus on the points in red and blue. The red correspond to a single locus and a few related platelet traits. The p-values in the locus, along with potentially biologically relevant genes and phenotype histograms, are displayed in Figure 3.16. There are many platelet-related genes in this region (Igfbp2, Igfbp534, Fn135, Epha436, Cps137,38, Ctla439 and Hspd140), and it is a natural candidate for a novel, biologically meaningful result; a more sophisticated interpretation, however, is beyond the scope of my analysis. We note that similarly plausible genes underlie the blue point in Figure 3.15 (not shown).

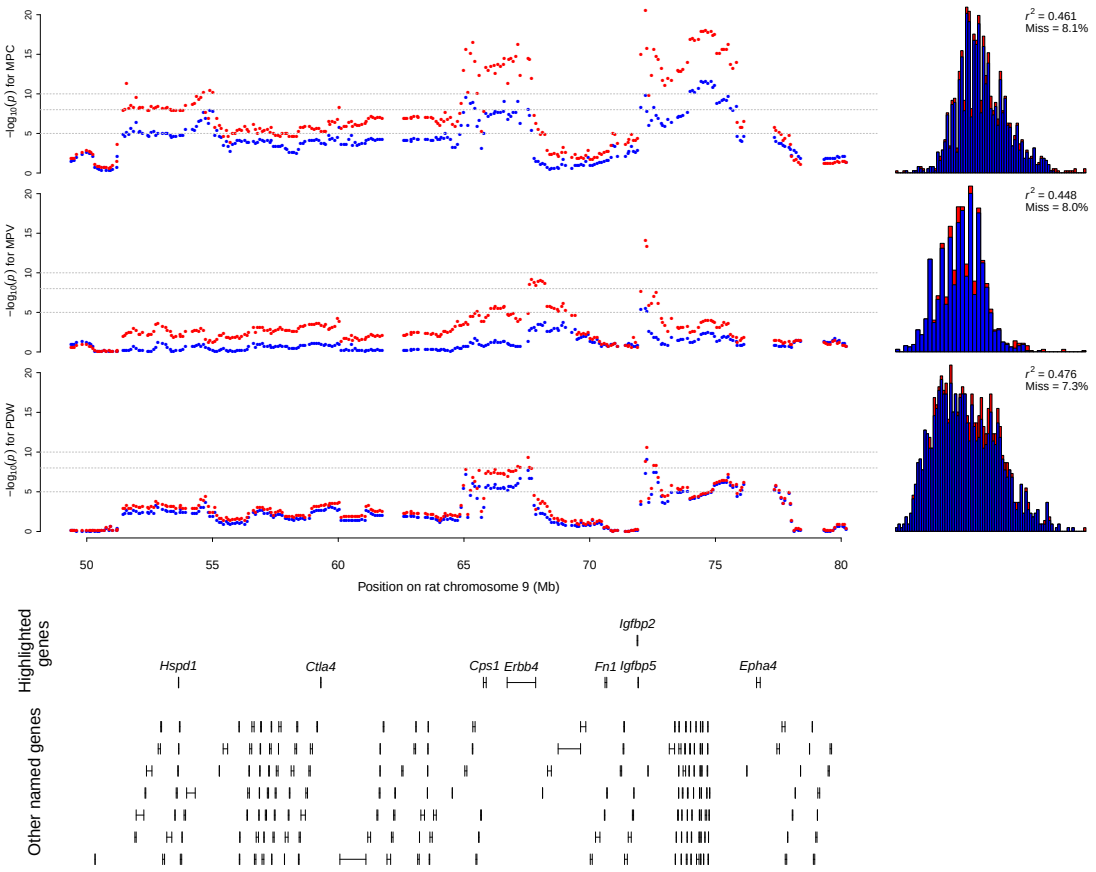


Figure 3.16: Association results for three platelet phenotypes in a region on rat chromosome 9. Blue points represent unimputed association values, while red represent imputed association values; similarly, blue parts of the histograms are observed while red parts are imputed. Genes related to platelet function are also shown. Trait missingness and imputation accuracy are shown above phenotype histograms.

Overall, it is clear phenix can boost power in univariate tests when many phenotypes with missing data are studied. However, it is also clear that phenotype imputation can (sometimes dramatically) attenuate association signal, and we advocate using phenix as a complementary analysis to ordinary GWAS. As QC filters identify usefully imputed traits *a priori*, this does not bring an excessive multiple

testing burden or risk of false positives.

3.7 Future directions

The primary shortcoming of the phenix model is its inability to model a large number of traits. In particular, the Wishart environmental noise model induces $O(P^3)$ computations, limiting P to the order of hundreds–for example, a simple $P \times P$ matrix inversion costs over five minutes when $P = 5,000$ on a standard laptop.

A second shortcoming is the impossibility of heritability estimation as phenix asymmetrically models the heritable and environmental components. For example, a rank-one environmental confounder, in principle creating structure useful for imputation, would be difficult to leverage in our model that partitions variance into a low-rank genetic term and a full-rank environmental term. For this reason, the size of $S\beta$ cannot reliably be used to infer heritability. Closely related, phenix summarizes all genetic similarity into the aggregate relatedness matrix K and thus cannot obtain the prediction advantages of modeling multiple levels of relatedness [Speed and Balding, 2014].

These two weakness are addressed, at least in part, by the compressive mixed models introduced in Section 5, which extend to thousands of traits and reliably estimate heritability (even for multiple relatedness matrices).

There are other directions phenix could be taken as well. First, binary and categorical traits are not explicitly modelled, though they can be encoded as 0/1 variables and treated as continuous–which we have done for the rat GWAS. As variational approximations to logistic link functions are standard, this presumably could be incorporated into the VB algorithm as an additional hierarchical layer.

Multiple imputation, too, would be an extremely valuable addition to the phenix pipeline. First, it would obviate QC concerns. More importantly, it would enable analysis on datasets with very high missingness: currently, phenix would simply impute poorly and, using the QC metric, decide not to take any traits forward. However, in the coming era of large biobanks and, more generally, big phenotype datasets, handling massively-missing matrices will be necessary.

Another possible direction would be to model non-CAR missingness amongst the phenotypes. While CAR may be a reasonable assumption in many contexts–for example, with experimentally designed missingness or with certain technical measurement failures–it will not generally hold. In massive phenotype databases, in particular, missingness is likely to be badly confounded by social factors like access to healthcare. Moreover, these factors are particularly problematic for genetic studies, as many potentially relevant social factors correlate with genetics.

Chapter 4

Generalized linear mixed models

4.1 Linear Mixed Models

The recent popularity of mixed models in genetics applications has spurred a parallel stream of research into computationally efficient algorithms to fit increasingly rich random effect structures. Beginning with tools dedicated specifically to genetics that exploited the problem's unique structure [Abney, Ober, and McPeek, 2002; Yu et al., 2006; Aulchenko, Koning, and Haley, 2007; Kang et al., 2008; Kang et al., 2010; Zhou and Stephens, 2012; Lippert et al., 2011; Pirinen, Donnelly, and Spencer, 2013], developments in the field have diverged into (at least) three mostly distinct directions, each generalizing LMMs in a unique way: models with multiple random effects, multiple phenotypes or likelihood penalties. This Section presents a model and class of algorithms that unifies and generalizes these three LMM extensions, which I call general linear mixed models. Even without penalization, a token `R` implementation can easily fit random effect models with thousands of individuals, dozens of phenotypes and dozens of variance components.

Multi-effect and multi-trait mixed models are discussed in Section 1.2.1 and below. Penalization is not ordinarily considered as important, but is ubiquitously

implicit, e.g. the truncation of heritability to [0,1] or adaptive construction of relatedness matrices [Listgarten et al., 2012; Speed and Balding, 2014]. Explicit uses have focused on improving estimate efficiency and generalization performance [Rakitsch et al., 2013]; penalties can also be used to estimate heritable graphical models, a novel approach to learning coheritability [Stegle et al., 2011].

As this section unifies well-known extensions, the goal is not development of new tricks but rather integration of existing ones. In particular, a bi-diagonalization idea allows efficient multi-trait inference [Rakitsch et al., 2013; Zhou and Stephens, 2014; Lippert et al., 2014]; a Woodbury application does the same for two [Listgarten et al., 2013] or arbitrarily many [Speed and Balding, 2014] VCs; and working with EM, rather than gradients, trivially allows incorporation of many penalty functions [Stegle et al., 2011]. The primarily tool is the GLMM decomposition from Section 2.3.

I first introduce the GLMM, which generalizes and unifies these above models. After developing an efficient class of algorithms to fit VCs in Section 4.3, I return in Section 4.4 to address some of the applications discussed in Section 1 in the more general GLMM framework. The cost of this generality is additional computational complexity and algebraic overhead, and computational scaling is assessed in Section 4.3.6. Despite the "efficient" GLMM implementation, tailored methods, especially for standard problems (e.g. GWAS), should outperform *any* generic GLMM implementation.

4.2 Generalized Linear Mixed Models

4.2.1 The GLMM

Let $Y \in \mathbb{R}^{N \times P}$ be a fully observed matrix of N individuals measured on P traits; we assume throughout that $N > P$, though this is not formally necessary when using penalization. A general linear mixed model (GLMM) with M structured variance components (in addition to the pure-noise variance component) is

$$\begin{aligned} Y &= \textstyle\sum_{m=0}^{M} Z_m \\ Z_m &\sim \mathcal{MN}(0, K_m, C_m) \end{aligned} \tag{4.1}$$

where the $K_m \in \mathbb{S}_+^N$ are *a priori* known relatedness matrices, the $C_m \in \mathbb{S}_+^P$ are unknown $P \times P$ covariance matrices, or variance components (VCs), and $M, P \geq 1$. Note that each VC C_m contributes to all traits unless one of its diagonal entries is zero; though this will happen with probability zero if VCs are estimated by maximum likelihood, a clever penalty function (as discussed below) could, in principle, induce sparsity in active VC-phenotype pairings.

We also allow penalty functions $\mathcal{P}_m$ on each component (discussed in Section 4.2.3) and we maximize the penalized log-likelihood

$$\ell(\{C_m\}_{m=0}^M | Y, \{K_m\}_{m=0}^M) + \sum_{m=0}^{M} \mathcal{P}_m(C_m)$$

Henceforth, we will write C in place of $\{C_m\}_{m \geq 0}$ (and analogously for K), e.g. we write the log-likelihood $\ell(C|Y, K)$.

Fixed effect estimates and tests can be easily developed, but are omitted due to space constraints. Note, however, covariates can simply be projected out. This approach, called REML, yields more precise and less confounded VC estimates when the covariates are meaningful, and even has advantages over full ML.

The Z_m can be analytically marginalized, giving

$$y \sim \mathcal{N}\left(0, \sum_{m=0}^{M} C_m \otimes K_m\right)$$

Perhaps surprisingly, we generally find this representation less useful than the missing-data representation in (4.1): roughly speaking, the latter decouples across components into well-studied forms, enabling the use of highly-optimized software to solve otherwise complex penalized likelihood problems. I am not aware of other EM applications where the likelihood can be written in closed form.

4.2.2 Related methods

This model generalizes and unifies many extensions of mixed models recently developed for genetics. When $P = 1$, the matrix normals collapse to multivariate normals and, without penalization, (4.1) becomes a univariate, multi-effect model [Yang et al., 2011; Listgarten et al., 2013; Speed and Balding, 2014]. When $M = 1$, (4.1) becomes the sum of two matrix normals, comprising aggregate genetic signal and noise, and becomes the standard MPMM [Korte et al., 2012; Zhou and Stephens, 2014; Rakitsch et al., 2013; Lippert et al., 2014]. If $M = 1$ and penalties are included, (4.1) is exactly the model used in [Dahl et al., 2013] and is closely related to that in [Stegle et al., 2011] and the factor models in [Runcie and Mukherjee, 2013] and Section 3. Except the factor models, GLMMs generalize and unify all these approaches and can be applied to the same problems.

In particular, [Casale et al., 2015], published during preparation of this thesis, proposes a very similar method with very similar computational ideas, albeit without penalization and requiring $M = 2$. They provide a thorough analysis of real data and compare their method to the $M = 1$ and $P = 1$ methods discussed

above and demonstrate the richer random-effect model can increase power. This analysis is surprisingly similar to, and in my opinion supercedes, that in Section 4.4.4.

Our generalization comes at no additional overhead computational cost, in the sense that our method restricted to any sub-model, e.g. with $M = 1, 2$ and/or $P = 1, 2$, has the same complexity as methods dedicated to that sub-model. The only exception is [Casale et al., 2015], which is incomparable as the requirement that $M = 2$ dramatically simplifies the Cholesky decomposition (which could be handled as a special case in the GLMM approach). Nonetheless, additional layers of complexity bring greater computational cost, and our general model (without penalties) is certainly more expensive than special cases.

Non-standard mixed models

LD score regression (LDSR) [Bulik-Sullivan et al., 2015b], an alternative to LMMs mentioned in Section 1, also generalizes to partition heritability amongst multiple random effects [Finucane et al., 2015] and multiple phenotypes [Bulik-Sullivan et al., 2015a]. Despite addressing the same question, the models are distinct in every other way. First, they are methodologically worlds apart, as LDSR reduces to OLS. Second, the approaches use different input data, as LDSR uses only GWAS summary statistics (and an LD reference panel). Third, I fit GLMMs via (RE)ML, while LDSR uses a moment method. Due to these inherent simplifications, LDSR dominates GLMMs–and LMMs, generally–in the world of big data. This is especially true in the current environment where, due to journal and funding body preferences, a very precise set of summary statistics are released by most GWAS.

BOLT-LMM is another recent LMM generalization–also proposed during prepa-

ration of this thesis–that GLMMs do not encapsulate [Loh et al., 2015a]. BOLT-LMM adds a second Gaussian component for the genetic effect, as in [Zhou, Carbonetto, and Stephens, 2013], and uses variational machinery to approximate the resulting posterior. These additions are shown to improve power and are only possible because a new inference engine is developed. In particular, conjugate gradient (CG) descent is used to circumvent inversion–the same inversion the GLMM decomposition laboriously rewrites–essentially modernizing an approach reviewed in [Legarra and Misztal, 2008]. (An overview of CG is given in Section A.5.) Unlike standard GWAS LMMs, which scale $O(N^3 + LN^2)$ for L SNPS, BOLT-LMM scales $O(LN^{1+\alpha})$, where $\alpha > 0$ depends on the requisite number of CG iterations and is argued to be roughly $\frac{1}{2}$, at least for $N < 10^6$ (analogous memory advantages are obtained).

BOLT-REML, a different application of the same basic engine, extends BOLT-LMM to multiple variance components and multiple traits (and drops the Bayesian elements in BOLT-LMM) [Loh et al., 2015b]. In many regimes, BOLT-REML has better computational complexity (not to mention vastly superior implementation) than the GLMM approach proposed here. Nonetheless, for many traits and VCs and small N (less than, say, 10,000) GLMMs are clearly better complexity-wise: BOLT-REML costs $O(LMN^{1+\alpha}P^{3+\alpha})$, where L is the number of included SNPs (and is huge when including all SNPs as advocated in [Loh et al., 2015b], but could be as small as r in Assumption 1), while GLMMs have complexity $O(N^3 + P^3 + NPR^2)$. Even in this case, however, [Loh et al., 2015b] advocates estimating multi-trait covariance from pairwise trait covariances to avoid the poor scaling in P (though this eliminates the possibility of multi-trait learning, e.g the graphical models in Section 4.4.3, and multi-trait improvements to GWAS power).

In summary, LDSR and BOLT-LMM/REML dramatically computationally outperform GLMMs in many scenarios, are implemented with more sophistication and, I predict, are the future for many mixed model applications in genetics. However, to my knowledge neither can be applied to the structured multi-trait problems discussed below and in some regimes are slower (for BOLT-LMM/REML) or less statistically efficient (for summary-based LDSR).

4.2.3 Penalty functions

We assume penalty functions are chosen so that the penalized covariance problems in the M-step (see below) have known solutions. Due to the general importance and relative cheapness of this step (the M-step has cost constant in N), this is a wide, and growing, class of functions: most simply, diagonal barrier functions recover standard mixed models; Frobenius penalties smoothly shrink the C_m^{-1}'s for robust prediction [Allen and Tibshirani, 2010]; ℓ_1 penalties on C_m^{-1} induce random effects with sparse graphical models [Friedman, Hastie, and Tibshirani, 2008; Banerjee, El Ghaoui, and d'Aspremont, 2008; Yuan and Lin, 2007]; and more complicated penalties can induce sparse graphical models even in the presence of confounders [Ma, Xue, and Zou, 2013].

Each of these penalties induces a simple M step of the form

$$\min_{X \succeq 0} \mathcal{O} := -\log|X| + \operatorname{tr}(SX) + \mathcal{P}(X)$$

For ℓ_1 regularization of the precision, we solve this with calls to `dpglasso`, and Frobenius and diagonal barrier penalties have analytic solutions (see Section A.4 for derivations).

4.2.4 Low-rank variance components

Let $r'_m := \text{rank}(C_m)$. If the C_m are learned via maximum likelihood, they will almost surely be full rank, i.e. $r'_m = P$. Maximum penalized likelihood estimates, however, may have strictly low rank, and this can be used to expedite computations with the C_m throughout this chapter. As we do not directly study low rank C_m, though, we have not incorporated this trick for simplicity (other than inside the GLMM decomposition, which naturally incorporates low rank C_m via the incomplete Cholesky decomposition). For example, if C_m has rank $r'_m < P$, multiplying C_m with a vector can be reduced in computational complexity from $O(P^2)$ to $O(r'_m P)$, but our naive implementation incurs the original $O(P^2)$ cost. Nonetheless, properly exploiting low-rankness in the C_m would be necessary to scale to truly large P.

Beyond this computational necessity argument, low-rank VCs can also be beneficial and realistic. First, their (approximately) low-dimensional nature can provide regularization for improved phenotype prediction [Rakitsch et al., 2013]; more generally, the non-convex nature of the rank constraint may be particularly useful for tasks that rely on accurate heritability estimation in high dimension. Second, low-rank VCs are biologically plausible if a random effect tags variation along some low-dimensional latent trait that then affects observed traits: for a cartoon example, the impact of a trans-regulating gene on genome-wide gene expression might well be approximated by a rank-one VC (assuming a linear model approximates the trans-gene's effect on expression).

4.3 Efficiently evaluating and using penalized MLEs

4.3.1 The EM algorithm for penalized ML estimation

We aim to fit maximum penalized-likelihood VCs, the C_m in equation (4.1). This missing-data representation is naturally maximized with expectation-maximization (EM) [Dempster, Laird, and Rubin, 1977]. EM is a general approach that iteratively predicts the likelihood function by integrating over missing data using current parameter estimates (the E-step) then updates parameter estimates by maximizing the resulting expected likelihood (the M-step).

In our case, these latent variables are the unobserved random effects, called Z_m in equation (4.1). Remembering that $z_m := \mathrm{vec}\,(Z_m)$, we write our missing random effects as $z := (z_1, \ldots, z_M)$ (excluding one z term, say z_0, for now). We denote the E-step expectations as $\mathbb{E}_t := \mathbb{E}_{z|C^{(t)},K,Y}$, so that the EM algorithm iterates

- **E step:** $Q_{t+1}(C) := -2\mathbb{E}_t\left(\ell(C|Y,z,K) + \sum_{m=0}^{M} \mathcal{P}_m\left(C_m\right)\right)$
- **M step:** $C^{(t+1)} \leftarrow \underset{C\in(\mathbb{S}_+^P)^{M+1}}{\arg\min}\ Q_{t+1}\left(C\right)$

As EM increases the penalized likelihood at each iteration, the resulting $\hat{C}_m$ are penalized local-MLEs under arbitrary penalties $\mathcal{P}_m$[1].

The M step could elide the PSD constraint on the C_m, as heritability estimates need not be positive (c.f. [Yang et al., 2011; Finucane et al., 2015]). We note, though, that many penalty functions are meaningful only when the C_m are covariance matrices.

[1]The Hessian of the likelihood is not generally non-negative. Nonetheless, this non-convexity is rarely problematic in practice, especially when C_m's are carefully initialized. Though I have not been able to prove results analogous to [Zwiernik, Uhler, and Richards, 2014], which shows hill-climbing is likely to recover the MLE despite non-convexity, I suspect a related result holds. However, if VCs remain low-rank as N grows, it is not obvious that the requisite concentration of measure is obtained.

Defining $z_0 := y - \sum_{m>0} z_m$, the likelihood splits over the z_m and the E-step becomes

$$\begin{aligned} Q_{t+1}(C) &:= -2\mathbb{E}_t \left(P\left(z, Y | C, K\right) + \sum_{m=0}^{M} \mathcal{P}_m\left(C_m\right) \right) \\ &:= -2\mathbb{E}_t \left(\sum_{m=0}^{M} P\left(Z_m | C_m, K_m\right) \right) + \sum_{m=0}^{M} \mathcal{P}_m\left(C_m\right) \\ &\equiv \sum_{m=0}^{M} \left[-\log|C_m| + \operatorname{tr}\left(C_m^{-1}\Omega_m\right) + \mathcal{P}_m\left(C_m\right) \right] \end{aligned}$$

where Ω_m, the expected sample covariance for component m, is defined

$$\Omega_m := \frac{1}{N}\mathbb{E}_t\left(Z_m^T K_m^+ Z_m\right)$$

If the Z_m were known, the term inside this expectation would be the obvious estimator for C_m.

Given the simple structure of Q_{t+1}, the EM algorithm boils down to partitioning sample covariance amongst the Ω_m in the E-step and solving standard penalized covariance problems in the M-step:

- **E step:** $\Omega_m \leftarrow \frac{1}{N}\mathbb{E}_t\left(Z_m^T K_m^+ Z_m\right)$
- **M step:** $C_m^{(t+1)} \leftarrow \arg\min_X -\log|X| + \operatorname{tr}\left(X^{-1}\Omega_m\right) + \mathcal{P}_m\left(X\right)$

Because the M-step takes a well-studied form, it is easy to incorporate a wide class of penalties (see Section 4.2.3). Incorporating non-smooth penalties, like the graphical lasso, in a hill-climbing framework would be non-trivial, e.g. requiring proximal functions.

It remains to evaluate the Ω_m's. As this involves an expectation over the latent variable z, we first find z's distribution and simplify its moments. Then we simplify the expressions for the Ω_m's by using the Kronecker and low-rank structures of the problem, which is parsimoniously encapsulated by the GLMM decomposition.

Evaluating the Ω_m

In this section and the next, all expectations are implicitly conditional on Y, $\{K_m\}_{m=0}^M$ and current parameter estimates, $\{C_m\}_{m=0}^M$.

Define the precision of z_m as $\Lambda_m := (C_m \otimes K_m)^{-1}$ and the precision of z as $\Lambda_\oplus := \oplus_{m=1}^M \Lambda_m$. Then

$$\begin{aligned}
-2\log P(z|y,K,C) &\equiv -2\log P(z,y|K,C) \\
&= -2\log P\left(y - \sum_{m>0} z_m | K_0, C_0\right) - 2\sum_{m>0} \log P(z_m|K_m,C_m) \\
&\equiv (y - \sum_{m>0} z_m)^T \Lambda_0 (y - \sum_{m>0} z_m) + \sum_{m>0} z_m^T \Lambda_m z_m \\
&\equiv -2\sum_{m>0} z_m^T \Lambda_0 y + \sum_{m>0}\sum_{l>0} z_m^T \Lambda_0 z_l + \sum_{m>0} z_m^T \Lambda_m z_m \\
&= -2z^T (1_M \otimes \Lambda_0)\, y + z^T (J_M \otimes \Lambda_0)\, z + z^T \Lambda_\oplus z
\end{aligned}$$

This means that $z|Y,K,C \sim \mathcal{N}(\mu,\Sigma)$, where

$$\Sigma := [\Lambda_\oplus + J_M \otimes \Lambda_0]^{-1}$$

$$\mu := \Sigma\,(1_M \otimes \Lambda_0)\, y$$

Now, to simplify Σ, we decompose $J_M = 1_M 1_1 1_M^T$ to get (4.2) and apply Wood-

bury to get (4.3):

$$\Sigma = \left[\Lambda_\oplus + (1_M \otimes I)(1_1 \otimes \Lambda_0)(1_M \otimes I)^T\right]^{-1} \tag{4.2}$$

$$= (\Lambda_\oplus)^{-1} -$$

$$(\Lambda_\oplus)^{-1}(1_M \otimes I)\left(1_1 \otimes \Lambda_0^{-1} + (1_M \otimes I)^T(\Lambda_\oplus)^{-1}(1_M \otimes I)\right)^{-1}(1_M \otimes I)^T(\Lambda_\oplus)^{-1} \tag{4.3}$$

$$= (\Lambda_\oplus)^{-1} - (\Lambda_\oplus)^{-1}(1_M \otimes I)\left(1_1 \otimes \left(\Lambda_0^{-1} + \sum_{m=1}^{M}\Lambda_m^{-1}\right)\right)^{-1}(1_M \otimes I)^T(\Lambda_\oplus)^{-1}$$

$$= (\Lambda_\oplus)^{-1} - (\Lambda_\oplus)^{-1}(1_M \otimes I)\left(1_1^{-1} \otimes \left(\sum_{m=0}^{M}\Lambda_m^{-1}\right)^{-1}\right)(1_M \otimes I)^T(\Lambda_\oplus)^{-1}$$

$$= \oplus_{m=1}^{M}\Lambda_m^{-1} - \left(\oplus_{m=1}^{M}\Lambda_m^{-1}\right)(J_M \otimes \Lambda_y)\left(\oplus_{m=1}^{M}\Lambda_m^{-1}\right) \tag{4.4}$$

defining the precision of y as $\Lambda_y := \left(\sum_{m=0}^{M}\Lambda_m^{-1}\right)^{-1} = \left(\sum_i C_i \otimes K_i\right)^{-1}$ in the last line.

(4.4) can be used to evaluate covariance between random effects $i, j > 0$. Define multi-indices (r) and (q) by

$$(A \otimes B)_{(r)(q)} = A_{rq}B$$

for any A and B. Multi-indices select entire blocks of a matrix in a way that commutes with KPs. Using this definition,

$$\mathrm{Cov}\left(z_m, z_l | Y, K, C\right) = \Sigma_{(m)(l)} = I_{m=l}\Lambda_m^{-1} - \Lambda_m^{-1}\Lambda_y\Lambda_l^{-1} \tag{4.5}$$

This can then be used to evaluate the mean of random effect $i > 0$:

$$\begin{aligned}
\mathbb{E}\left(z_i | Y, K, C\right) = \mu_{(i)} &= \left(\Sigma\left(1_M \otimes \Lambda_0\right) y\right)_{(i)} \\
&= \sum_{m>0}\Sigma_{(i)(m)}\Lambda_0 y \\
&= \Lambda_i^{-1}\left(I - \Lambda_y\sum_{m>0}\Lambda_m^{-1}\right)\Lambda_0 y \\
&= \Lambda_i^{-1}\Lambda_y y
\end{aligned} \tag{4.6}$$

using $\sum_{m>0} \Lambda_m^{-1} = \Lambda_y^{-1} - \Lambda_0^{-1}$ in the last line.

Finally, even though Z_0 was treated differently, symmetric equations can be derived for its moments using the above equations:

$$\begin{aligned}
\mu_{(0)} &:= \mathbb{E}(z_0) = y - \sum_{m>0} \mathbb{E}(z_m) = y - \sum_{m>0} \Lambda_m^{-1}\Lambda_y y = \Lambda_0^{-1}\Lambda_y y \\
\Sigma_{(0)(0)} &:= \mathbb{V}(z_0) = \sum_{m,l>0} \mathrm{Cov}(z_m, z_l) = \sum_{m,l>0} \Sigma_{(m)(l)} \\
&= \sum_{m>0} \Lambda_m^{-1} - \sum_{m,l>0} \Lambda_m^{-1}\Lambda_y\Lambda_l^{-1} \\
&= \left(\Lambda_y^{-1} - \Lambda_0^{-1}\right) - \left(\Lambda_y^{-1} - \Lambda_0^{-1}\right)\Lambda_y\left(\Lambda_y^{-1} - \Lambda_0^{-1}\right) \\
&= \Lambda_0^{-1} - \Lambda_0^{-1}\Lambda_y\Lambda_0^{-1}
\end{aligned}$$

This means that even though z_0 is a deterministic function of z and y, it can be treated symmetrically with all other z_m; this is natural since our choice to treat component 0 as special was arbitrary, but does not necessarily hold for approximate algorithms [Stegle et al., 2011].

Now that we have the moments of each Z_m, the Ω_m's are easy to evaluate. Starting from the definition of Ω_m,

$$\Omega_m := \mathbb{E}\left(Z_m^T K_m^+ Z_m\right) = \frac{1}{N}\mathrm{tr}_P\left(\mathbb{E}\left(z_m z_m^T\right)(I \otimes K_m)^+\right) = \frac{1}{N}\mathrm{tr}_P\left(\left(\mu_{(m)}\mu_{(m)}^T + \Sigma_{(m)(m)}\right)(I \otimes K_m)^+\right)$$

first using identify (2.4) and then using $\mathbb{E}_t\left(z_m z_m^T\right) = \mu_{(m)}\mu_{(m)}^T + \Sigma_{(m),(m)}$.

Then, plugging in (4.6) and (4.5) for $\mu_{(m)}$ and $\Sigma_{(m)(m)}$,

$$\begin{aligned}
\Omega_m &= \frac{1}{N}\mathrm{tr}_P\left(\left(\Lambda_m^{-1}\Lambda_y y y^T \Lambda_y \Lambda_m^{-1} + \Lambda_m^{-1} - \Lambda_m^{-1}\Lambda_y\Lambda_m^{-1}\right)(I \otimes K_m)^+\right) \\
&= \frac{1}{N}\mathrm{tr}_P\left(\Lambda_m^{-1}(I \otimes K_m)^+\right) + \frac{1}{N}\mathrm{tr}_P\left(\Lambda_m^{-1}\left(\Lambda_y y y^T \Lambda_y - \Lambda_y\right)\Lambda_m^{-1}(I \otimes K_m)^+\right) \\
&= C_m + \frac{1}{N}C_m\mathrm{tr}_P\left((I \otimes K_m)\left(\Lambda_y y y^T \Lambda_y - \Lambda_y\right)\right)C_m \qquad (4.7)
\end{aligned}$$

using (2.2) and the Kronecker structure of Λ_m to pull C_m's out of the partial trace and obtain (4.7).

Without penalties, the update for C_m is simply $C_m \leftarrow \Omega_m$; at convergence, this fact combined with (4.7) means that

$$\mathrm{tr}_P\left((I \otimes K_m)\left(\Lambda_y y y^T \Lambda_y - \Lambda_y\right)\right) = 0$$

$\Lambda_y y y^T \Lambda_y - \Lambda_y$ is analogous to residuals in linear regression: our "residuals" are nonzero but, at convergence, their projection[2] onto each component is zero, i.e.

$$\Lambda_y y y^T \Lambda_y - \Lambda_y \mapsto \Pi_m\left(\Lambda_y y y^T \Lambda_y - \Lambda_y\right) := \mathrm{tr}_P\left((I \otimes K_m)\left(\Lambda_y y y^T \Lambda_y - \Lambda_y\right)\right) = 0$$

Intuitively, the EM update (4.7) steps in the direction of maximum unexplained residual variance, partitioned amongst components via K_m's.

Though equation (4.7) is a computable expression for the Ω_m's and, thus, completes the EM algorithm, the $O(N^3P^3)$ cost of forming Λ_y is prohibitive. The next section shows how to compute the key equation (4.7) without forming Λ_y.

Using the GLMM decomposition

The goal is to simplify equation (4.7) using the GLMM representation of Λ_y derived in Section 2.3 and restated here for ease:

$$\Lambda_y = (T \otimes Q_{K_1})\left[\Xi - \Psi\Psi^T\right](T \otimes Q_{K_1})^T \tag{4.8}$$

I call equation (4.8) the GLMM decomposition of Λ_y into (T, Ξ, Ψ).

The first term is

$$\begin{aligned}\mathrm{tr}_P\left(\left(\Lambda_y y y^T \Lambda_y\right)(I \otimes K_m)\right) &= \mathrm{mat}\left(\Lambda_y y\right)^T K_m \mathrm{mat}\left(\Lambda_y y\right) && \text{(by (2.4))} \\ &= T\mathrm{mat}\left(\psi_0\right)^T K_m' \mathrm{mat}\left(\psi_0\right) T^T && \text{(by Proposition 2)}\end{aligned}$$

[2] If $\mathrm{tr}\left(K_i\right) = 1$, this is a projection in the sense

$\Pi_i(\Pi_i(X) \otimes I_N) := \mathrm{tr}_P\left((I_P \otimes K_i)\left(\Pi_i(X) \otimes I_N\right)\right) = \mathrm{tr}_P\left(\Pi_i(X) \otimes K_i\right) \overset{*}{=} \Pi_i(X)\mathrm{tr}\left(K_i\right) = \Pi_i(X)$

using (2.2) for $*$.

The Ψ term is pulled outside the partial trace similarly:

$$\begin{aligned}
&\mathrm{tr}_P\left((T\otimes Q_{K_1})\left(\Psi\Psi^T\right)(T\otimes Q_{K_1})^T(I\otimes K_m)\right)\\
&= T\mathrm{tr}_P\left(\left(\left[\sum_{r=1}^{R}\psi_r\psi_r^T\right]\right)(I\otimes K'_m)\right)T^T && \text{(using (2.3) and (A.1))}\\
&= T\sum_{r=1}^{R}\mathrm{mat}\left(\psi_r\right)^T K'_m\mathrm{mat}\left(\psi_r\right)T^T && \text{(using (2.4))}
\end{aligned}$$

As $\mathrm{tr}_P\left(\Xi\left(I\otimes K'_m\right)\right)$ is the partial trace of a block diagonal matrix, it can be computed cheaply (details below). These results can be plugged directly into (4.7), giving the E-step of our algorithm:

$$\Omega_m = C_m + \frac{1}{N}C_m T\left[\sum_{r=0}^{R}\mathrm{mat}\left(\psi_r\right)^T K'_m\mathrm{mat}\left(\psi_r\right) - \mathrm{tr}_P\left(\Xi\left(I\otimes K'_m\right)\right)\right]T^T C_m$$

Computational complexity

Ignoring low-rankness in the C_m for simplicity, evaluating the final expression for all Ω_i requires:

- $P\times P$ matrix multiplications for each i, costing $O(MP^3)$
- GLMM decomposition, costing an additional $O(NPR^2)$ by Proposition 1
- Forming ψ_0, costing $O(NP^2)$ by Proposition 2
- Evaluating the quadratic forms $\sum_{r=1}^{R}\mathrm{mat}\left(\psi_r\right)^T K'_i\mathrm{mat}\left(\psi_r\right)$, collectively costing $O(RrNP + RNP^2)$:
  - When $m = 0, 1$, K'_m is diagonal, giving $R\,O\left(NP^2\right)$ multiplications
  - When $m > 1$, K'_m is low rank, giving $R\,O\left(r_m NP\right)$ multiplications
- Partial-tracing block diagonal matrices $\Xi\left(I\otimes K'_m\right)$, collectively costing $O(rNP)$:

- If $m = 0, 1$, $\Xi\left(I \otimes K'_m\right)$ is diagonal and the partial trace involves P multiplications of length-N vectors
- If $m > 1$, then $\mathrm{tr}_P\left(\Xi\left(I \otimes K'_m\right)\right)_{pp} = ||\Xi^{1/2}_{(p),(p)} G'_m||^2_F$, requiring diagonal multiplication and summation of P $N \times r_m$ matrices

Overall, the computational complexity is $O(NPR^2)$[3]; this omits the upfront eigendecomposition of K_1 and formation of $Y' := Q^T_{K_1} Y$ and $\{G'_m\}_{m>1}$, collectively costing $O(N^3 + N^2 P)$.

Finally, note that the M-step will have complexity depending only on M and P and thus will typically be negligible, even for complex penalties. For example, if penalties are omitted the cost is literally zero; if all VCs use a graphical lasso penalty, the cost will be roughly $O(MP^4)$ (and cheaper for sparse graphs, see, e.g., [Mazumder and Hastie, 2012]).

4.3.2 Repurposing EM computations for quasi-Newton optimization

Mixed models are typically optimized with EM and/or hill-climbing. Because EM naturally respects the geometry of the problem–gradient descent can easily escape the feasible set of PSD matrices, which can be a virtue if exact zeros are crucial [Listgarten et al., 2013; Speed and Balding, 2014] or innocuous if the PSD constraint is dropped–it is considered a robust means to initialize the problem. However, EM converges slowly and, when near an optimum, one Newton step can be equivalent to a large number of EM steps, motivating a hybrid approach [Zhou and Stephens, 2014].

[3]This assumes that $P < R$ and $MP^2 < NR^2$; both hold automatically for full-rank VCs, and the less-obvious latter also holds whenever $N > P^2$.

The log-likelihood derivative with respect to some scalar x parameterizing the C's is

$$\begin{aligned}
-2\nabla_x \ell(C|K,y) &= -2\nabla_x \left[\log|\Lambda_y| + \operatorname{tr}\left(yy^T\Lambda_y\right)\right] \\
&= \operatorname{tr}\left(\Lambda_y(\sum_m [\nabla_x C_m]\otimes K_m)\right) - \operatorname{tr}\left(yy^T\Lambda_y(\sum_m [\nabla_x C_m]\otimes K_m)\Lambda_y\right) \\
&= \sum_m \operatorname{tr}\left(([\nabla_x C_m]\otimes K_m)\left(\Lambda_y - \Lambda_y yy^T\Lambda_y\right)\right)
\end{aligned}$$

Then, using the fact that, for any X, $\operatorname{tr}(X) = \operatorname{tr}(\operatorname{tr}_P(X))$,

$$\begin{aligned}
-2\nabla_x \ell(C|K,y) &= \sum_m \operatorname{tr}\left(\operatorname{tr}_P\left(([\nabla_x C_m]\otimes K_m)\left(\Lambda_y - \Lambda_y yy^T\Lambda_y\right)\right)\right) \\
&= \sum_m \operatorname{tr}\left([\nabla_x C_m]\operatorname{tr}_P\left((I\otimes K_m)\left(\Lambda_y - \Lambda_y yy^T\Lambda_y\right)\right)\right)
\end{aligned}$$

using (2.2) to pull $[\nabla_x C_m]$ outside the partial trace.

Taking x to be the p,q-the entry of C_j, $\nabla_x C_m = I_{pq} I\{i=j\}$, and so

$$\operatorname{tr}\left(I_{pq}\operatorname{tr}_P\left((I\otimes K_j)\left(\Lambda_y - \Lambda_y yy^T\Lambda_y\right)\right)\right) = \operatorname{tr}_P\left((I\otimes K_j)\left(\Lambda_y - \Lambda_y yy^T\Lambda_y\right)\right)_{pq}$$

In turn, this means

$$\nabla_{C_m}\ell(C|K,y) = \frac{1}{2}\operatorname{tr}_P\left((I\otimes K_m)\left(\Lambda_y yy^T\Lambda_y - \Lambda_y\right)\right) \tag{4.9}$$

This is closely related to the update for Ω_m given in (4.7). In particular, EM without penalization sets $C_m = \Omega_m$ and, thus, the EM update for C_m can be rewritten

$$C_m \leftarrow C_m + \frac{2}{N} C_m \left[\nabla_{C_m}\ell(C_m|K,y)\right] C_m$$

and so EM is a quasi-Newton method on $\vec{C} := (\operatorname{vec}(C_0), \ldots, \operatorname{vec}(C_M))$ with fixed step size and preconditioner $H_{\mathrm{EM}}^{-1} := \oplus_m (C_m \otimes C_m)$.

More generally, this means that any method to compute Ω_m, e.g. that in Section 4.3, can be repurposed to perform any gradient-based update using the

identity

$$\nabla_{\vec{C}} \ell(C|Y,K) = \frac{N}{2} H_{\mathrm{EM}} \left(\vec{\Omega} - \vec{C} \right)$$

defining $\vec{\Omega}$ analogously to $\vec{C}$. Since this gradient has length only MP^2, preconditioning with some generic H^{-1}–for example, by using BFGS–is unlikely to affect overall runtime.

Importantly, though, the preconditioner may itself be expensive to compute. To my knowledge, when H is the Hessian, giving Newton steps, no published algorithm evaluates H for $M > 1$ and, when $M = 1$, the best algorithm costs $O(NP^6)$ [Zhou and Stephens, 2014].

4.3.3 Likelihood evaluation

Likelihood-ratio tests require the log-likelihood ℓ:

$$-2\ell(C|K,Y) \equiv -\log|\Lambda_y| + y^T \Lambda_y y$$

This can be computed in $O(P^3 + NPR^2)$ from the GLMM decomposition.

First, using (4.8) and the multiplicative property of the determinant,

$$\log|\Lambda_y| = 2N \log|T| + \log|\Xi - \Psi^T \Psi|$$

Sylvester's determinant theorem can be applied to the second term, giving

$$\log|\Xi - \Psi^T \Psi| = \log|\Xi| + \log|I - \Psi \Xi^{-1} \Psi^T|$$

The quadratic form can be handled similarly, using Proposition 2 in the first equality:

$$y^T \Lambda_y y = y^T \left(T \otimes Q_{K_1} \right) \psi_0 = \mathrm{vec}\left(Y' T \right) \psi_0$$

4.3.4 Out-of-sample prediction

In this section, we derive equations efficient algorithms for out-of-sample best linear unbiased predictors (BLUPs). We call using completely phenotyped samples to predict completely unphenotyped samples "prediction"; we use "imputation" to mean predicting sporadically missing entries. In both cases, we assume we have already estimated the C_m's from Y, which is natural for prediction as the C_m's can be learned from the fully-observed rows; for imputation, VCs from Ymust be learned either after case-wise-deletion (CWD), potentially dramatically reducing sample size, or after phenotype imputation (see Section 4.3.5).

For both prediction and imputation, we use BLUPs: given observed entries y_o and missing entries y_m, the BLUP is

$$\mathbb{E}\left(y_m|y_o, K, C\right) = \mathrm{Cov}\left(y_o, y_m|K, C\right)\mathbb{V}\left(y_o\right)^{-1} y_o \tag{4.10}$$

Out-of-sample prediction is comparatively easy because the Kronecker structure commutes with the subsetting operations. By assumption on the missingness patterns o and m, y_o can be reshaped into a fully observed matrix $Y_{O,} \in \mathbb{R}^{|O|\times P}$ and, similarly, y_m into $Y_{-O,} \in \mathbb{R}^{(N-|O|)\times P}$, so that

$$\begin{aligned}
\mathbb{E}\left(Y_{-O,}|Y_{O,}, K, C\right) &= \left(\sum_m C_m \otimes (K_m)_{-O,O}\right)\Lambda_y \mathrm{vec}\left(Y_{O,}\right) \\
&= \left(\sum_m C_m \otimes (K_m)_{-O,O}\right)\left(T \otimes Q_{(K_1)_{OO}}\right)\psi_0 \\
&= \sum_m (K_m)_{-O,O} Q_{(K_1)_{OO}} \mathrm{mat}\left(\psi_0\right) T^T C_m
\end{aligned}$$

which costs $O(Nr'|O|)$, assuming $|O| > P$.

The out-of-sample prediction setting is somewhat contrived as, in practice, one would almost always have covariates on the unphenotyped samples: we assume the

unphenotyped samples have been genotyped, and it seems unlikely that literally no other data would have been collected. When even one trait is measured on even one of the genotyped-but-not-phenotyped samples, the necessary Kronecker structures evaporate and this approach is no longer possible.

Related, prediction accuracies are dramatically lower than imputation accuracies. For example, prediction has essentially zero accuracy in the unrelated UKBS dataset, while imputation correlation (with CAR missingness) is roughly 70%. Prediction is intrinsically harder: when imputing a sporadically missing entry, one can leverage both inter-row and inter-column correlations, and Section 3 argues the latter are almost always stronger; this is clearly true in the UKBS. Out-of-sample prediction, however, relies solely on less-informative, inter-row correlation. Real datasets tend to have both sporadic missingness and block-wise missingness (e.g. the rats in Section 3.6.4), and an ideal method would combine the computational advantages of the algebraically simple prediction task with the performance benefits resulting from highly informative inter-trait correlations.

4.3.5 Phenotype imputation for sporadic missingness

As for phenotype prediction, we assume we have already estimated the C_m's and then predict missing data with BLUPs. We have not pursued an EM algorithm that iteratively fits the C_m and imputes the missing data, and in practice impute with phenix (though fitting the C_m after case-wise deletion of the phenotype matrix is also an option).

Though the bottlenecking $\mathbb{V}(y_o)^{-1} y_o$ computation in (4.10) no longer admits an obvious Kronecker structure, the inversion can be circumvented. Conjugate gradient (CG) computes this sort of expression (a.k.a. solves linear systems) with-

out inverting $\mathbb{V}(y_o)$ relying, instead, on repeatedly multiplying vectors by $\mathbb{V}(y_o)$. This approach is useful in situations, like ours, where inversion is expensive but multiplications are cheap.

More formally, the key computation is

$$\left(\left(\sum_i C_i \otimes K_i\right)_{oo}\right)^{-1} y_o$$

As written, this costs $O(|o|^3)$ which, for typical applications, will resemble $O(N^3P^3)$, which is prohibitive for moderate samples sizes and even a few traits. Fortunately, conjugate gradient reduces this inversion to repeated multiplications of the form

$$\left(\sum_i C_i \otimes K_i\right)_{oo} z$$

where z is a dummy variable changes throughout iterations of the algorithm–see Section A.5 for a review of conjugate gradient descent for square loss.

This multiplication can be performed quickly. Define $\dot{z}$ by padding z with zeros, so that $\dot{z}_o = z$ and $\dot{z}_{-o} = 0$, and

$$\mathbb{V}(y_o)\, z = (\mathbb{V}(y)\, \dot{z})_o = \left(\left(\sum_i C_i \otimes K_i\right) \dot{z}\right)_o = \sum_{m=0}^{M} \operatorname{vec}\left(K_m \dot{Z} C_m\right)_o$$

Each summand can be evaluated with sparse matrix multiplication by skipping padded entries of $\dot{Z}$:

- $m = 0$: $K_0 = I$, and so $\dot{Z}C_0$ costs $O(|o|P)$
- $m = 1$: Since K_1 is full-rank, $K_1\dot{Z}C_1$ costs $O(|o|N + NP^2)$
- $m > 1$: Since $K_m = G_m G_m^T$ for $G_m \in \mathbb{R}^{N \times r_m}$, $G_m G_m^T \dot{Z} C_m$ costs $O(\min(P, r_m)|o| + r_m NP)$

Doing this for all m costs $O(|o|N+NP(r+P)))$ per CG iteration, or $O(|o|P+rNP)$ when K_1 is omitted (Schur complements can be used to derive a similar method scaling with $|N-o|$ rather than $|o|$). When $|o| \propto NP$, this becomes $O(N^2P)$. Figure 4.1 shows the savings from this approach using $M = 1$, full-rank K_1 and 5% completely-at-random missingness.

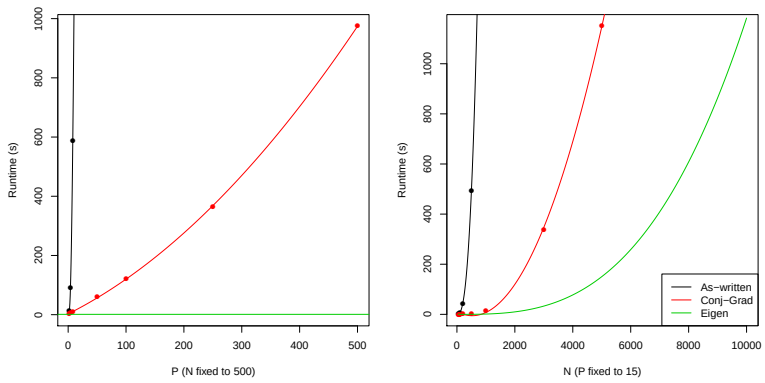


Figure 4.1: Runtime comparison for forming $\mathbb{V}(y_o)^{-1} y_o$ as-written (black), as in [Zhou and Stephens, 2014], or with our proposed padded conjugate gradient approach (red). Fitted curves, using the theoretical complexities of the two algorithms, are added in red. The time to eigendecompose K_1, which is necessary to learn VCs as we use full-rank K_1 here, is plotted for reference (green).

4.3.6 Runtimes of a simple `R` implementation

To address scalability, the EM algorithm was coded in `R` and run on data simulated from the model in (4.1). The Cholesky decomposition was performed using `Rcpp`; a full low-level implementation would be considerably faster.

Our baseline simulation uses a full-rank GRM and one additional, low-rank kinship built from r SNPs (so $M = 2$) and we fit the VCs by unpenalized maximum likelihood. We also take $N = 3,000$, $P = 10$ and $r = 20$. As we omit penalties here, VCs are full rank, and so $r'_m = P$ for all m and, thus, $R = rP$. The r_m

values are implicitly defined by r and M by assuming the r SNPs are spread evenly across the $M-1$ low-rank components–i.e. I take $r_m := \frac{r}{M-1}$. N, P, R and M are varied independently in the separate rows of Figure 4.2 (with r implicitly determined). The columns of this Figure assess runtime for the two key steps–whitening (eigendecomposing K_1 and forming Y' and the G'_m) and Cholesky decomposition–as well as overall runtime.

For clarity, we take exactly 100 steps per dataset; this is a form of early stopping. Though the number of steps for convergence is much higher and likely depends on the data and model parameters, EM is known to quickly converge to a neighborhood of the MLE [Zhou and Stephens, 2014], suggesting the 100-step estimates will be qualitatively similar to the MLE. In this sense, early stopping provides a fast and tuning-free approximation to the MLE.

More importantly, though, early stopping can in fact improve generalization performance by implicitly regularizing the MLE. This is well-established in neural networks, which are particularly prone to overfitting [Caruana, Lawrence, and Giles, 2000]. Intuitively, algorithm iterations trace a path in parameter space from the initialization to the MLE. This path increases in complexity from the clearly-underfitting initial values to often-overfitting MLEs. In principle, the optimal number of steps along this path can be learned by cross validation, similar to how the optimal penalization magnitude can be learned. In practice, the 100 EM step default appears widely applicable for GLMMs (perhaps because EM approximately converges quickly), and Section 4.4.2 shows the simple 100 EM step approach can often outperform the MLE in prediction.

Figure 4.2 displays the runtime results. Each point is a GLMM run and black dotted lines are polynomial regressions with degrees given by the complexities in

Section 4.3.1. Red dotted lines are linear regressions fit only to red points where Cholesky decomposition is the dominant computation and the linear approximation is good, and demonstrate that the complexity scales approximately linearly in N and P below, say, 5,000 and 100, respectively.

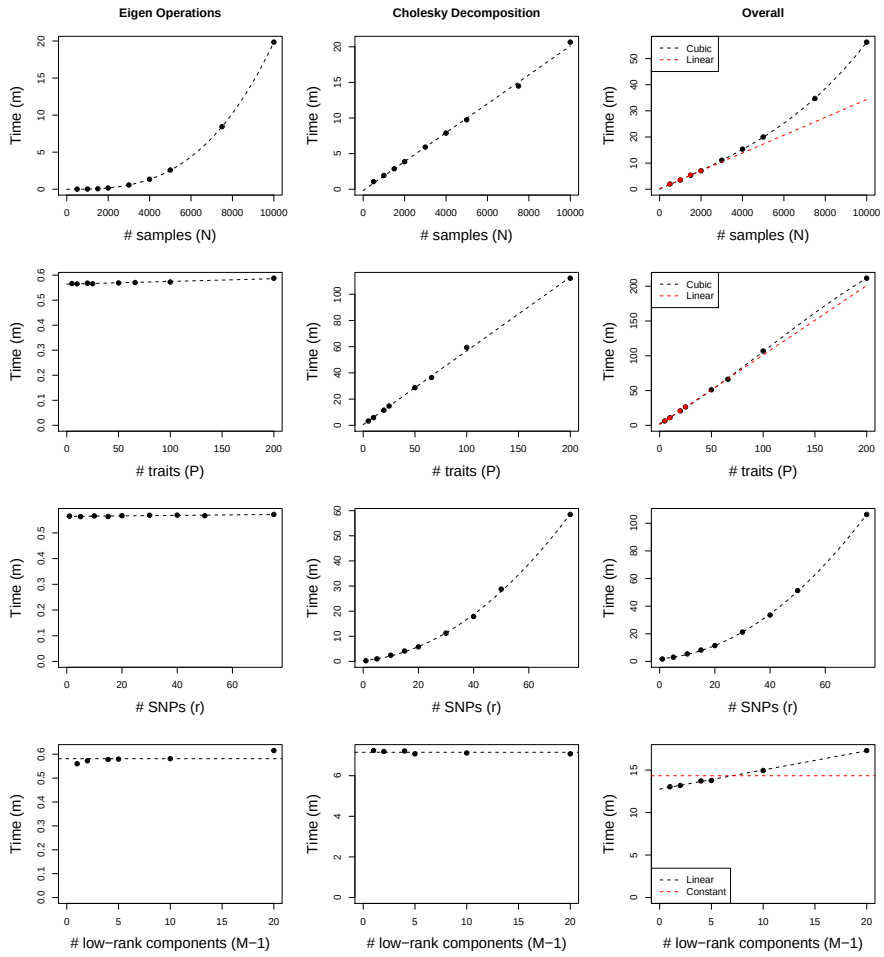


Figure 4.2: Runtimes for upfront eigendecomposition costs and 100 iterations of our EM algorithm on simulated data. Points represent one run of our method and black dotted lines are the fitted OLS polynomial regressions with degree given by the relevant computational complexity; red dotted lines linearly extrapolate runtime from small values of N or P, given by red dots. The eigendecomposition step, which includes whitening of SNPs and phenotypes, costs $O(N^2(N+r+P))$; the Cholesky step costs $O(NPR^2)$; the overall cost additionally includes $O(NP^2 + P^3)$ operations which become relevant for large enough N and P.

For typical parameter values, the Cholesky decomposition dominates both the practical runtime and the computational complexity. Thus any theoretical improvement to scalability or practical improvement to runtime of would require a

different square root of $\sum_{m>1} C_m \otimes K_m$, possibly by leveraging the Kronecker product structure of the summands, or through an entirely different approach, perhaps using CG as in [Loh et al., 2015a]–for example, the $\Lambda_y y$ term in (4.7) is easily computed by CG.

When $M = 2$, incorporating the Kronecker structure in the square root is natural [Casale et al., 2015]. In this case, the problem reduces to decomposing $C_2 \otimes K_2$, and the Cholesky decomposition commutes with the Kronecker product: letting $LL^T = C_2$,

$$C_2 \otimes K_2 = (LL^T) \otimes (G_2 G_2^T) = [L \otimes G_2]\,[L \otimes G_2]^T$$

When $M = 2$, our $O(NPR^2)$ Cholesky decomposition becomes $O(NPr_2^2 {r'_2}^2)$; the above decomposition of C_2, however, costs only $O(P{r'_2}^2)$. This example illustrates the general principle that specialized algorithms typically outperform more general ones, as our generic-M approach cannot leverage properties idiosyncratic to subproblems.

4.4 Applications

Our model and method opens the door to many extensions of established applications from single-trait, single-VC and/or unpenalized mixed models.

Generalizing a multi-trait method to multiple effects is primarily useful for flexibility, as additional components can incorporate any background signal. In genetics, these components may be used to absorb additional confounders or refine "genetic similarity" into meaningful notions of similarity. We address the former in Section 4.4.1 by performing a token simulation with heritability confounded both by common environments within-family and by population structure between-

families. We show that modeling this additional structure significantly improve heritability and genetic correlation estimates as well as predictive performance. Interestingly, including the family effect is more important for VC estimation, while the population effect is more important for imputation; in such situations, the ability to model all effects simultaneously is invaluable.

We then generalize gene-set tests, which ask whether SNPs in a gene collectively contribute to trait variance [Listgarten et al., 2013], to multiple traits. Using the NFBC dataset [Sabatti et al., 2008], we perform a token analysis by studying a locus known to be pleiotropic [Korte et al., 2012] and find that the multiple-trait test improves the signal. We note this analysis overlaps substantially with [Casale et al., 2015] and that their empirical results are genome-wide and far more thorough.

Next, we assess out-of-sample prediction in a variety of real datasets and find that multi-trait methods are not necessarily more accurate than single-trait methods, as the former easily overfit. We show, however, that (fast) ℓ_2 likelihood penalization can improve performance, as can computational regularization in the form of early stopping.

The above questions all can be addressed with simpler, published models, potentially at a cost in power. More interestingly, Section 4.4.3 asks an intrinsically multi-trait question: which phenotypes share a heritable basis [Stegle et al., 2011]? We fit maximum ℓ_1-penalized VCs to induce a sparse and heritable graphical model. We show improved graph recovery in simulations both over non-genetic approaches [Mazumder and Hastie, 2012] and an approximate approach that ignores environmental trait correlations [Stegle et al., 2011].

4.4.1 Coheritability estimation with multiple confounders

Data simulation

This section uses relatedness matrices based on the Human Connectome Project (HCP) [Van Essen et al., 2013] and simulated phenotypes. The HCP contains a mixture of unrelated people and nuclear families with known pedigrees. Though genotype data is not yet available, the "expected" GRM can be computed from the pedigree (assuming families are independent).

In this section, I simulate from and then fit the model

$$Y \sim \mathcal{MN}(0, I, C_0) + \mathcal{MN}(0, K, C_1) + \mathcal{MN}\left(0, ZZ^T, C_2\right) + \mathcal{MN}\left(0, xx^T, C_3\right) \tag{4.11}$$

where:

- $K \in \mathbb{S}_+^N$ is the expected GRM.
- $Z \in \mathbb{R}^{N \times F}$ is a matrix of family indicators: for each person i and family $f \in \{1, \dots F = 214\}$, $Z_{if} = I\{$ person $i \in$ family $f\}$. Z represents shared familial environment.
- $x \in [0,1]^{N \times 1}$ represents admixture proportions between two hypothetical populations. Unlike K and Z, x is simulated: for each family f, I draw $u_f \sim \text{Unif}(0,1)$ and set $x_i = u_f$ for individuals i in family f. This gives family members identical admixture proportions, which is approximately realistic.

As dictated by Section 1.4, these incidence matrices are then normalized which facilitates, in particular, easy comparison across C_m (otherwise, only (C_m, K_m) pairs are identified). Similarly, we assume that columns of Y have been scaled to unit variance as well as mean 0, and so variances and heritabilities coincide.

The VCs are all simply (normalized) Wisharts with minimal degrees of freedom:

$$C_m = h_m^2 \texttt{cov2cor}\left(\text{Wishart}(P, I_P)\right)$$

where the "heritability" h_m^2 of component i is constant across traits (note the environmental $i = 0$ also gets a "heritability" with this definition). The importance of multi-effect modelling will only increase for more disparate VCs. Finally, we choose $h^2 = (.5, .3, .1, .1)$: the population and family terms have the same size but are small compared to the GRM and pure-noise effects, and the deviation from the standard 2-component GLMM, a.k.a MPMM, is not dramatic.

Tested GLMM parameterization

We then simulate Y from model (4.11) and fit various GLMMs to it.

Specifically, the GLMMs we consider are:

base $C_1 \leftarrow \frac{1}{N} Y^T Y$; this entirely ignores genetics

grm $C_1 \leftarrow \text{GLMM}\left(Y, K\right)$; this standard MPMM uses only the GRM and makes no attempt to control for confounders

fam $C_1 \leftarrow \text{GLMM}\left(Y, K, Z\right)$; this models family structure only

pop $C_1 \leftarrow \text{GLMM}\left(Y, K, x\right)$; this models population structure only

all $C_1 \leftarrow \text{GLMM}\left(Y, K, Z, x\right)$; this is the generative model and captures all confounders

The hierarchy among these models is displayed in Figure 4.3. Starting with the degenerate "base" model that completely ignores genetics, "grm" adds a genome-wide, heritable component, enabling heritability estimation and out-of-sample prediction. But "grm" fails to account for two levels of confounding, family and

population structure: these are addressed, respectively, by the "fam" and "pop" models, which each partially correct (co)heritability estimates for the confounding structure.

But even these two models, which have 3 VCs and are richer than the standard 2 VC mixed models in genetics, fail to account for both types of confounding, and only the 4 VC "all" model is rich enough to capture all salient features of the data. (The 3 VC models can be fit "efficiently" by [Casale et al., 2015] but, to our knowledge, only the tools in this thesis can efficiently fit more than 3 VCs.)

Results

We compare the models in terms of genetic correlations ($\texttt{cov2cor}(C_1)$), heritability ($\text{diag}(C_1)$) and prediction, all shown in Figure 4.3. First, increasing model complexity generally move genetic correlation estimates (small heatmaps) toward the true genetic correlation (large heatmap). However, the "pop" model yields much worse estimates than the "fam" model, which is essentially as accurate as the "all" model. This is because the family structure is far more similar to the GRM than the population structure–in fact, for clonal families "fam" and the GRM are identical. A similar story holds for heritability estimation: the GRM heritability overzealously explains the family structure when "fam" is omitted, leading to consistent upward heritability bias; conversely, while including "pop" improves accuracy by explaining residual variation, it has negligible impact on bias. Though only one simulated dataset is presented, these qualitative results held in five (of five) additional simulation replicates.

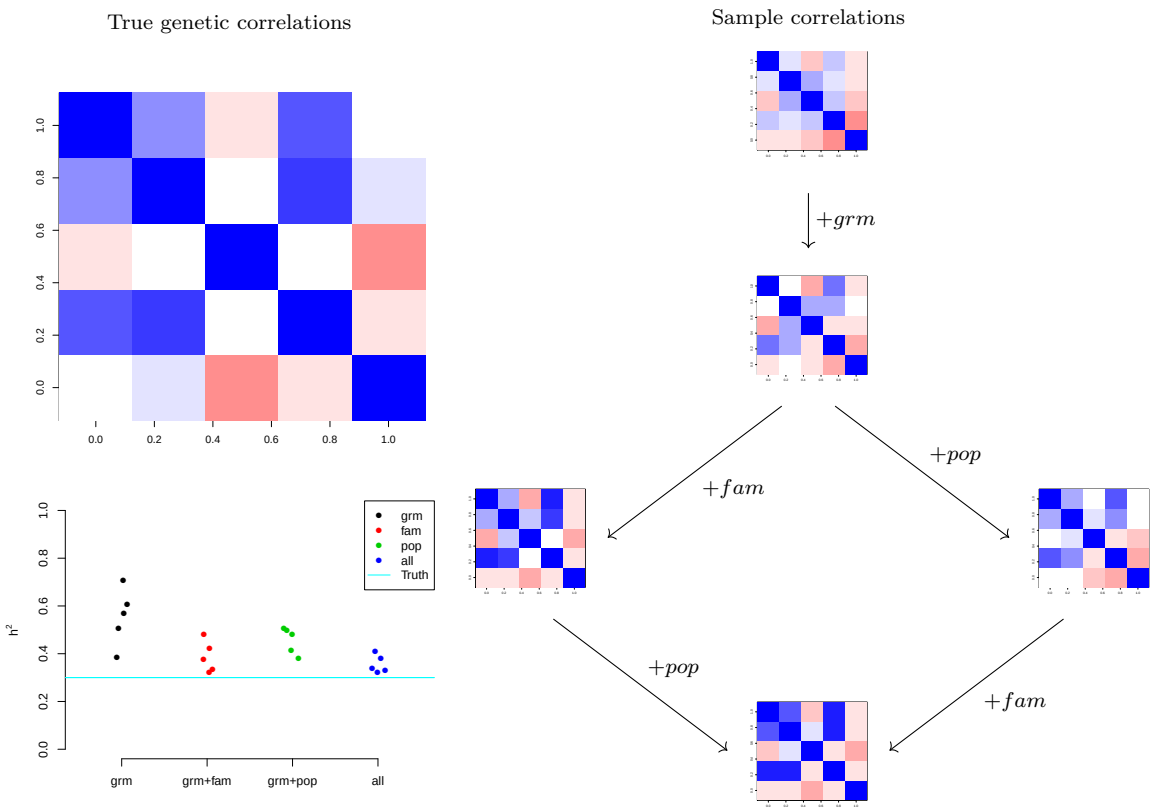
In addition to VC estimation, richer GLMMs can also improve phenotype prediction. Repeating the above simulation to create Y, we then masked 10% of the

samples and predicted them with BLUPs. The table at the bottom of Figure 4.3 summarizes the prediction accuracy and runtime of each GLMM, averaged over 24 independently simulated Y matrices. Unsurprisingly, richer GLMMs tend to perform better.

However, the relative importance of "fam" and "pop" has flipped: "pop" gives twice the improvement of "fam". This reversal is natural: for prediction, the goal is to introduce novel information beyond the GRM; for VC estimation, confounder correction is best accomplished with information maximally related to the GRM. At the extreme where the confounding incidence matrix is identical to the GRM, including the confounder reveals the impossibility of coheritability estimation but does not affect prediction performance at all.

Computational GLMM scaling, too, is evident from the simulation. While, unsurprisingly, more complex GLMMs are more expensive, this cost depends heavily on the rank(s) of the additional VC(s), which dictate r in the complexities stated in Section 4.3. For example, modelling the rank-one x increases runtime by an order of magnitude; including the rank-214 Z, however, increases runtime by more than two orders of magnitude. Fortunately, though, these costs add rather than multiply (as Section 4.3 indicates), and fitting the full model is only twice as expensive as "fam". Practically, "pop" is a reasonable choice for prediction; in fact, it may even outperform "full", which can overfit, in MSE.

These simple simulations confirm the obvious result that unmodelled confounders can bias heritability estimates and that this bias can be corrected with richer models. Moreover, richer models can improve prediction performance by exploiting structure missed by the simple "grm" model. Generally, GLMMs may improve both bias and variance of heritability estimators by modelling known



	grm	fam	pop	all
Time (m)	0.35 (0.01)	95.77 (3.11)	2.7 (0.01)	180.61 (0.28)
Prediction Corr	0.39 (0.01)	0.42 (0.01)	0.46 (0.02)	0.47 (0.02)
Prediction MSE	0.85 (0.03)	0.82 (0.03)	0.78 (0.02)	0.77 (0.02)

Figure 4.3: Four variance component simulation results. Top left: the true matrix of genetic correlations used to simulate the data. Right: estimated genetic correlations using increasingly rich random effect structures approximate the truth increasingly well. Center left: estimated heritability for the four random effect structures with a heritable component and the five phenotypes. Bottom: out-of-sample prediction accuracy and runtimes. Note that in VC analyses including "fam" is more beneficial than including "pop"; the opposite is true for prediction accuracy.

confounders as random effects, and modelling more than one confounder enables simultaneous optimization of both the VC estimation and out-of-sample prediction tasks.

Interestingly, confounders can have drastically different impact on different aspects of the model fit: if heritability estimation (prediction) is the goal, "pop" ("fam") is unequivocally more important to incorporate. The approach of [Casale et al., 2015] can efficiently fit either of these 3 VCs models but not the 4 VC "all" model, putting the onus on the practitioner to choose which confounder to model and which to ignore. This choice takes time and, more importantly, depends on the mixed model application (e.g. heritability estimation versus prediction). Thus the ability of GLMMs to efficiently fit multi-effect VCs not only improves performance by fitting a richer (and, in this case, correct) model but also eliminates a significant practical dilemma.

4.4.2 Penalized, multi-trait models for phenotype prediction

As in [Rakitsch et al., 2013], the goal in this section is to predict the phenotypes of entirely unobserved samples–$Y_{-O,}$–using entirely observed samples–$Y_{O,}$–from relatedness matrices (constructed from genotype data) spanning both the observed and unobserved samples. This is distinct from phenotype imputation, where the missing entries of Y are scattered throughout the matrix. But the approach is otherwise similar to the imputation analyses in Section 3.5.3, except we mask entire rows of the phenotype matrix rather than entries.

We apply this approach to 8 real datasets. The UKBS, NSPHS, yeast, wheat and rat datasets are exactly as in Section 3.5.3; we drop the chicken dataset because of its extreme level of missingness ($> 50\%$). We add the Northern Finland Birth Cohort (NFBC) [Sabatti et al., 2008], using $N = 5,144$ samples and $P = 4$ phenotypes (CRP, HDL, LDL, and TG). We also use two homogeneous subsets of the HCP phenotypes, each measured on $N = 460$ samples: "brain," consisting of $P = 105$ fMRI measurements, and $P = 26$ standard "biometric" traits. Many of these datasets have sporadic missingness; these are problematic for GLMMs, and so we impute with phenix before fitting GLMMs.

In addition to LMM and MPMM (which is run for 100 EM iterations), both described in Section 3.4, I test the following methods:

- MPMM_short terminates MPMM after 30 EM iterations
- MPMM_long terminates MPMM after 10,000 EM iterations
- MPMM_penal runs MPMM (for 100 EM iterations) with ℓ_2 penalties on both genetic and environmental VCs. Penalty magnitudes are chosen by

cross-validation, iteratively updating each VC's parameter in turn until convergence; this offered a slight improvement over using the same penalty for both VCs (not shown)

Results are displayed in Figure 4.4. I assess accuracy using (signed) squared-correlation, roughly following [Rakitsch et al., 2013; Tucker et al., 2015], and I plot change relative to LMM (so LMM is a horizontal line at 0). Due to the large and varied computational expense, each (dataset,training sample size,method) triple has been run for a different number of simulated missingness patterns, though all have run for at least 200 (except MPMM_penal for the largest training sample sizes in the HCP "brain" and NFBC datasets).

There is no generally dominant method. This is less surprising when LMM is interpreted as a penalized MPMM: the non-convex barrier penalty gains advantages of parsimony without biasing the crucial heritability estimates. With this interpretation, we expect LMM to do well when $\frac{N}{P}$ is small, which roughly holds: as training size increases within-panel, all MPMM methods always improve. However, Figure 4.4 is sorted by aspect ratio and no clear trend in relative performance is evident, suggesting other dataset parameters–like overall inter-trait correlation, heritability and relatedness–are generally more important. For example, the HCP "brain" dataset, which has high inter-trait correlation, unsurprisingly prefers MPMM. Conversely, NFBC, with $P = 4$ and unrelated, albeit homogeneous, human samples, requires roughly 4,000 training samples before MPMM outperforms LMM.

Further, we find that the computational regularization in the form of early stopping typically improves results, especially for small aspect ratios where overfitting

is most likely. Conversely, only in wheat–with very large inter-trait correlation–does running MPMM until convergence (MPMM_long) improve accuracy. When the aspect ratio is large, convergence is quick and all stopping criteria give similar performance (not shown). Together, it is clear that MPMMs typically need not be run to full convergence, and in such cases EM–which nears optima quickly but converges slowly–is an attractive optimization approach.

Finally, penalization always improves prediction accuracy, though this effect wanes as the amount of training data grows–asymptotically, MPMM_penal automatically picks no penalization. Nonetheless, the simpler penalization implicitly employed by LMM often outperforms the complex and computationally expensive multi-trait cross-validation used in MPMM_penal, and the employed ℓ_2 penalization is not enough to guarantee that multi-trait models outperform LMMs for prediction.

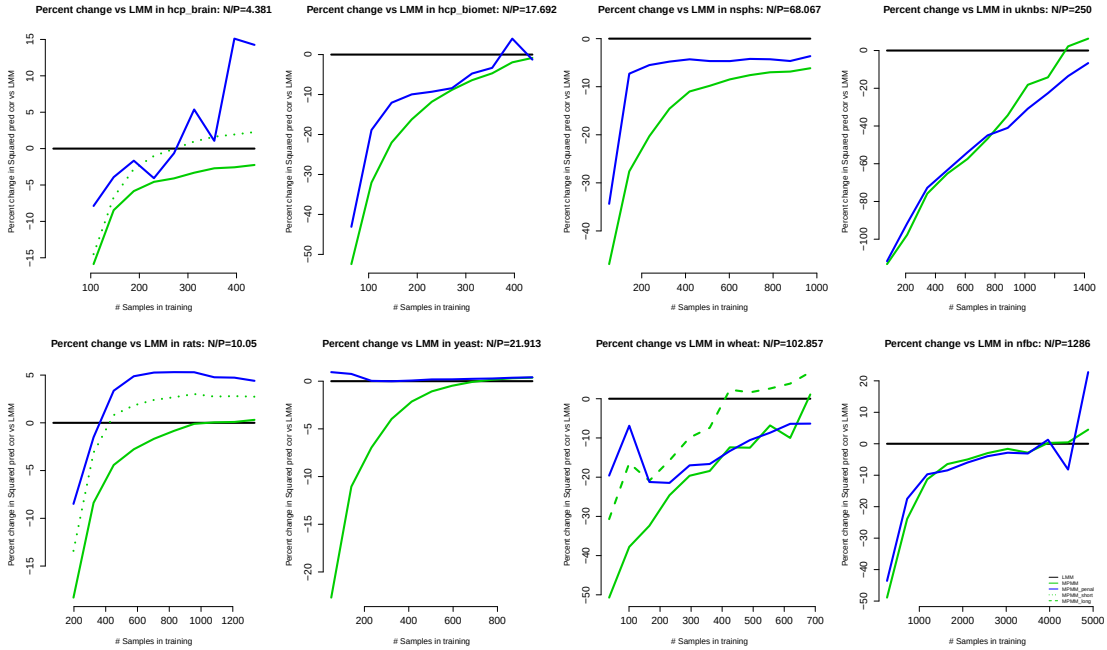


Figure 4.4: Real out-of-sample prediction accuracy for mixed model BLUPs. Accuracy is measured relative to LMM in terms of squared imputation correlation. No method is dominant, though penalization in the form of early stopping and/or ℓ_2 likelihood regularization consistently improve the performance of MPMM.

4.4.3 Heritable graphical model estimation

The basic approach in this section, pioneered by [Stegle et al., 2011], is to estimate a sparse genetic precision matrix, i.e. the inverse of the GRM VC. This is useful because sparse, Gaussian precision matrices naturally correspond to sparse undirected graphs: entry (i, j) is zero if an only if variables i and j are conditionally independent given the others [Lauritzen, 1996].

Specifically, we model Y with a standard MPMM written in terms of precision matrices $\Theta_g := C_1^{-1}$ and $\Theta_e := C_0^{-1}$:

$$Y \sim \mathcal{MN}\left(0, K, \Theta_g^{-1}\right) + \mathcal{MN}\left(0, I, \Theta_e^{-1}\right) \tag{4.12}$$

We then optimize the ℓ_1-penalized log likelihood:

$$\left(\hat{\Theta}_g, \hat{\Theta}_e\right) \min_{\Theta_g, \Theta_e \succeq 0} -\ell(Y|\Theta_g, \Theta_e, K) + \lambda\|\Theta_g\|_1 \tag{4.13}$$

Because the ℓ_1 norm is non-smooth and convex, this problem is manageable and yields sparse $\hat{\Theta}_g$. We call this special case of GLMMs G3M (Genetics Gaussian Graphical Model) and refer to the Gaussian graphical model (GGM) implied by Θ_g as a"heritable graphical model," though the precise biological interpretation of this graph is not obvious.

This approach is closely related to the graphical lasso [Banerjee, El Ghaoui, and d'Aspremont, 2008; Yuan and Lin, 2007], a common approach to estimate high-dimensional GGMs that is obtained by setting $K = I$ and removing the environmental Θ_e in (4.12). This well-known approach has sophisticated implementations, and we use the package `dpglasso` [Mazumder and Hastie, 2012] both to fit the graphical lasso and to evaluate GLMM M-steps. Algorithm development for this problem and generalizations to penalties inducing structured graphs are active areas of research, and GLMMs can easily incorporate any new algorithms or penalties.

In the presence of noise, the graphical lasso ought to fail: if (4.12) holds and Θ_g is sparse, it is not generally true that $\mathbb{V}\left(Y_{i,}\right)^{-1}$ is sparse. This is not surprising, but demonstrates that even small noise contributions not only cause estimation error but fundamentally change the structure of interest.

Other generalizations of the graphical lasso to non-iid data exist. If we again set $\Theta_e^{-1} = 0$ but allow K to be a general, learned covariance matrix, we obtain the model from [Leng and Tang, 2012]. Related, [Kalaitzis et al., 2013] also learns row- and column-specific GGMs, but connects them through the Kronecker sum

of these precisions rather than the Kronecker product. The GLMM decomposition leans heavily on the partial trace, which was introduced to us by [Kalaitzis et al., 2013].

A very closely related model, which inspired this section, is considered in [Stegle et al., 2011]. Their model constrains Θ_e to be spherical and, thus, all variable correlations must be explained as genetic; they also learn K (as a superposition of low-rank and spherical matrices), but our implementation of this model below uses the known K. Another difference is algorithmic: they use a first-moment approximation to an EM algorithm, while we derive exact EM updates.

Simulations

We performed simple simulations to demonstrate the benefit of our more general approach when model (4.12) holds. We compare to the vanilla graphical lasso and KronGlasso from [Stegle et al., 2011]. We use a variant of KronGlasso, implemented by Victoria Hore, to use the known K for fair comparison to our approach.

Similar to [Stegle et al., 2011], we simulated 40 phenotype matrices with $N = 400$ and $P = 50$ from (4.12). We use a "Sibs" K matrix, as in Section 3.5.1, and various established choices for the Θ matrices. Summarized in Table 4.1, each Θ choice gives a different level of sparsity: At one extreme, Θ_e is spherical, and so the KronGlasso model holds exactly and the glasso model is minimally violated; at the other extreme, Θ_e is dense. VCs are are transformed, as in Section 3, first with `cov2cor` and then by rescaling so each trait has heritability 0.17.

Each tested method requires a regularization parameter – λ in equation (4.13) – controlling overall sparsity. We vary λ by setting $\lambda = 5^x$ for x linearly interpolated between -7 and 3 and, for each value of λ, compute power and type I

Θ_g	Θ_e
AR(1)	AR(1)
Random(1%)	Wishart $\left(P-3, I_P/\ (P-3)\right)$
Random(10%)	iid

Table 4.1: Different choices for precision matrices. For AR(1) we use an autocorrelation of 0.8. Random(p) in defined by two steps: a fraction p of off-diagonal entries are set to a nonzero constant, and then a multiple of the identity is added such that the resulting condition number of the matrix is exactly P, as in [Leng and Tang, 2012; Kalaitzis et al., 2013].

error for the resulting graph. Figure 4.5 presents receiver-operating-characteristic (ROC) curves for Glasso, KronGlasso and G3M, each averaged over all 40 datasets. Each panel corresponds to a (Θ_g, Θ_e) pair; rows and columns index Θ_g and Θ_e, respectively.

Unsurprisingly, we substantially outperform competitors for dense Θ_e (left column of Figure 4.5) and slightly underperform, due to overfitting, for spherical Θ_e (right column of Figure 4.5). Surprisingly, though, glasso outperforms KronGlasso for spherical Θ_e – where KronGlasso's model holds exactly; this is presumably because of the approximations in the EM algorithm, and the difference with results in [Stegle et al., 2011] is presumably because our choice for R is far less structured.

While the expected results are obtained in the left- and right-columns of Figure 4.5, the center column–where Θ_e, as well as Θ_g, is sparse–no method is clearly dominant; that said, G3M does perform much better for low type I error rates, the typical regime of interest. This suboptimality is not shocking, as our choice for regularization was motivated by the assumption that only Θ_g is sparse.

Nonetheless, we can ℓ_1-penalize Θ_e as well as Θ_g. That is, we solve

$$\min_{\Theta_g, \Theta_e \succeq 0} -\ell(Y|\Theta_g, \Theta_e, K) + \lambda\|\Theta_g\|_1 + \gamma\|\Theta_e\|_1 \tag{4.14}$$

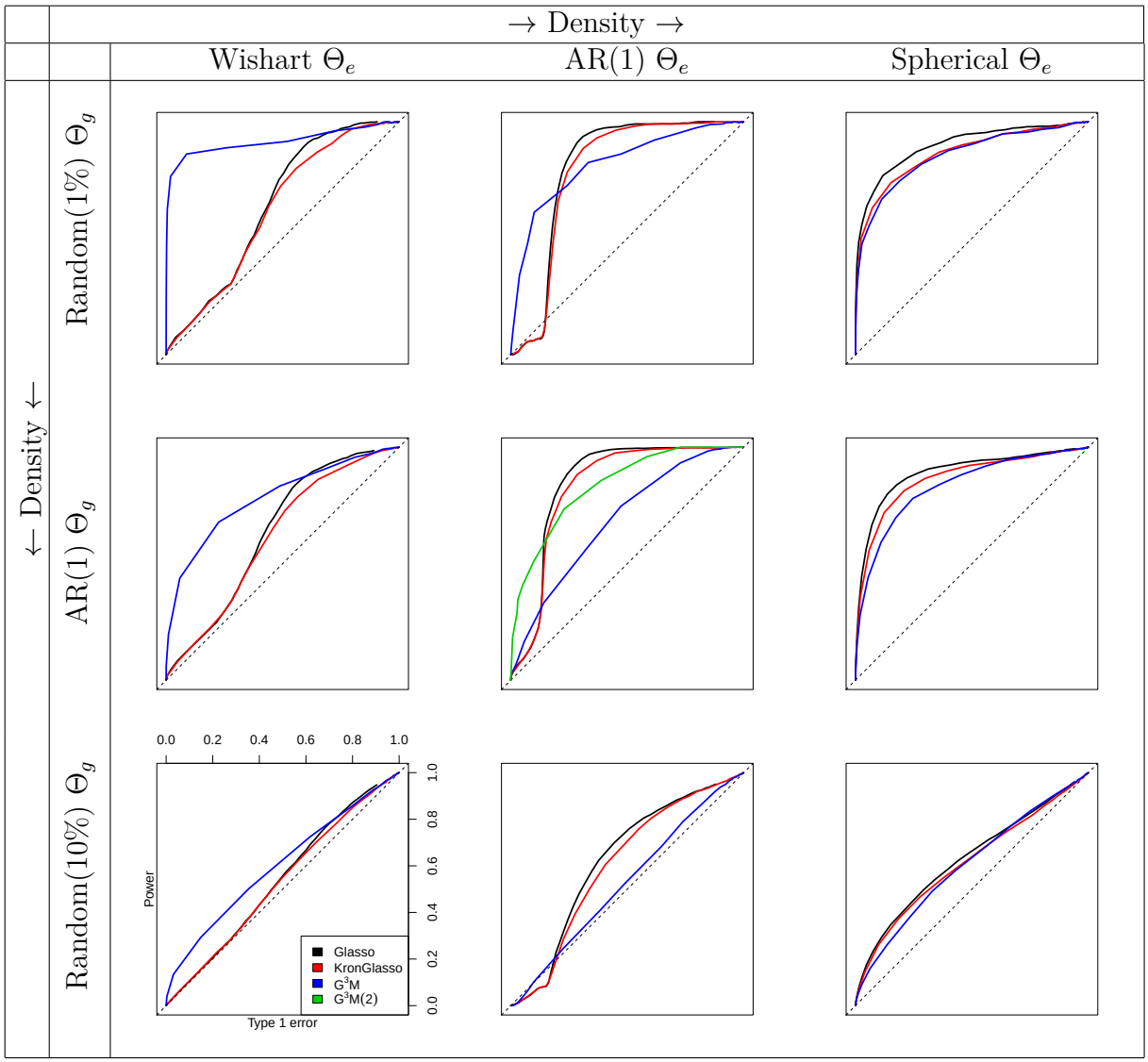


Figure 4.5: Comparison of methods for graphical model recovery with various heritable precision Θ_g and environmental precision Θ_e.

Note that γ and λ are distinct, and so this requires a 2D parameter search. We plot a 1D regularization path in Figure 4.5 by profiling out γ: for each λ, we choose γ to minimize the difference between the estimated and true Θ_g. The green line in Figure 4.5 thus uses two shortcuts: this choice of γ is infeasible in real data (but presumably could be replaced by some cross-validation procedure) and we only run 1, rather than 40, datasets. Despite these strong caveats, it seems clear that regularizing Θ_e can be valuable.

Instead of averaging accuracy over 40 datasets, we can pick one dataset and produce graphs for G3M, KronGlasso and glasso. The dataset was generated using the Random(1%) model for Θ_g and the Wishart model for Θ_e (the (1,1) panel in Figure 4.5), which we feel is realistic. λ was picked for each method to yield 70% power, shown by the dashed horizontal line in the ROC plot in Figure 4.6. Though the ROC shows G3M performs best, the actual graphs demonstrate how drastically the methods differ. Only G3M returns a result that is remotely useful: though an erroneous red edge is found and some black edges are missed, our method recapitulates the qualitative truth that only a few variables are connected while retaining a low false positive rate. (Thanks to Victoria Hore for code to plot these graphs.)

We note that extending this analysis to real data is non-trivial. Here, we chose the regularization based on oracle knowledge of the true graph. In practice, however, choosing a single regularization parameter requires optimizing some (out-of-sample) criterion–though we are not aware of criteria yielding consistent estimates of the graph–or sample-splitting approaches, which all require iid data to my knowledge–even permutation testing is non-trivial in the presence of genetic relatedness [Abney, 2015].

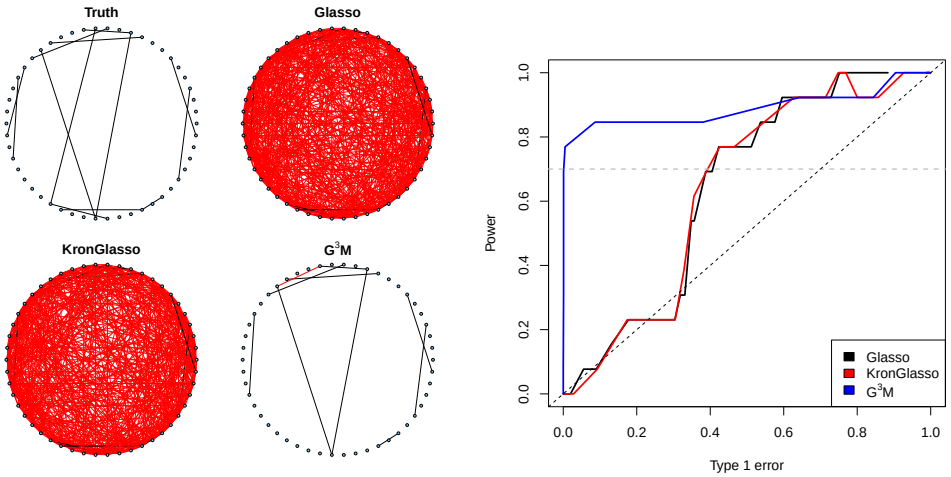


Figure 4.6: GGM estimation on simulated datasets with Wishart noise. Left : reconstructed graphs at 70% power. Right: ROC curves for each method. The dashed line is drawn at 70% power.

4.4.4 Multiple trait gene tests can increase power

In this section, we use the model

$$Y \sim \mathcal{MN}(0, K, C_1) + \mathcal{MN}\left(0, GG^T, C_2\right) + \mathcal{MN}(0, I, C_0) \tag{4.15}$$

where K is a GRM as usual and $G \in \mathbb{R}^{N \times r}$ is a matrix of the r SNPs in a region of interest. For gene tests, the region is a gene, and we study, in particular, FADS1-2.

We perform a simple likelihood ratio test, which fits both the above model and its null variant that restricts $C_2 = 0$. As C_2 is on the boundary of the feasible set, ordinary LRT asymptotics are inappropriate. Following [Listgarten et al., 2013], we approximate the null distribution of the LRT with extensive permutations: we permute rows of G–disrupting any conceivably real signal while maintaining LD between SNPs–and re-compute the LRT. We do this 1,000 times, and thus obtain 1,000 draws from the null LRT distribution.

Rather than use this non-parametric distribution, which cannot yield p-values below $\frac{1}{1{,}000}$, [Listgarten et al., 2013] fits a parametric form to these null LRTs. Specifically, (π, a, d) are fit to the null LRT distribution with a $(\pi, 1-\pi)$ mixture of a χ^2_0 (a point mass at 0) and a $a \cdot \chi^2_d$. This parametric form enables computation of far smaller p-values; note, however, this approximation is fit to LRTs with $p > 10^{-3}$, and extrapolation to much smaller p is not obviously reliable. Nonetheless, we adopt this established approach below.

We pre-processed the NFBC data as in [Zhou and Stephens, 2014]. We performed two-trait association tests on low-density lipoprotein (LDL) and triglycerides (TG) for each SNP in FADS1-2 with `gemma` [Zhou and Stephens, 2014] and plotted the resulting p-values in Figure 4.7 after Bonferroni correction. We then performed our two-trait gene test using exactly the same SNPs and represent the resulting p-value as a horizontal line as the signal corresponds to the entire locus. The set test improves power by nearly one order of magnitude. However, without Bonferroni correction the most-significant SNP is (slightly) more significant than our gene test; whether multiple-test reduction is the only reason the gene test increases power is biologically important but, statistically, moot. Very similar results were obtained by [Casale et al., 2015] using a very similar method.

4.5 Future GLMM applications

As mentioned before, all established mixed model applications can be performed with GLMMs. For example, compressed GRMs (see Section 5.2.4) can easily be incorporated, as can other notions of genetic relatedness, e.g. based on haplotypes [Bhatia et al., 2015], pedigrees [Zaitlen et al., 2013] or genomic annotation [Finucane et al., 2015].

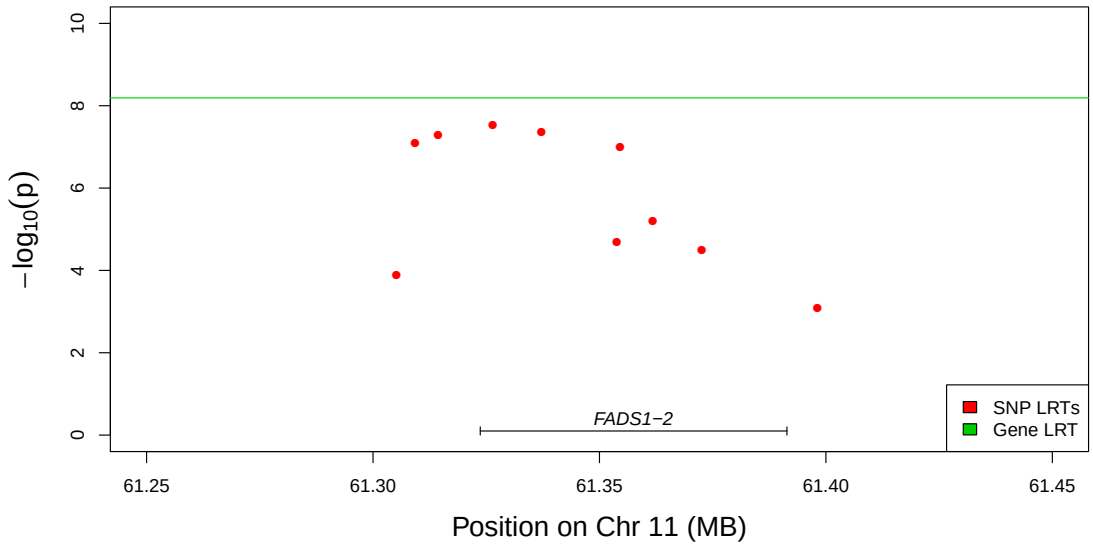


Figure 4.7: Association test results for the FADS1-2 gene and LDL and TG in the NFBC dataset. SNP-based LRT tests are performed for each SNP in the region, and the gene-based LRT test is performed on all SNPs jointly.

Another new possibility is multi-gene random effect testing, perhaps using greedy forward selection (as is common in ordinary regression) to improve association power by better modelling noise background. This is a natural extension of [Casale et al., 2015] and related to greedily building the GRM from large-effect SNPs [Listgarten et al., 2012]. In addition to testing these genes, a looser significance threshold could be used for forward selection to define a larger set of genes which could be brought forward as a rich background (as [Speed and Balding, 2014] did for genomic regions), which may improve power for fixed-effect testing and prediction accuracy.

Chapter 5

Compressive mixed models for high-dimensional traits

5.1 Introduction and motivation

This section introduces a new multi-trait model that generalizes both the phenix model and the GLMM. The data is again as described in Section 1.4: a partially complete phenotype matrix $Y \in \mathbb{R}^{N \times P}$ of P phenotypes measured on N individuals; a known kinship matrix, or GRM, $K \in \mathbb{S}_+^N$, summarizing genome-wide similarity; and k known random effect incidence matrices, $\{G_m\}_{m=1,\ldots,k}$, with corresponding linear kernels $K_m := G_m G_m^T$. Again, all matrices are assumed normalized. A difference, however, is that CMMs will fit much larger P: rather than dozens of traits for GLMMs or a couple hundred for phenix, CMMs can fit thousands. Additionally, sporadically missing entries in Y are naturally handled (unlike in GLMMs).

CMMs are naturally introduced as a modification of phenix, as the Bayesian model (discussed in Section 5.2) and variational Bayesian algorithm (discussed in Section 5.3) are redolent of the phenix approach. However, CMMs are better understood as dimensionality-reducing GLMMs, as they both incorporate more

general notions of relatedness and can partition (co)heritability. In Section 5.4 I show CMMs can outperform both phenix and GLMMs for phenotype prediction in 5 real datasets, albeit by a small margin. In Section 5.5, I return to the rat GWAS from Section 3.6.5 and show that GWAS on latent CMM factors uncovers novel and biologically plausible loci. Standard mixed model applications–e.g. heritability estimation, imputation, and genomic prediction–can also be accomplished with CMMs; the prediction results, which rely on accurate heritability estimation and imputation and are closely related to genomic prediction, shed some light on these applications.

5.2 The compressive mixed model

CMMs are a simple modification of the probabilistic matrix factorization model (PMF, see Section 3.2.1), replacing the spherical prior on rows of the rank-M scores matrix S–which can be interpreted as latent phenotypes–with a GLMM prior:

$$\begin{array}{rcl} Y|S,\beta,\nu & \sim & S\beta + \mathcal{MN}\left(0, I, \nu^{-1}\right) \\ S|C & \sim & \sum\limits_{m\geq 0} \mathcal{MN}\left(0, K_m, C_m\right) \\ \beta|\tau & \sim & \mathcal{MN}\left(0, I, \tau^{-1}\right) \\ \tau_p|a,b & \overset{\text{iid}}{\sim} & \text{Gamma}\left(a,b\right) \\ \nu_p|c,d & \overset{\text{iid}}{\sim} & \text{Gamma}\left(c,d\right) \end{array} \tag{5.1}$$

τ and ν are diagonal precision matrices, and to simplify notation, I use $\tau/\text{diag}(\tau)$ (and $\nu/\text{diag}(\nu)$) interchangeably.

5.2.1 Identifiability

There is a rotation non-identifiability in the CMM model. Essentially, one can rotate the latent factors (that is, S, β and the C matrices) without affecting

the model; this is formalized in Proposition 4 below as choosing a rotation R. This problem generalizes the well-known permutation non-identifiability arising in factor and mixture models, in which case R must be a permutation (a subgroup of the rotations).

In standard matrix factorization (MF) models, factors are identified as C (there is only one C in standard VBMF) is restricted to be diagonal (this assumes C has no repeated eigenvalues). This is no free lunch, however; rather, it represents an implicit choice of R that always exists and is unique when C has no repeated eigenvalues. That is, a canonical choice for the rotation matrix R is made *a priori* without loss of generality.

phenix goes one step further, requiring C not only to be diagonal but in fact spherical, which does sacrifice generality. All eigenvalues of the identity are 1, and now factors are independent for *all* choices of R. This means R is not unique and non-identifiability re-emerges. This is no problem for the applications in Section 3, which depend only on the (identified) product $S\beta$; this is also true for many CMM applications, e.g. imputation, prediction and heritability estimation.

In summary, both phenix and standard MF place matrix-normal priors on S, and thus columns of S can be assumed independent without loss of generality. CMMs, however, use a richer GLMM prior on S and, thus, no R can achieve independent components (in general)–only one C can be freely diagonalized. This eliminates the obvious canonical choice for R, leaving CMMs unidentified. – Yikes paragraph

Rather than resolve this equivalence class by choosing R–as implicitly done in typical PMF and impossible in phenix–I ignore this degeneracy and let R be automatically determined by the below VB algorithm. As the algorithm is initialized

from an SVD–which, by construction, has independent components–it is intuitive expected to converge to a "relatively-uncorrelated" representation of the factor space. Fortunately, faith in this interpretation is not required for downstream results, as all representatives are equivalent to the CMM; this does not, however, mean my determination of R is in any sense optimal.

One may choose R to minimize genetic correlation between factors, as in [Klei et al., 2008]; in a CMM, this simply means letting R be eigenvectors of C_1. This may be useful for GWAS, as one wishes to avoid splitting a factor-locus association across multiple components, which decreases power and adds superfluous tests. Indeed, I observe this factor splitting twice in the GWAS in Section 5.5; nonetheless, this same analysis also demonstrates that simply initializing CMMs via SVD is an effective strategy. Investigating principled choice of CMM representative may be interesting for future work, but fundamentally cannot be done without injecting prior information–from this perspective, my simple, implicit strategy is as objective as any.

Formal rotation invariance

Define the equivalence relation $\sim$ by $(S, \beta, C) \sim (S', \beta', C')$ if

$$\begin{aligned} S\beta &= S'\beta' \text{ and} \\ s^T \Lambda_0 s &= s'^T \Lambda_0' s' \text{ and } \log|\Lambda_0'| = \log|\Lambda_0| \text{ and} \\ \text{tr}\left(\beta\tau\beta^T\right) &= \text{tr}\left(\beta'\tau\beta'^T\right) \end{aligned}$$

If this equivalence holds, then $P(S, \beta, C|Y, \tau) = P(S', \beta', C'|Y, \tau)$: the first, second and third lines above ensure the likelihood, prior on S and the prior on β are constant, respectively. I assess this notion of equivalence, rather than equal posterior

probability (a weaker notion), only for simplicity.

Proposition 4. *Assume* $(S', \beta', \{C'_m\}) := (SR, R^T\beta, \{R^T C'_m R\})$ *for some rotation* R*. Then* $(S, \beta, C) \sim (S', \beta', C')$

I refer to R as the choice of latent factors.

Proof. Each of the equivalence relation identities is easily verified:

$$S'\beta' = SRR^T\beta = S\beta$$

$$s'^T\Lambda'_0 s' = [(R \otimes I)\, s]^T \left[(R \otimes I)^T \Lambda_0 (R \otimes I)\right] [(R \otimes I)\, s] = s^T\Lambda_0 s$$

$$|\Lambda'_0| = |\left[(R \otimes I)^T \Lambda_0 (R \otimes I)\right]| = |\Lambda_0|$$

$$\mathrm{tr}\left(\beta'\tau\beta'^T\right) = \mathrm{tr}\left(R^T\beta\tau(R^T\beta')^T\right) = \mathrm{tr}\left(\beta\tau\beta^T\right)$$

□

5.2.2 Key differences from the phenix model

The relation to the phenix model is straightforward: both modify the PMF model with a new prior on S, with phenix using a kinship-only prior and CMMs a GLMM. Roughly, phenix requires that latent traits are 100% heritable, while CMMs learn these heritabilities. This generality is necessary for accurate heritability estimation, as phenix does not distinguish between low-rank and heritable contributions–both are modeled (exclusively) by S. A low-rank environmental confounder would both falsely inflate heritability estimates and be poorly modeled by the kinship-only prior in phenix.

A second, subtler difference from phenix is that ϵ now has independent columns. Though I extensively argue that modeling environmental correlations is key, CMMs–unlike phenix–incorporate these correlations in the low-rank component, eliminating the need for correlated noise. ϵ plays fundamentally different roles in phenix

and CMMs: in phenix, ϵ captures all environmental contributions; in CMMs, ϵ represents low-rank reconstruction error. The latter is far more reasonably treated as trait-wise independent, which in turn eliminates the $O(P^3)$ steps that limit phenix to a moderate number of traits.

5.2.3 Key differences from GLMMs

CMMs explicitly use GLMMs to encode between-sample correlations. The marginal models on a single trait are quite similar, the former reducing to a Bayesian LMM and the latter to a standard LMM. They both aim to partition heritability amongst variance components, too, and these estimates are commensurate. Moreover, the same computational engine (the GLMM decomposition) is used for both models, and thus both allow the same incidence matrices (Assumption 1).

The difference is in the combination of across-row and across-column correlations: in CMMs, a latent factor model is used, while GLMMs use full-rank matrix normals. Neither is expected to be uniformly superior, though dimensionality reduction is generally preferred as P grows.

A more interesting parallel emerges with low-rank GLMM VCs, e.g. by employing explicit low-dimensional parameterizations [Stegle et al., 2011; Rakitsch et al., 2013; Lippert et al., 2014; Casale et al., 2015] or nuclear norm/rank constraints. In this case, both GLMMs and CMMs produce low-rank representations of phenotypes.

The relationships between these representations are not obvious, but one property is clear: CMMs partition the heritability of latent factors, while low-rank GLMMs compress individual VCs, binding each low-rank factor to a single random effect. Again, which model is superior surely depends on the dataset. But

if these latent factors are meant to represent biologically meaningful endophenotypes, it seems CMMs are superior. First, endophenotypes ideally represent real biological processes and, thus, will likely have both genetic and environmental contributions. Moreover, the question of an endophenotype's heritability can be of primary interest, and this cannot be addressed by low-rank GLMMs that restrict each latent factor to a single variance component.

5.2.4 Key differences from other compressed mixed models

I call the above model a compressed mixed model because it performs a GLMM on a low-dimensional representation of the phenotype matrix. I mean compression only in the sense of compressed sensing, i.e. low-complexity computations replacing higher complexity computations.

This differs significantly from another popular notion of compression in mixed models: low-rank GRMs. These ideas extend back to the animal breeding literature (e.g. [Henderson, 1975]), where this compression was lossless–because pedigree-based expected kinship was used–and substantial–because the pedigrees were highly structured. Starting (at least) from [Zhang et al., 2010], this idea has been extended to approximate low-rank representations of the GRM. This approximation obtains the same computational advantages as exact compression but results are no longer unaffected; depending on dataset and implementation, GRM compression may improve or harm results [Zhou and Stephens, 2012].

One compression approach computes the GRM from a few SNPs dispersed throughout the genome [Lippert et al., 2011]. This compression directly uses the nature of genetic data, hoping LD mitigates information loss; equivalently, however, this approach is just a context-specific incantation of the Nystrom method.

This approximation was shown to attenuate power in highly structured datasets [Zhou and Stephens, 2012].

This approach can be modified to select SNPs based on marginal association [Listgarten et al., 2012], motivated by the interpretation that K represents causal, polygenic (yet sparse) signal; this approximates the goal of [Golan and Rosset, 2011], which aims to refine the GRM approach in [Yang et al., 2010a] to only causal SNPs (though ungenotyped causal variation is, to my knowledge, always ignored). This approach controls well for certain types of confounders including, surprisingly, a simulated dataset designed to demonstrate the impossibility of controlling for fine-scale population structure [Mathieson and McVean, 2012; Listgarten, Lippert, and Heckerman, 2013]. Nonetheless, confounding remains a phenomenological inevitably and the question is which correction methods apply to which confounding scenarios [Mathieson and McVean, 2013; Widmer et al., 2014]. More generally, it seems multiple notions of genetic relationship can be useful [Zaitlen et al., 2013; Tucker, Price, and Berger, 2014; Tucker et al., 2015], and it is unlikely a consensus GRM definition–much less an optimal compression–will ever emerge.

Overall, these approaches exploit low-dimensional representations of genetic relatedness, either using naturally low-rank GRMs, truncating eigenvalues [Kumar et al., 2016a], or a modified Nystrom approach. These all compress genetic relatedness; CMMs, however, compress phenotypic relatedness. Again, which compression is more useful depends on context, though (nominally) unrelated humans have very low inter-sample relatedness and high-dimensional traits typically have large correlations. Moreover, CMMs (and GLMMs) can trivially incorporate the compressed GRMs by presenting it as a tall-and-skinny G matrix; further, the

computational advantages of GRM compression are automatically obtained by the underlying GLMM decomposition.

A second difference between CMMs and past notions of compression is procedural. There are a litany of means to construct a low-rank GRM, and even within-package the optimal method depends on dataset [Widmer et al., 2014]. CMMs, however, use rank-regularization motivated by generic notions of compression rather than biologically-informed choices of SNP sets. In summary, CMMs encapsulate GRM compression and are simpler and more generic.

5.2.5 Parameter choices

M, the *a priori* rank of S (or number of latent phenotypes), is a crucial parameter for CMMs. For phenix, I simply set $M = \min(N, P)$, which is possible only for small P. CMMs, however, aim at larger P, and both runtime and over-fitting become prohibitive if $M = \min(N, P)$.

I use two strategies to choose M. First, I simply fit CMMs with $M = 100, 200, \ldots$, until $M \geq P$ or M becomes prohibitively large. The logic of this approach is twofold: first, automatic rank determination makes CMMs relatively insensitive to choice of M, motivating the large spacing between values of M; second, the large spacing attenuates problems of over-fitting and minimizes computational cost.

Second, I choose M by greedy forward selection until some cross-validated error increases (in Section 5.4, I use prediction error): starting from $M = 10$, I increment M by 10 until cross-validated performance decays. In Section 5.4, this approach chooses $M \approx 70, 50$, and 40, respectively, for the rat ($P = 140$), yeast ($P = 46$) and HCP ($P = 298$) datasets which, qualitatively, seems too low.

Despite the flexibility of this approach, it consistently under-performs to simply setting $M = 100$; this is because the greedy algorithm can easily terminate too early, the $M = 100$ approach adapts to lower-rank surprisingly adroitly, and the datasets are not too large.

By default, I use non-informative $a = 1$, $b = 1e8$. This is essentially a point mass at 0, as in phenix. An exact point mass can be used, but this leads to computational instabilities by creating exactly-zero components. In theory, these components can simply be dropped (which even expedites computation); this is redolent of Gaussian mixture models, where components can converge to an infinite-likelihood point mass. As for GMMs, regularization is a viable means to avoid this singularity.

Similarly, model performance in Section 4.4.2 depends very little on (reasonable) choices for c and d, but non-informative priors allow singular solutions. I simply set $c = 10$ and $d = 10$, providing a modicum of regularization while remaining relatively non-informative.

I terminate the algorithm when the relative change in the marginal likelihood (lower bound) is less than 10^{-4} or after 1,000 VB iterations; results in Section 4.4.2 were insensitive to reasonable variations on these choices.

5.3 Fitting CMMs with Variational Bayes

I fit CMMs with variational Bayes under a mean-field approximation that S, β, τ, ν and Y_{miss} are independent in the posterior; the VCs C are optimized with empirical Bayes [Nakajima and Sugiyama, 2011] (see Section 5.3.3).

5.3.1 The parametric forms of the approximate posterior marginals

This section takes the same approach as Section 3.3.2, iteratively updating each variational parameter block with equation (A.12).

$$Y : Q_{Y_{np}} \overset{\text{ind}}{\sim} \mathcal{N}\left(\mu^Y_{np}, \nu^Y_p\right)$$

$$\begin{aligned} -2\log Q_{Y^m} &\equiv -2\mathbb{E}_{-Y^m}\left(\log P(Y|S,\beta,\nu)\right) \\ &\equiv \mathbb{E}_{-Y^m}\left(||(Y - S\beta)\nu^{1/2}||^2_F\right) \\ &\equiv ||(Y - \mathbb{E}\left(S\right)\mathbb{E}\left(\beta\right))\mathbb{E}\left(\nu^{1/2}\right)||^2_F \implies \\ Q_{Y_{np}} &\overset{\text{ind}}{\sim} \mathcal{N}\left(\mu^Y_{np}, \nu^Y_p\right) \text{ for } (n,p) \in miss \end{aligned}$$

The sufficient statistics μ_Y and ν_Y are defined by the other approximate marginals:

$$\mu^Y_{miss} = (\mu_S \mu_\beta)_{miss} \tag{5.2}$$

$$\nu^Y_p = \frac{c'}{d'_p} \tag{5.3}$$

Copying ν as ν^Y is analogous to copying Ω as Ω_β in phenix: at convergence and at many steps in the algorithm, $\nu_Y = \nu_\beta = \frac{c'}{d'_p} = \mathbb{E}_Q\left(\nu\right)$. This minor annoyance ensures the marginal likelihood is always non-decreasing.

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

This step and the gradient step for C are the only complicated parts of the algorithm, both algebraically and computationally (typically, these steps comprise $\sim 90\%$ of overall runtime).

Let $\Lambda_0(C) = \left(\sum_{m\geq 0} C_m \otimes K_m\right)^{-1}$ be the *a priori* GLMM precision of s given C; except where multiple versions of C are involved, I simply write Λ_0. The update

for S is

$$\begin{aligned}
-2\log Q_{-S} &\equiv -2\mathbb{E}_{-S}\left(\log P(Y|S,\beta,\nu) + \log P(S|C)\right) \\
&\equiv \mathbb{E}_{-S}\left(||(Y - S\beta)\nu^{1/2}||_F^2 + ||\Lambda_0^{1/2}s||_2^2\right) \\
&\equiv -2\mathrm{tr}\left(S\mathbb{E}\left(\beta\nu Y^T\right)\right) + s^T\left(\mathbb{E}\left(\beta\nu\beta^T\right)\otimes I + \Lambda_0\right)s \implies \\
s &\sim \mathcal{N}\left(\mu_s, \Lambda_s^{-1}\right)
\end{aligned}$$

where

$$\Lambda_s = V_\beta \otimes I + \Lambda_0 \qquad \text{(implicit)}$$

$$V_\beta = \mu_\beta \frac{c'}{d'}\mu_\beta^T + \mathrm{tr}_M\left(\left(\frac{c'}{d'}\otimes I\right)\Lambda_b^{-1}\right) \qquad (5.4)$$

$$\mu_s = \Lambda_s^{-1}\mathrm{vec}\left(\mu_Y\frac{c'}{d'}\mu_\beta^T\right) \qquad (5.5)$$

(5.4) can be evaluated in $O(M^3 + MP)$ using the definition of Λ_b below and

$$\mathrm{tr}_M\left(\left(\frac{c'}{d'}\otimes I\right)\Lambda_b^{-1}\right) = Q_{V_S}\mathrm{tr}_M\left(\left(\frac{c'}{d'}\otimes I\right)\left[\tau_\beta\otimes I + \nu_\beta\otimes\Lambda_{V_S}\right]^{-1}\right)Q_{V_S}^T$$

Λ_s, unfortunately, is not a GLMM precision. Fortunately, though, it can be rewritten in terms of a GLMM precision: write $B = V_\beta^{-1}\otimes I_N$, so that

$$\Lambda_s^{-1} = \left[B^{-1} + \Lambda_0\right]^{-1} = B - B\left(B + \Lambda_0^{-1}\right)^{-1}B$$

using Woodbury. Defining $C_0' := C_0 + V_\beta^{-1}$, $C_m' := C_m$ for $m > 0$, and $\Lambda_* = \Lambda_0(C')$,

$$\Lambda_s^{-1} = B - B\Lambda_* B \qquad (5.6)$$

It is natural that C_0' combines C_0, which captures the environmental contribution to the latent traits, and the likelihood part of the precision (B) that reflects errors in the low-rank representation of Y, which are assumed entry-wise independent.

Given the representation in equation (5.6), μ_s can be evaluated using Proposition 2

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

Let $b := \text{vec}\,(\beta)$, so that

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

giving the updates

$$\Lambda_b = \nu_\beta \otimes V_S + \tau_\beta \otimes I_M \qquad \text{(implicit)}$$

$$\mu_b = \Lambda_b^{-1}\text{vec}\left(\mu_S^T\mu_Y\nu_\beta\right) = Q_S\left[\left(Q_S^T\mu_S^T\mu_Y\right) / \left(\lambda_{V_S}1_P^T + 1_M(\tau_\beta/\nu_\beta)^T\right)\right] \qquad (5.7)$$

$$V_S = \mu_S^T\mu_S + W_S \qquad (5.8)$$

$$W_S = \text{tr}_M\left(\Lambda_s^{-1}\right) = NV_\beta^{-1} - V_\beta^{-1}\text{tr}_M\left(\Lambda_*\right)V_\beta^{-1} \qquad (5.9)$$

$$\tau_p^\beta = \frac{a'}{b'_p}; \nu_p^\beta = \frac{c'}{d'_p} \qquad (5.10)$$

The divisions in (5.7) are element-wise. Together, (5.7) and (5.8) cost $O(MNP)$ as written. (5.9) uses the GLMM structure of Λ_s given in (5.6) and can be cheaply evaluated using the partial trace computations from Section 4.3.1.

$\tau : Q_{\tau_p} \overset{\text{ind}}{\sim} \mathbf{Gamma}(a', b'_p)$

$$\begin{aligned}
-2\log Q_\tau &\equiv -2\mathbb{E}_{-\tau}\left(\log P(\beta|\tau) + \log P(\tau)\right) \\
&\equiv \left[-M\sum_p \log\tau_p + \text{tr}\left(\tau\mathbb{E}\left(\beta^T\beta\right)\right)\right] + 2\left[-(a-1)\sum_p \log\tau_p + b\sum_p \tau_p\right] \\
&\equiv -(2a + M - 2)\sum_p \log\tau_p + \sum_p (2b + \mathbb{E}\left(||\beta_p||_2^2\right))\tau_p \implies \\
Q_{\tau_p} &\overset{\text{ind}}{\sim} \text{Gamma}(a', b'_p)
\end{aligned}$$

where

$$\begin{aligned} a' &= a + \frac{M}{2} && (5.11) \\ b'_p &= b + \frac{1}{2}\left(\mathbb{E}\left(\|\beta_{,p}\|_2^2\right)\right) \\ &= b + \frac{1}{2}\left(\|(\mu_\beta)_{,p}\|_2^2 + \mathrm{tr}\left((\Lambda_b)^{-1}_{(pp)}\right)\right) && (5.12) \end{aligned}$$

These updates jointly cost only $O(MP)$ using

$$\mathrm{tr}\left((\Lambda_b)^{-1}_{(pp)}\right) = \mathrm{tr}\left(\left(\nu_p^\beta V_S + \tau_P^\beta I\right)^{-1}\right) = \sum_m \frac{1}{\nu_i^\beta \lambda_m^{V_S} + \tau_i^\beta}$$

$\nu : Q_{\nu_p} \overset{\text{ind}}{\sim} \mathbf{Gamma}(c', d'_p)$

First, using multi-indices defined in Section 4.3.1 (i.e. $(A \otimes B)_{(r)(q)} = A_{rq}B$),

$$\begin{aligned} \mathbb{E}\left((S_{i,}\beta_{,j})^2\right) &= \mathrm{tr}\left(\mathbb{E}\left(S_i^T S_i\right)\mathbb{E}\left(\beta_j \beta_j^T\right)\right) \\ &= \mathrm{tr}\left(\left(\mu_i^{S^T}\mu_i^S + (\Lambda_s^{-1})_{(ii)}\right)\left(\mu_j^\beta \mu_j^{\beta^T} + (\Lambda_b^{-1})_{(jj)}\right)\right) \\ &= (\mu_S \mu_\beta)^2_{ij} + \mathrm{tr}\left(\left(\mu_i^{S^T}\mu_i^S + (\Lambda_s^{-1})_{(ii)}\right)(\Lambda_b^{-1})_{(jj)}\right) + \mu_j^\beta (\Lambda_s^{-1})_{(ii)} \mu_j^\beta \implies \end{aligned}$$

$$\sum_i \mathbb{E}\left((S_{i,}\beta_{,j})^2\right) = \sum_i (\mu_S \mu_\beta)^2_{ij} + \mathrm{tr}\left(\tilde{V_S}(\Lambda_b^{-1})_{(jj)}\right) + \mu_j^\beta \tilde{W_S} \mu_j^\beta$$

where $\tilde{V_S}$ and $\tilde{W_S}$ are up-to-date versions of V_S and W_S. The complicated term can be rewritten

$$\begin{aligned} x_j &:= \mathrm{tr}\left(\tilde{V_S}(\Lambda_b^{-1})_{(jj)}\right) \\ &= \mathrm{tr}\left(\tilde{V_S}(\nu_j^\beta V_S + \tau_j^\beta I)^{-1}\right) \\ &= \mathrm{tr}\left((Q_{V_S}^T \tilde{V_S} Q_{V_S})(\nu_j^\beta \Lambda_{V_S} + \tau_j^\beta I)^{-1}\right) \end{aligned}$$

After evaluating $x' = \mathrm{diag}\left(Q_{V_S}^T \tilde{V_S} Q_{V_S}\right)$ in $O(M^2)$ (or if β, and thus W_S and V_S, has been more recently updated than S, $x' = \lambda^{V_S}$ is free),

$$x_j = \sum_m \frac{x'_m}{\lambda_m^{V_S} \nu_j^\beta + \tau_j^\beta}$$

Now, the update for ν can be written in a simple form:

$$
\begin{aligned}
2\log Q_\epsilon(\nu) &\equiv 2\mathbb{E}_{-\nu}\left(\log P(Y|S,\beta,\nu) + \log P(\nu)\right) \\
&\equiv \mathbb{E}_{-\nu}\left(\left(N\sum_j \log\nu_i - \sum_j \nu_j||(Y-S\beta)_{,j}||_2^2\right) + \left(2(c-1)\sum_j \log\nu_j - 2d\sum_j \nu_j\right)\right) \\
&\equiv \sum_j \left((N+2c-2)\log\nu_i - \sum_j \nu_j\left(\mathbb{E}\left(||(Y-S\beta)_{,j}||_2^2\right) + 2d\right)\right) \implies \\
Q_{\nu_j} &\overset{\text{ind}}{\sim} \text{Gamma}(c', d'_j)
\end{aligned}
$$

where

$$
\begin{aligned}
c' &:= c + \frac{N}{2} \qquad (5.13) \\
d'_j &:= d + \frac{1}{2}\left(\mathbb{E}\left(||(Y-S\beta)_{,j}||_2^2\right)\right) \\
&= d + \frac{1}{2}\left(\sum_i \left[\mathbb{E}\left(Y_{ij}^2\right) - 2\mathbb{E}\left(Y_{ij}S_{i,}\beta_{,j}\right) + \mathbb{E}\left((S_{i,}\beta_{,j})^2\right)\right]\right) \\
&= d + \frac{1}{2}\left(\sum_i \left[\left(\mu_{ij}^{Y\,2} + I\{j\in m_i\}(\nu_j^Y)^{-1}\right) - 2\mu_{ij}^Y(\mu_S\mu_\beta)_{ij} + (\mu_S\mu_\beta)_{ij}^2\right] + x_j + \mu_j^\beta W_S^0 \mu_j^\beta\right) \\
&= d + \frac{1}{2}\left(||(\mu^Y - \mu_S\mu_\beta)_{,j}||_2^2 + |m_j|(\nu_j^Y)^{-1} + x_j + ||(W_S^0)^{1/2}\mu_j^\beta||_2^2\right) \implies \\
d' &= d + \frac{1}{2}\left(\text{colSums}((\mu_Y - \mu_S\mu_\beta)^2) + |m| * \nu_Y^{-1} + x + \text{colSums}(((W_S^0)^{1/2}\mu_\beta)^2)\right) \qquad (5.14)
\end{aligned}
$$

using colSums to indicate column-wise sums and applying the squares inside this operator element-wise. This step costs $O(NMP)$ in total.

5.3.2 The marginal likelihood lower bound

I compute the marginal likelihood lower bound for a variational approximation Q as for phenix:

$$\begin{aligned}
F(Q) &= \mathbb{E}_{\theta\sim Q}\left(\log P(Y^o,\theta) - \log Q(\theta)\right) \\
&= \mathbb{E}_Q\left(\log P(Y^o, Y^m, \beta, S, \nu, \tau, C) - \log Q(Y^m, \beta, S, \nu, \tau, C)\right) \\
&= \mathbb{E}_Q\left(\log P(Y|\beta, S, \nu) + \log P(\nu) - \log Q_Y(Y^m) - \log Q_\nu(\nu)\right) \quad (5.15) \\
&\quad + \mathbb{E}_Q\left(\log P(\beta|\tau) + \log P(\tau) - \log Q_\beta(\beta) - \log Q_\tau(\tau)\right) \quad (5.16) \\
&\quad + \mathbb{E}_Q\left(\log P(S|C) - \log Q_S(S)\right) \quad (5.17)
\end{aligned}$$

I now evaluate each component of the sum:

$$\begin{aligned}
(5.15) &\equiv 2\mathbb{E}_Q\left(\log P(Y|\beta, S, \nu) + \log P(\nu) - \log Q_{Y,\nu}(Y^m, \nu)\right) \\
&\equiv \sum_p \mathbb{E}_Q\Big(N\log\nu_p - \nu_p\|Y_p - S\beta_p\|^2 + 2(c-1)\log\nu_p - 2d\nu_p \\
&\qquad - \sum_p \left(|m_p|\log\nu_p^Y - \nu_p^Y\|Y_{m_p,p} - \mu_p^Y\|^2 + 2c'\log d'_p + 2(c'-1)\log\nu_p - 2d'_p\nu_p\right)\Big) \\
&\equiv \sum_p \mathbb{E}\left(\nu_p\left(2d'_p - 2d - \left|\left|Y_p - S\beta_p\right|\right|^2\right)\right) - \sum_p |m_p|\log\nu_p^Y - 2c'\sum_p \log d'_p \\
&\equiv \sum_p \frac{2c'}{d'_p}\left(d'_p - \tilde{d}'_p\right) - \sum_p |m_p|\log\nu_p^Y - 2c'\sum_p \log d'_p
\end{aligned}$$

where $\tilde{d}'_p$ is a version of d'_p computed at the current variational parameters; if ν was updated last, $\tilde{d}'_p = d'_p$, and (5.15) simplifies.

Next, for β and its hyper-parameter τ,

$$
\begin{aligned}
(5.16) &\equiv 2\mathbb{E}_Q\left(\log P(\beta|\tau) + \log P(\tau) - \log Q_\beta(\beta) - \log Q_\tau(\tau)\right) \\
&\equiv \mathbb{E}_Q\Big(M\sum_p \log\tau_p - \sum_p \tau_p||\beta_{,p}||^2 + 2(a-1)\sum_p \log\tau_p - 2b\sum_p \tau_p \\
&\qquad - \left[\log|\Lambda_b| - ||b-\mu_b||^2_{\Lambda_b} + 2a'\sum_p \log b'_p + 2(a'-1)\sum_p \log\tau_p - 2\sum_p b'_p\tau_p\right]\Big) \\
&\equiv \sum_p \mathbb{E}\left(\tau_p(2b'_p - 2b - ||\beta_{,p}||^2)\right) - \log|\Lambda_b| - 2a'\sum_p \log b'_p \\
&\equiv \sum_p \frac{2a'}{b'_p}(b'_p - \tilde{b'}_p) - \log|\Lambda_b| - 2a'\sum_p \log b'_p
\end{aligned}
$$

As with $\tilde{d'_p}$, $\tilde{b'_p}$ is a version of b'_p computed at the current variational approximation to the posterior; if τ was updated more recently than β, then $\tilde{b'_p} = b'_p$ and the expression simplifies. $|\Lambda_b|$ can be cheaply computed from its eigenvalues, which can be read off from the definition of Λ_b.

Finally, the S part is

$$
\begin{aligned}
(5.17) &= 2\mathbb{E}_Q\left(\log P(S|C) - \log Q_S(S)\right) \\
&= \mathbb{E}_Q\left(\log|\Lambda_0(C)| - ||s||^2_{\Lambda_0(C)} - \log|\Lambda_s| + ||s-\mu_s||^2_{\Lambda_s}\right) \\
&\equiv \log|\Lambda_0\Lambda_s^{-1}| - \|\mu_s\|_{\Lambda_0} - \mathrm{tr}\left(\Lambda_0\Lambda_s^{-1}\right)
\end{aligned}
$$

$\|\mu_s\|_{\Lambda_0}$ can be computed quickly using Proposition 2 .

Computing the Λ_s terms is the principal difficulty. $|\Lambda_0\Lambda_s^{-1}|$ and $\mathrm{tr}\left(\Lambda_0\Lambda_s^{-1}\right)$ can be simplified if S was updated more recently than C. Otherwise, Λ_0 will have moved from its value used to construct Λ_s, and no obvious relationships will remain. This can be accomplished if the likelihood is never evaluated immediately after updating C, always waiting for S to additionally be updated. While not problematic for monitoring algorithm convergence, it complicates otherwise useful approaches to update C–see Section 5.3.3 for details.

Under this assumption,

$$\Lambda_0\Lambda_s^{-1} = \Lambda_0\left(\Lambda_0 + V_\beta \otimes I_N\right)^{-1} = \left(\Lambda_0^{-1} + V_\beta^{-1} \otimes I_N\right)^{-1}\left(V_\beta^{-1} \otimes I_N\right) = \Lambda_*\left(V_\beta^{-1} \otimes I_N\right)$$

and so

$$\begin{aligned}\operatorname{tr}\left(\Lambda_0\Lambda_s^{-1}\right) &= \operatorname{tr}\left(\Lambda_*\left[V_\beta^{-1} \otimes I_N\right]\right)\\ |\Lambda_0\Lambda_s^{-1}| &= |V_\beta|^{-N}|\Lambda_*|\end{aligned}$$

Results from Section 4.3.3 can now be applied to compute both these terms in $O(Nr^2M^3)$.

5.3.3 Gradient steps for C

I learn C using an established empirical Bayes approach to maximize the (approximate) marginal likelihood [Nakajima and Sugiyama, 2011] using gradient ascent (optionally projected to $\mathbb{S}_+^M$).

As a function of C, the marginal likelihood is equivalent to

$$\log|\Sigma_0^{-1}| - \operatorname{tr}\left(\Sigma_0^{-1}(\mu_s\mu_s^T + \Lambda_s^{-1})\right) = -\log|\Sigma_0| - \operatorname{tr}\left(\Sigma_0^{-1}(\mu_s\mu_s^T + \Lambda_s^{-1})\right)$$

where $\Sigma_0 := \Lambda_0^{-1}$. The gradient with respect to some x that parameterizes Σ_0 is

$$\begin{aligned}-\operatorname{tr}\left(\Sigma_0^{-1}[\nabla_x\Sigma_0]\right) -\operatorname{tr}\left((-\Sigma_0^{-1}[\nabla_x\Sigma_0]\Sigma_0^{-1})(\mu_s\mu_s^T + \Lambda_s^{-1})\right)&\\ = \operatorname{tr}\left([\nabla_x\Sigma_0]\left[\Lambda_0(\mu_s\mu_s^T + \Lambda_s^{-1})\Lambda_0 - \Lambda_0\right]\right)&\end{aligned}$$

When $x = C_{pq}^{(m)}$, $\nabla_x\Sigma_0 = I_{pq} \otimes K_m$, and so

$$\begin{aligned}\nabla_{C^{(m)}}\mathcal{L} &= \operatorname{tr}_M\left([I \otimes K_m]\left(\Lambda_0\mu_s^T\mu_s\Lambda_0 - \Lambda_0 + \Lambda_0\Lambda_s^{-1}\Lambda_0\right)\right)\\ &= \operatorname{tr}_M\left([I \otimes K_m]\Lambda_0\mu_s^T\mu_s\Lambda_0\right) - \operatorname{tr}_M\left([I \otimes K_m]\Lambda_*\right)\end{aligned}$$

The last line assumes S has been updated more recently than C, so that $\Lambda_s = V_\beta \otimes I + \Lambda_0$, and then uses Woodbury in reverse:

$$\Lambda_* := \left(\Sigma_0 + V_\beta^{-1} \otimes I\right)^{-1} = \Sigma_0^{-1} - \Sigma_0^{-1}\left(\Sigma_0^{-1} + V_\beta \otimes I\right)^{-1}\Sigma_0^{-1} = \Lambda_0 - \Lambda_0\Lambda_s^{-1}\Lambda_0$$

This can be computed with results from Section 4.

This gradient is similar in form to the standard GLMM gradient in 4.3.2, with μ_S replacing Y and, in the second term, Λ_* replacing Λ_0. Λ_* replacing Λ_0 serves to propagate uncertainty in S: as $\Lambda_0 \succeq \Lambda_*$, Λ_* yields weaker regularization of C toward 0 or, equivalently, larger C that reflect both variance across S and uncertainty in S.

Step size

At each VB iteration, I take a gradient step for C with step size η. I implement two strategies to choose η. The first simply sets $\eta = \frac{\eta_0}{t}$ at iteration t, where $\eta_0 = .01$ was hand-tuned in simulations; the prediction results in Section 4.4.2 varied only slightly when using either $\eta_0 = .1$ or $\eta_0 = .001$ (not shown). This approach works well for tasks that do not require precise identification of latent factors, like prediction; however, it regularly results in unreasonable estimates of C, and so I follow this approach by fitting a GLMM to the resulting μ_S matrix. In fact, setting C to some fixed large multiple of the identity (I used $10^4\ I_M$) gave very similar prediction results (not shown). Overall, when C is not of direct interest, every reasonable approach to choose C (i.e. excluding small C) appears to work relatively well.

I also implement line search to choose η. Specifically, I use standard backtracking line search with Armijo's rule to choose η, setting the attenuation rate $\beta = .7$

and objective improvement criterion $\alpha = .1$; again, results were insensitive to reasonable choices for these parameters. This approach gives much better estimates of the C matrices. One difficulty that arises is simply evaluating the objective function–as mentioned above, the likelihood is expensive if C has been updated more recently than S. To circumvent this, I simply update S prior to evaluating the likelihood, throwing away this S update if the tested value for η is rejected. While this invalidates some guarantees of backtracking line search, it retains all guarantees of the VB algorithm, as no step is ever performed that decreases the marginal likelihood lower bound.

Asymptotically, the two approaches give the same result modulo local modes, but the gradient approach can easily escape the well-behaved region of the parameter space and, more importantly, avoiding this requires small τ_0 which, in turn, means the C matrices take dramatically longer to converge than the other model parameters. If full convergence of the marginal likelihood is required, then, line search is recommended; if only the product $S\beta$ is relevant and runtime is crucial, then fixed step sizes (or, more simply, fixed C) eliminate unnecessary and expensive likelihood evaluations. In the prediction analyses, I use both approaches, and find that line search is more expensive but typically performs slightly better. In the GWAS, I use only the line search approach, as precise identification of the factors is key.

5.3.4 Computing out-of-sample predictions

This section predicts entirely-blank rows of Y, as in 4.4.2; one difference, though, is that GLMMs require pre-imputation with phenix. Prediction also arises internally when fitting CMMs: no information is added by including unphenotyped sam-

ples, and thus excluding them cannot hurt; conversely, the approximate Bayesian approach does not guarantee including vacuous information is innocuous, and I consistently observe that entirely blank rows of Y encourage underfitting. So I hold out blank samples when fitting CMMs and predict them *post hoc*.

I predict unphenotyped samples to their conditional expectations under model (5.1) given the variational posterior Q:

$$\mathbb{E}\left(Y_{-O,}|Q\right) = \mathbb{E}\left(S_{-O,}|Q\right)\mathbb{E}\left(\beta|Q\right) + \mathbb{E}\left(\epsilon_{-O,}|Q\right) = \mathbb{E}\left(S_{-O,}|Q\right)\mu_\beta$$

The out-of-sample scores can evaluated using iterated expectations:

$$\mathbb{E}\left(S_{-O,}|Q\right) = \mathbb{E}\left(\mathbb{E}\left(S_{-O,}|S_{O,}\right)|Q\right) = \mathbb{E}\left(\Sigma_{(MO)}(\Sigma_{(OO)})^{-1}\text{vec}\left(S_{O,}\right)|Q\right)$$

where $\Sigma = \mathbb{V}\left(s\right) = \sum_{m=0}^{k} C_m \otimes K_m$ is the variance of S given C. This uses the conditional independence of $S_{-O,}$ and other parameters given $S_{O,}$.

Evaluating the next expectation gives

$$\mathbb{E}\left(S_{-O,}|Q\right) = \Sigma_{(MO)}(\Sigma_{(OO)})^{-1}\text{vec}\left(\mu^S_{O,}\right)$$

because the C are deterministic. Random C would typically create unmanageable expected Σ submatrices that would not (obviously) inherit the necessary GLMM parsimony. With fixed C, though, the prediction tools from Section 4.3.4 can be applied without modification.

In summary, CMMs predict scores via standard BLUP and then combine them with β to predict phenotypes.

Miscalibration in the scale of the C matrices

[Nakajima and Sugiyama, 2011], reviewed in Section 3, suggests that the scale of the estimated C matrices is not reliable. This means downstream inference should

not depend on the scale of C. Fortunately, this automatically holds for most analyses, which naturally depend on (co)heritabilities, not covariances. Indeed, typical mixed models eliminate this scale factor first, either by scaling traits to unit variance or, equivalently for MLEs, profiling out overall trait variance [Lippert et al., 2011; Zhou and Stephens, 2012].

Prediction, for example, is clearly invariant under rescaling by some $\lambda \in \mathbb{R}$:

$$\begin{aligned}
\Sigma_{(MO)}(\Sigma_{(OO)})^{-1} &= \left(\sum_{m=0}^{M} C_m \otimes K_m\right)_{(MO)} \left(\sum_{m=0}^{M} C_m \otimes K_m\right)^{-1}_{(OO)} \\
&= \left(\sum_{m=0}^{M} C_m \otimes K_m\right)_{(MO)} (\lambda\lambda^{-1}) \left(\sum_{m=0}^{M} C_m \otimes K_m\right)^{-1}_{(OO)} \\
&= \left(\sum_{m=0}^{M} (\lambda C_m) \otimes K_m\right)_{(MO)} \left(\sum_{m=0}^{M} (\lambda C_m) \otimes K_m\right)^{-1}_{(OO)}
\end{aligned}$$

This scale-invariance may also partially explain the disappointing performance of GLMM penalization in Section 4.4.2, as the simultaneous shrinkage of both genetic and environmental VCs in some sense cancels out. This is certainly not the whole story, though, as different penalty magnitudes were allowed (and GRM-only penalization was investigated) and ℓ_2 regularization is not simply a rescaling operation. Nonetheless, this suggests standard shrinkage notions might not be the appropriate way to regularize for prediction out-of-sample; perhaps dimensionality reduction, as in CMMs, or penalization of some VC transformation, like in [Meyer and Kirkpatrick, 2010], is preferable. Certainly, the implicit barrier penalty that recovers LMMs from GLMMs performs well, obtaining benefits of parsimony while naturally avoiding this pitfall of moot-VC-rescaling.

5.3.5 Computational complexity

The updates for Y cost $O(|miss|M) < O(NMP)$; for τ cost $O(MP)$; and for ν step cost $O(NMP)$. These steps, along with the cheap parts of the S and β steps, are collectively negligible at a cost of $O(MNP)$.

This leaves only the Λ_s-dependent terms in (5.5) and (5.9). Both depend on the GLMM decomposition of Λ_* (performed as part of the update for S), which costs $O(Nr^2M^3)$ by Proposition 1. (5.5) then involves multiplying a vector by $V_\beta^{-1} \otimes I_N$, $O(MNP)$ matrix multiplications and, primarily, multiplying a vector by Λ_*, costing $O(NrM^2)$ by Proposition 2. (5.9) involves trivial $O(M^3)$ multiplications and the partial trace of a GLMM precision, costing $O(rNM)$ by computations in Section 4.3.1.

All together, a VB iteration costs $O(Nr^2M^3 + N^2M + NPM)$ (which simplifies if K_1 or $\{G_m\}_{m>1}$ are omitted).

Unfortunately, equation (5.5) induces $O(N^2M)$ iterations, unlike standard mixed models with only one $O(N^2)$ operation. As in phenix, the complexity can be (nominally) improved by storing a whitened, padded Y matrix. However, this is even less valuable here, as CMMs apply to much larger P, no-larger N and $M < P$:, all shift the balance of computation to operations scaling in P. Moreover, if $O(N^2M)$ computations are prohibitive, the one-off $O(N^3)$ eigendecomposition will be prohibitive, too, and the GRM must be omitted, anyway.

5.4 Real phenotype prediction with CMMs

This section replicates the analysis in Section 4.4.2: I mask entire rows of real phenotype matrices and assess prediction accuracy. Sporadically missing phenotypes

are explicitly modeled by phenix and CMMs and are handled naturally in LMMs by dropping missing samples trait-wise (without data loss). GLMMs, however, require complete training matrices, and I pre-impute the down-sampled datasets with phenix.

Accuracy is assessed as in Section 4.4.2 and, roughly, as in [Rakitsch et al., 2013; Tucker et al., 2015]: I compute prediction (signed) squared correlations display the relative accuracy of each method compared to LMM. Using MSE, instead, to measure accuracy yields qualitatively similar results (not shown).

I use the real datasets from Section 4.4.2 with a few modifications. First, I discard the UKBS as relatedness is too low–in MSE, no method significantly outperformed setting all traits to 0 (Y is normalized). Second, CMMs can fit the full HCP dataset with $P = 298$, and I no longer divide it into sub-datasets. Finally, I assess only 1% missingness, rather than varying this amount.

Table 5.1 shows the resulting prediction accuracies relative to LMM. Primarily, all methods perform similarly; especially, neither phenix, LMM or MPMM globally dominates another. CMMs, at least, always outperform LMM, albeit by slim margins. Further complicating the story, no implementation of CMMs is generally optimal, with fixed step sizes for learning C (CMM+_eta0) surprisingly outperforming line search (used in all other CMM implementations) in the NSPHS; moreover, greedy selection of M (CMM+) was always inferior to simply setting $M = 100$ (CMM_100). Nonetheless, it seems that running CMMs with line search with a few fixed M generally works well.

In contrast to method choice, additional random effects can have substantial effects. The wheat dataset includes pedigree summaries that group the 720 samples into 28 "lines." I defined an incidence matrix $G \in \mathbb{R}^{720\times 28}$ by $G_{ij} = 1$ if sample

i is in line j and 0 otherwise (and standardizing as in Section 1.4). Including this matrix improves accuracy by 50% over LMM, an order of magnitude larger difference than seen between *any* methods.

Unlike multiple-VC approaches that construct secondary relatedness matrices directly from genotypes (e.g. [Speed and Balding, 2014; Tucker et al., 2015]), including "line" is not a generally meaningful prescription. Moreover, other random effects in other datasets need not be so informative (though sex, population and common environment likely have strong effects in many human datasets). Nonetheless, it is clear that additional random effects can be dramatically more important than choice of prediction method. One implication for prediction method development, then, is that modeling multiple VCs–unlike modeling multiple traits–is crucial.

	NSPHS	Wheat	Yeast	Rats	HCP
LMM	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
PHENIX	-4.1 (1.0)	12.6 (1.8)	-0.3 (0.1)	**3.1** (0.4)	–
MPMM	-3.9 (0.8)	-7.3 (1.1)	0.6 (0.1)	0.9 (0.4)	–
CMM_100	-2.3 (0.6)	**17.1** (2.2)	**1.0** (0.6)	2.2 (0.5)	0.3 (0.9)
CMM_200	–	–	–	-0.5 (1.0)	**2.1** (1.0)
CMM+	–	–	0.8 (0.5)	-3.7 (0.4)	-11.0 (3.8)
CMM+_eta0	**1.3** (0.3)	12.8 (2.2)	-2.8 (1.3)	-2.4 (1.0)	-20.8 (2.8)
CMM_2K	–	60.3 (5.5)	–	–	–

Table 5.1: Compressive mixed models slightly improve out-of-sample prediction. Prediction is measured by signed squared-correlation with held-out phenotype data, and percent change relative to LMMs is reported in the above table. At least one CMM option always outperforms univariate LMM; MPMM, phenix and LMM all perform similarly. A substantial difference is observed when an additional random effect is included (in wheat). Table generated with `xtable` [Dahl, 2009]

	nsphs	wheat	yeast	rats	hcp	
LMM	1.8	0.2	4.5	27.1	2.7	
PHENIX	0.3	0.1	0.3	6.4	–	
MPMM	0.3	0.2	0.5	7.8	–	
CMM_100	2.0	0.8	15.6	22.0	1.5	
CMM_200	–	–	–	17.4	5.9	
CMM+	–	–	48.1	246.6	50.7	
CMM+_eta0	0.2	0.1	54.8	110.7	28.3	
CMM_2K	–	2.9	–	–	–	–

Table 5.2: Runtime in minutes. Methods were run on the same machine but with varying load on the machine, and only order-of-magnitude comparisons can be reliably inferred. Table generated with `xtable` [Dahl, 2009]

5.5 GWAS of CMM factors uncovers novel and biologically plausible loci

Testing low-dimensional summaries of phenotype matrices is not a new idea (see [Klei et al., 2008; Aschard et al., 2014] and their references to studies of phenotypic principal components). However, these approaches ignore genetic relatedness when learning the low-rank subspace, which is theoretically unsatisfying and incongruous with downstream LMM analyses. Furthermore, PCA cannot be applied to incomplete matrices. Though generalization of PCA to missing data exist [Ilin and Raiko, 2010], one such approach, softImpute, was suboptimal in the presence of heritable signal in Section 3. More practically, I have not seen such approaches applied in genetics, presumably due to their complexity.

This section closely follows the GWAS in Section 3.6.5: there, I compared the standard GWAS (baseline) to an imputed GWAS; here, I compare the baseline to a GWAS on latent CMM factors (columns of S). The goal, again, is to find novel, biologically plausible associations to demonstrate the potential utility of the new approach. The results, too, are qualitatively similar: both GWAS modifications

yield new (and non-overlapping) associations near plausible genes, and both miss many baseline associations. This suggests the new analyses complement, rather than replace, ordinary GWAS.

One difference, though, is that phenix found new associations by slightly improving signal at near-significant loci, while GWAS on CMM factors prioritizes variants that are not remotely baseline-significant (i.e. $p > 10^{-4}$). In this sense, CMM factor analysis adds truly novel information, while phenix merely boosts power. This novelty is partially why CMM GWAS can find meaningful signal despite typically attenuating power at known hits: any suboptimality in the model or implementation appears small compared to the benefit of studying latent factors.

Specifically, I use exactly the haplotype dosages, kinship matrix, and (unimputed) phenotypes from Section 3.6.5. Further, I use almost the identical GWAS pipeline: for each trait (or CMM factor), I first fit an ordinary mixed model under the genome-wide null hypothesis of no SNP association to estimate heritability; given this estimate, I then perform an F-test of the hypothesis that no haplotype dosage (of the 8 founder haplotypes) affects the phenotype for each locus (conservatively approximating the VCs under the alternative by the VCs under the null).

One difference is covariate choice. In the original analyses (both in Section 3.6.5 and [Sequencing and Consortium, 2013]), covariates were matched to traits using prior biological knowledge. CMM factors, however, aggregate over all traits, and so I conservatively regress on all covariates for all factors.

A second difference is presentational. Baseline GWAS hits are paired with a trait; CMM GWAS hits, however, correspond to a (dense) linear combination of all traits. It is thus difficult to compare power, as the two approaches perform

fundamentally different tests. However, power in each genomic region can be compared by aggregating GWAS results over traits (or factors). Though involved univariate statistic aggregation methods exist (e.g. [Sluis, Posthuma, and Dolan, 2013; Yang et al., 2010b]), I simply use Bonferroni (as only a fair comparison is desired) and summarize association evidence at a locus with the minimum p-value across traits (or factors).

The resulting $-\log_{10}$ p-values for each non-overlapping 30 SNP window (Bonferroni-adjusted for multiple traits, but not multiple SNPs) are displayed in Figure 5.1. This plot has two clear features. First, many baseline hits are completely missed by the CMM factor GWAS. Moreover, hits detected by both approaches are, almost always, more significant in the baseline. This certainly may be due to suboptimality in the CMM model or implementation or choice of CMM factors (see Section 5.2.1). Less pessimistically, it likely is also because observed traits were meaningfully chosen by knowledgeable biologists, thus injecting prior information not inherited by CMM factors. Regardless, it is clear that GWAS on CMM factors ought not replace standard GWAS, redolent of Section 3.6.5.

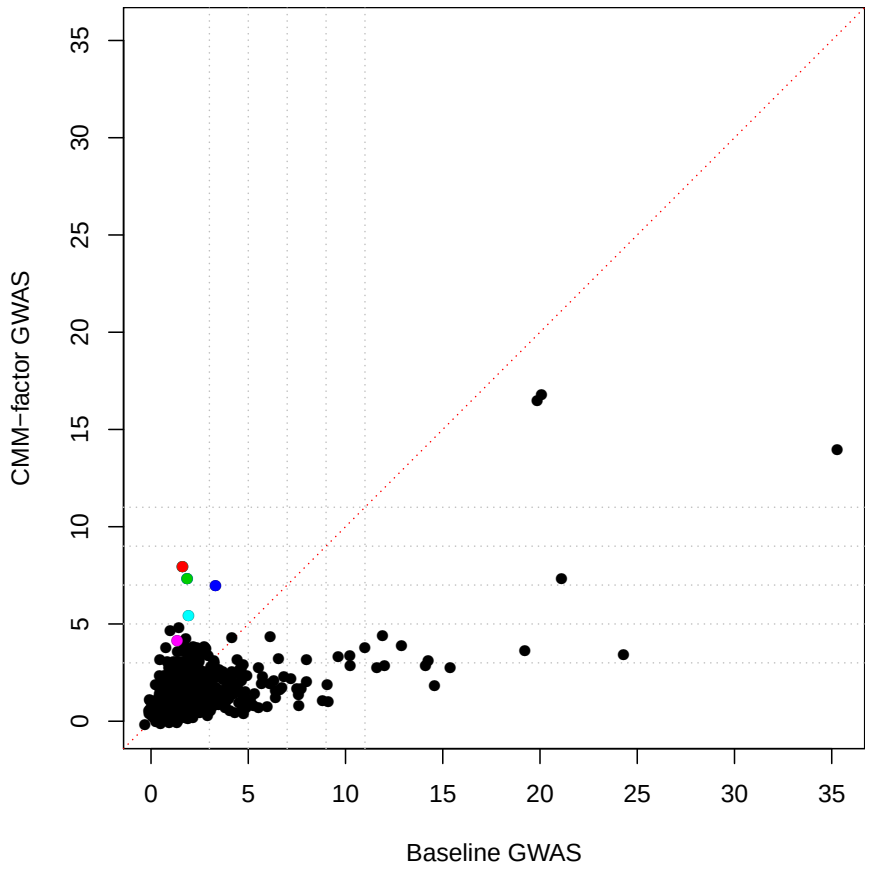


Figure 5.1: GWAS on CMM factors uncovers novel loci. $-\log_{10}$ p-values are plotted both for the baseline GWAS on observed traits (x-axis) and the GWAS on compressive mixed model factors (y-axis). Primarily, the CMM GWAS deflates signal. However, many newly significant loci emerge. The colored points correspond to interesting hits for factors representing: iron-related traits (red); lumbar and femur traits (cyan); immune traits (pink and blue); and a variety of physical size traits (green).

Fortunately, though, an entirely new set of significant loci emerge from the CMM GWAS. Interpreting associations is not as simple as in standard GWAS, as loci correspond to a combination of all traits. A sparsifying model on β would automatically identify relevant traits for each CMM factor; instead, I simply study the largest-loaded traits, implicitly sparsifying β *post hoc*.

I now look in detail at the colored points in Figure 5.1. First, the red point corresponds to Chromosome 11 and a factor consisting primarily of bone measurements, blood cell volume and iron levels. Figure 5.2 displays this hit, hoping to depict both the observed-trait contributions to the significant CMM factor and the regional $-\log_{10}$ p-values for the factor-wise and trait-wise minima and the traits contributing most to the significant CMM factor (for comparison, none are Bonferroni corrected). Contributions of observed traits to the relevant factor are quantified by PVE, visually represented by point sizes for the single-trait p-values. These single-trait p-values inform which trait contributions to the CMM factor drive its significance; note, however, that the association strength of the factor, which combines traits, is not simply the combined association strengths of the constituent traits (as in univariate-GWAS aggregating approaches).

Genes in the region are also displayed, and two are highlighted for this factor. *Tfrc* encodes the protein that modulates cellular iron levels [Andrews, 2009; Grey, 2010; Mandò et al., 2011] and has twice been reported to associate with MCV [Ganesh et al., 2009; Kamatani et al., 2010], which is closely related to the other platelet phenotypes (MPC_bc and PCDW_bc); further, iron is obviously linked to bone density phenotypes (lumbar_BMD, LW_bc, fem_neck_TRAB_DEN and fem_neck_CRT_DEN) and is known to associate with blood pressure [Kamatani et al., 2010], an otherwise surprising contributor to this factor. This analysis shows that combining biological knowledge of relevant genes and constituent traits uncovers a clear and parsimonious biological story.

Less plausible, but still relevant, *Pcyt1a* is associated with a constellation of diseases called spondylometaphyseal dysplasia, which are unified by the presentation of skeletal problems [Yamamoto et al., 2014].

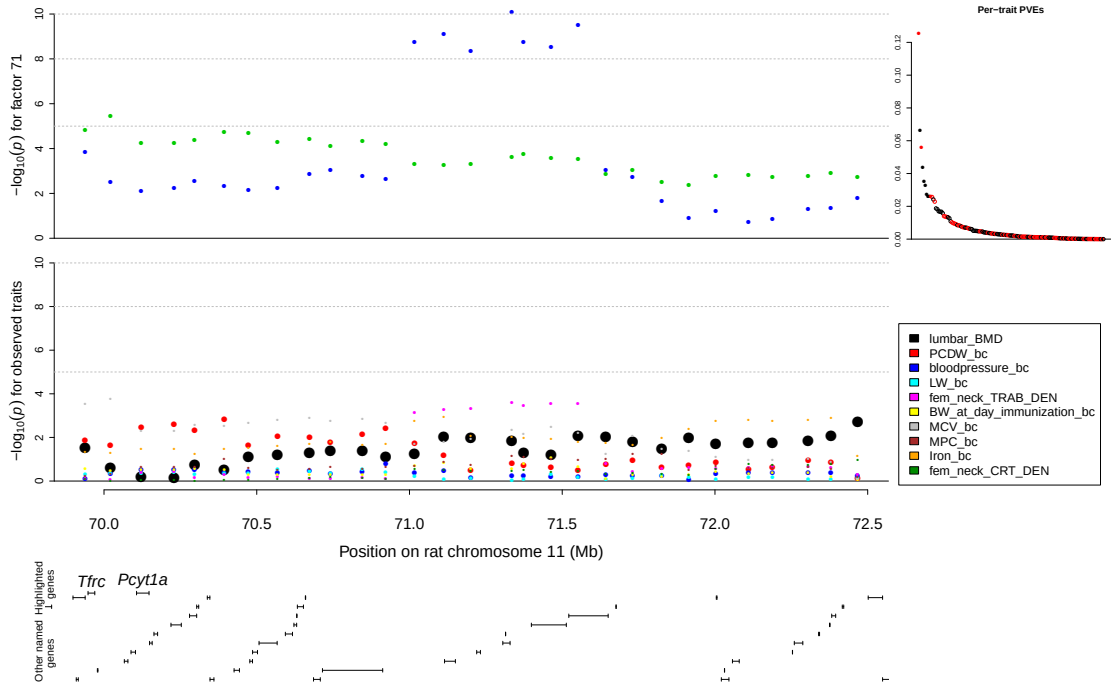


Figure 5.2: GWAS hit for bone-related CMM factor on Chromosome 11. Top: comparison of minimal p-value in ordinary GWAS (green) and CMM GWAS (blue), not Bonferroni corrected. Top right: contribution of each trait to the significant factor. Solid points represent phenotypes displayed below; red and black indicate opposite direction of the respective phenotype loading. Middle: univariate GWAS p-values for the traits contributing most to the significant CMM factor, with point sizes proportional to the size of the contribution. Bottom: genes in the region.

Next, the regional association for the cyan point in Figure 5.1 is shown in Figure 5.3. There are also two nearby genes known to associate with bone measurements, *Dock1* and *Ptpre*. Both genes are clearly involved with bone development: the former was associated with "osteoporosis-related traits" [Hsu et al., 2010], and *Ptpre* aids the bone-degrading osteoclasts [Granot-Attas, Knobler, and Elson, 2007] and more generally 'plays a major role in regulating bone structure' [Rangkasenee et al., 2013].

This locus also associates with another CMM factor (not shown). This factor

also puts large negative loadings on femur density measures and large positive loadings on lumbar area; further, both components moderately load a variety of related femur and lumbar phenotypes. These factors differ, nonetheless, by their loading magnitudes, and the second factor has a less convincing p-value (uncorrected $-\log_{10}(p)$ of 5.7 rather than 7.6). Together, these two factors suggest the novel association is robust to modest latent factor perturbations. But no trait is marginally significantly at this locus: this emphasizes the general value of factor analyses.

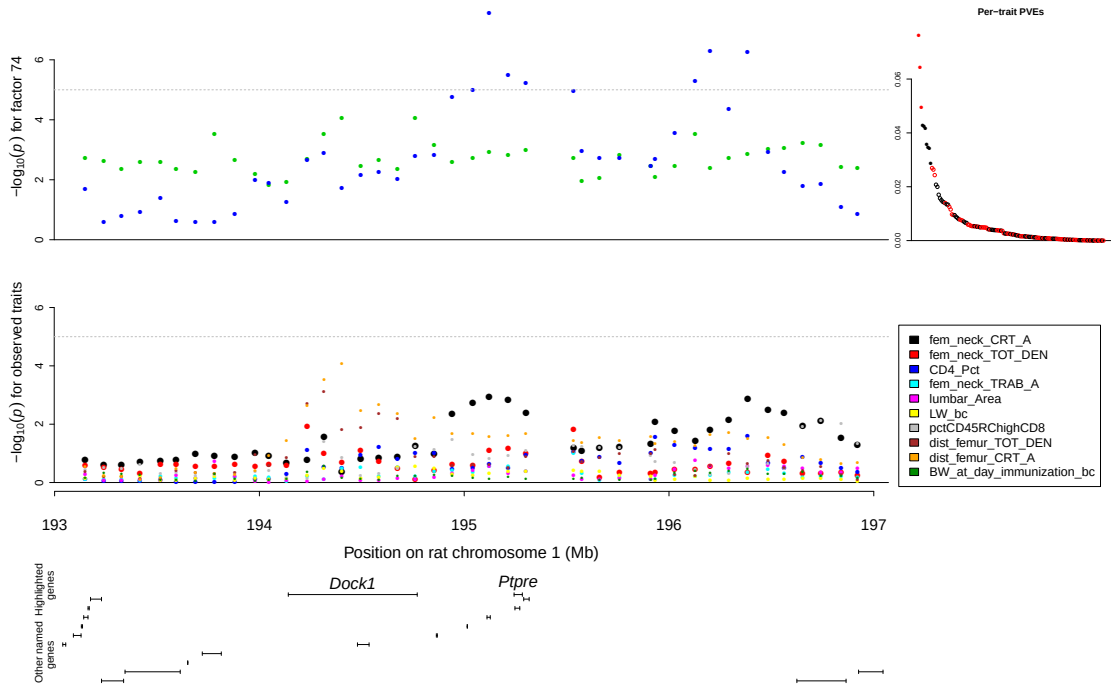


Figure 5.3: GWAS hit for bone-related CMM factor on Chromosome 1. Top: comparison of minimal p-value in ordinary GWAS (green) and CMM GWAS (blue), not Bonferroni corrected. Top right: contribution of each trait to the significant factor. Solid points represent phenotypes displayed below; red and black indicate opposite direction of the respective phenotype loading. Middle: univariate GWAS p-values for the traits contributing most to the significant CMM factor, with point sizes proportional to the size of the contribution. Bottom: genes in the region.

Finally, the pink point in Figure 5.1 is displayed in Figure 5.4. This locus contains *B2m*, an important immune gene that is strictly necessary for CD8 T cell development [Tarleton et al., 1992], matching the consistently large and negative loadings on CD8 cells in this factor (though the sign is unidentified). Further, there are even links between this primarily-CD8 protein and CD4CD25 T cells [Joetham et al., 2007], suggesting the large negative CD4CD25 loading are not red herrings. Though I found no definitive link to neutrophils, the molecule encoded by *B2m* was hypothesized to indicate neutrophil degranulation [Bjerrum, Bjerrum, and Borregaard, 1987].

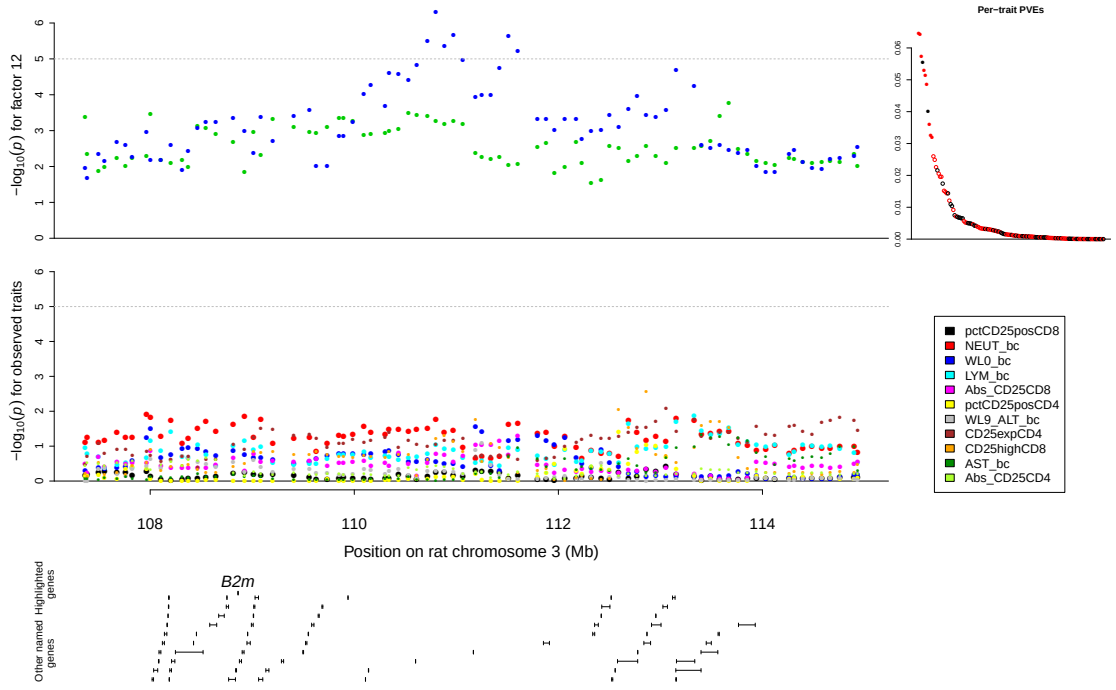


Figure 5.4: GWAS hit for immune-related CMM factor on Chromosome 3. Top: comparison of minimal p-value in ordinary GWAS (green) and CMM GWAS (blue), not Bonferroni corrected. Top right: contribution of each trait to the significant factor. Solid points represent phenotypes displayed below; red and black indicate opposite direction of the respective phenotype loading. Middle: univariate GWAS p-values for the traits contributing most to the significant CMM factor, with point sizes proportional to the size of the contribution. Bottom: genes in the region.

No gene-factor associations were immediately obvious for the other colored points in Figure 5.1. It is nonetheless possible, or likely, that a thorough investigation would reveal similar putative biology. Moreover, the factors themselves appear biologically meaningful–for example, the green point corresponds to a factor giving large positive loadings to roughly a dozen bone measurement traits, along with body and heart weight.

Additionally, the blue point corresponds to a region very near one of the most interesting hits in the original study [Sequencing and Consortium, 2013]. This

locus harbors *Tbx21*, a gene known to influence T cell development and activation, and was associated with many immune phenotypes in the original study. The factor that associates with this locus, unsurprisingly, has high loadings on many of the same T cell phenotypes. Moreover, this factor is itself interesting, as it combines phenotypes along an axis resembling T cell commitment and/or a shift from T cells to B cells: the former because of large positive loadings on CD4 and CD4-to-CD8 ratio and large negative loadings on CD8 and, more interestingly, the amount and fraction of CD4 cells expressing the activation protein CD25; the latter because large positive loadings on overall T cell amount and T-to-B cell ratio and large negative loading on overall B cell amount. Whether these pathways interact, whether one is spurious, and a generally sophisticated understanding of the role of *Tbx21* in these processes are beyond the scope of this thesis. Nonetheless, it is clear that such pleiotropic loci can be powerfully detected with CMM GWAS and, moreover, that this factor provides potentially useful immunological information.

Implementing the CMM

I fit the CMM with much more stringent criteria here than in Section 5.4, terminating only when the relative change in the marginal likelihood is below 10^{-6}; this ensures that all elements of the model converge, not just the well-identified product $S\beta$. This is partially because GWAS requires greater precision in S, but also because the analysis need only be run once (rather than hundreds of times as for the prediction tests). Finally, rather than choose M by cross-validation, which is primarily computationally valuable, I simply set $M = P$, as in phenix. I then test all $M = P$ factors (though many are essentially 0), motivated by arguments that low-order factors can capture biologically meaningful signal [Aschard et al.,

2014].

5.6 Future work

There are many possible extensions and applications of CMMs. As discussed in Section 4, all mixed model (co)heritability partitioning problems can be generalized to multiple traits and multiple variance components with GLMMs, and CMMs are no different. The main advantage of CMMs over GLMMs is computational, as they extend to much larger P. There is also an interpretational advantage, as GLMMs can output large and inscrutable $P \times P$ covariance matrices which, often, will be analyzed downstream using some dimensionality reduction, anyway.

If coheritability is not of interest, there is little information added by fitting a multitrait model, as demonstrated by the consistent difficulty in improving beyond LMM for out-of-sample prediction. However, in the presence of missing data, CMMs can even be faster than LMMs–as discussed in Section 3, LMMs require P $O(N^3)$ eigendecompositions, while phenix and CMMs require only one. In the real datasets in Section 5.4, for example, LMM and CMM (with $M = 100$ and line search for C) have similar runtime.

Employing sparsity is one obvious CMM extension. Sparse S may be meaningful for some applications, but sparse β will almost always be attractive–for example, in the GWAS in Section 5.5, even though implicit *post hoc* sparsification appeared to work well in that analysis and, generally, is standard in the interpretation of (non-sparse) principal components. The Bayesian approach employed here can easily enforce sparse β using any of a litany of well-known approaches (c.f. [Dueck, Morris, and Frey, 2005; Guan and Dy, 2009; Mairal et al., 2010]). This is particularly attractive for CMMs: the S steps are by far the most expensive,

and moderately increasing the sophistication of β steps will have negligible impact on overall runtime. Finally, component sparsity eliminates the rotation ambiguity discussed in Section 5.2.1 as sparsity is not generally preserved under rotation.

The most important future direction, I feel, would be understanding the induced norm on the product $S\beta$. Even for spherical Gaussian priors this was non-trivial (c.f. [Srebro, 2004; Salakhutdinov and Mnih, 2008]), but the link to frequentist nuclear-norm penalizing approaches has been invaluable. The problem is far more difficult for CMMs, however, as I am not aware of analogues of Theorem A.7 that apply under the GLMM prior on S.

Finally, CMMs may be useful outside genetics, much as random effect models evolved from genetics into other fields (e.g. economics, spatial statistics, and experimental design). This evolution has not been mirrored in recent developments for mixed models in genetics, which focus on expediting existing approaches using the unique structure of genetics problems. CMMs, however, extend mixed models in a more generic way, at no point specifically referring to SNPs. And although CMMs are limited by Assumption 1, this rank constraint (on all but one incidence matrix) is quite general. The greater requirement for CMM application is simultaneous interest in multiple response variables.

Beyond generalizing mixed models, CMMs also generalize dimensionality reduction to non-i.i.d. data (and to missing data, though this is not novel). How CMMs fit into the broader picture of dimensionality reduction is an interesting topic for future work, though making them competitive for large, generic datasets will at least require a more sophisticated implementation and, likely, an entirely new algorithm (the same could be said, though, of the original lasso [Tibshirani, 1996] and Dantzig selector [Candes and Tao, 2007]).

In summary, CMMs have been shown to be a useful dimensionality reduction tool for GWAS, defining axes of variation that uncover novel and biologically plausible associations in the presence of missing phenotype data. They are competitive with, and arguably improve on, established mixed models for prediction, which is closely linked to many other mixed model applications. More generally, CMMs may be a first step towards a new approach for dimensionality reduction with non-i.i.d. data. Investigating applications in other fields, improving computational efficiency and deriving a more thorough understanding of the model are all crucial and substantial remaining hurdles.

Chapter 6

Future directions for multiphenotype models

I discuss phenix, GLMM and CMM extensions in their respective chapters; here, I speculate more broadly on the future of multi-trait modeling.

Mixed model computation

One question is whether mixed models (of any flavor) can successfully model large genetic datasets (i.e. N over a million). Mixed models are slow; to my knowledge, the fastest implementation (by far) requires roughly a year to perform GWAS of 10 million SNPs on half a million samples[1]; perhaps with ever-increasing compute power (and in comparison to the cost of obtaining GWAS data) this will be feasible for increasing sample size and many traits (in parallel). But it seems a losing battle–eventually, large enough N will be problematic (though, to be fair, N is bounded above for humans).

Methods like LD score regression (LDSR), which entirely ignore raw data,

[1]This very rough number uses Figure 1 from [Loh et al., 2015a], which says fitting 480,000 people and 360,000 loci takes approximately 300 hours, and extrapolates to 10 million SNPs using their claimed linear complexity in the number of SNPs.

have shown that approximate methods applied to massive datasets can outperform maximum likelihood applied to smaller datasets. Moreover, LDSR is in its infancy, and theory is being developed to extend the reach and interpretability of such summary-based methods [Zhu and Stephens, 2016].

Randomized linear algebra, however, may help mixed models extend to larger sample sizes. This has already led to a resurgence of research into Gaussian processes, and kernel methods more generally, in the machine learning literature, which are faced with the same apparently prohibitive task of manipulating matrices in $\mathbb{S}_+^N$. Originally applied to kernel approximation in the early 2000s for computational reasons – e.g. Nystrom-type approaches [Williams and Seeger, 2000; Smola and Bartlett, 2001; Drineas and Mahoney, 2005] and incomplete Cholesky [Fine and Scheinberg, 2001] – these approximations have recently been shown to perform essentially as well as exact approaches [Bach, 2013; Rudi, Camoriano, and Rosasco, 2015].

Such approximations could almost certainly be used to expedite operations with the full-rank GRM; less clear, however, is whether they can aid already low-rank aspects of mixed models (e.g the $\{G_m\}_{m>1}$ incidence matrices used in this thesis) or whether approximate decomposition of the GRM can be integrated with GLMM machinery (other than by naively presenting the compressed-rank GRM as another G matrix). Indeed, the incomplete Cholesky algorithm used in the GLMM decomposition is a classic choice for random PSD decomposition, but I used it only for exactly low-rank matrices – it is unclear whether the rank of this decomposition can reliably be further reduced.

Out-of-the-box applications of randomized linear algebra are beginning to attract attention in genetics [Galinsky et al., 2016]. In a similar spirit, [Loh et al.,

2015a; Loh et al., 2015b] use iterative algorithms to manage matrices in $\mathbb{S}_+^N$; although the iterative CG method is run until convergence, this comes at a (claimed) cost of $\sqrt{N}$, which perhaps could be reduced without hurting performance. Finally, the computational regularization used to improve mixed model prediction in 4.4.2 is another example of computational shortcuts performing as well as, or better than, an exact approach.

Mixed model theory

Mixed models are still not theoretically well understood. What happens as the number of loci increases in a finite genome? Is the estimator biased? What data sizes are necessary to approximately obtain asymptotic MLE distributions? What happens when only some loci are causal, especially if these loci are not genotyped? How do random matrices behave in the presence of LD, population structure and family relatedness? We have attempted to partially answer some of these questions in unpublished work [Steinsaltz, Dahl, and Wachter, 2016], but have not attempted to model multiple random effects or phenotypes.

More broadly, it is not always clear how other estimators for heritability perform under these types of model misspecification. Simple moment based estimators, like in Haseman-Elston regression and LD score regression, inherit the stability of linear regression. The only model misspecification studied in detail, to my knowledge, is for binary traits, where moment methods outperform (misspecified) ML [Golan, Lander, and Rosset, 2014]. Nonetheless, these approaches still rely on appropriate moment specification, which will not generally (or, realistically, ever) hold.

A mixed model alternative

Another model I would like to pursue decomposes phenotypes Y using genotypes G:

$$Y = GS + GL + \epsilon$$

where S is a sparse matrix of causal genotype effects and L is a low rank matrix of confounders. Estimates for S and L can be found by solving

$$\min_{S,L} \|Y - GS - GL\|^2 + \lambda_S \|S\|_1 + \lambda_L \|L\|_*$$

Algorithms that scale linearly in the number of samples, phenotypes and loci have been developed [Mu et al., 2014]. With some work, the proximal gradient algorithms proposed there can be generalized to group-sparsity penalties on S to incorporate and detect pleiotropy. The appropriate way to incorporate correlated noise (i.e. non-Frobenius norms on $Y - GS - GL$), however, is far from obvious.

The key advantage to this approach is that GL naturally represents confounding. First, population structure, which is typically modelled with low-rank approximations [Engelhardt and Stephens, 2010], is obviously analogous to the low-rank GL. More generally, simple confounding structure is often modelled as low-rank. This confounding is exactly what the inter-sample correlations in LMMs attempt to address, and so assuming ϵ is row-wise independent–which solves the poor N scaling of mixed models–may be palatable. More generally, both $L+S$ approaches and mixed models decompose signal into foreground (S/fixed effects) and background (L/random effects) and are natural substitutes.

Very broadly, mixed models have clearly been successful in modelling inter-sample relationships. Any satisfying replacement of LMMs, then, must somehow

incorporate these inter-sample relationships. One option is to explicitly model this structure as a fixed effect; this is accomplished above by assuming low rank confounding but could be accomplished in other ways, perhaps by returning to regression on admixture covariates [Yu et al., 2006] or using massive biobanks to control for a massive number of potential confounders.

Appendix A

Additional proofs

A.1 Proof of Section 2 results

Equivalence of partial trace definitions

Two definitions of the partial trace were given in Section 2.2; I show here they are equivalent.

Proposition 5. *The two definitions of the partial trace agree.*

Proof. The primary definition of the partial trace is the matrix of traces of blocks of a matrix:

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

By equation (2.1) (a simple consequence of Lemma 2, proven below), $\mathrm{tr}_P(A \otimes B) = A\mathrm{tr}(B)$ for all $A \in \mathbb{R}^{P\times P}$ and $B \in \mathbb{R}^{N\times N}$; this is precisely the second definition of the partial trace (linearity follows from linearity of the trace and matrix multiplication).

Conversely, suppose $\mathrm{tr}_P(A \otimes B) = A\mathrm{tr}(B)$ for all $A \in \mathbb{R}^{P\times P}$ and $B \in \mathbb{R}^{N\times N}$ and $\mathrm{tr}_P(\cdot)$ is linear. A generic matrix M, using the above block representation,

can be written $M = \sum_{p,q} I_{pq} \otimes M_{(pq)}$. Then the concrete definition of the partial trace can be recovered:

$$\begin{aligned}
\mathrm{tr}_P(M) &= \mathrm{tr}_P\left(\sum_{p,q} I_{pq} \otimes M_{(pq)}\right) \\
&= \sum_{p,q} \mathrm{tr}_P\left(I_{pq} \otimes M_{(pq)}\right) && \text{(by linearity)} \\
&= \sum_{p,q} I_{pq} \mathrm{tr}\left(M_{(pq)}\right) && \text{(by assumption)} \\
&= \begin{bmatrix} \mathrm{tr}\left(M_{(11)}\right) & \dots & \mathrm{tr}\left(M_{(1P)}\right)) \\ \vdots & \ddots & \vdots \\ \mathrm{tr}\left(M_{(P1)}\right) & \dots & \mathrm{tr}\left(M_{(PP)}\right) \end{bmatrix}
\end{aligned}$$

□

KP extraction from the partial trace

Lemma 2.

$$tr_P\left(\left(A \otimes I_N\right) M\right) = A tr_P(M) \tag{A.1}$$

$$tr_N\left(\left(I_P \otimes B\right) M\right) = B tr_N(M) \tag{A.2}$$

Proof. The i,j-th block of a matrix T is given by $\left(e_i \otimes I_N\right)^T T \left(e_j \otimes I_N\right)$, and so

$$\begin{aligned}
\mathrm{tr}_P\left(\left(X \otimes I_N\right) M\right)_{ij} &= \mathrm{tr}\left(\left(e_i \otimes I_N\right)^T \left(X \otimes I_N\right) M \left(e_j \otimes I_N\right)\right) && \text{(by definition)} \\
&= \mathrm{tr}\left(\left(\left(I_{ji} X\right) \otimes I_N\right) M\right) && \text{(the trace is cyclic)} \\
&= \mathrm{tr}\left(\left(I_{ji} X\right) \mathrm{tr}_P(M)\right) && \text{(by (A.3) below)} \\
&= \left(X \mathrm{tr}_P(M)\right)_{ij}
\end{aligned}$$

To my knowledge, this requires an inelegant brute-force computation; this is somewhat expected as I must start from the inelegant matrix-of-traces-of-blocks definition, rather than the simple (and basis invariant) definition of the partial

trace. Using the index convention that $(A \otimes B)_{ij:pq} = A_{ij}B_{pq}$:

$$\begin{aligned}
\mathrm{tr}\left((X \otimes I)\, M\right) &= \sum_{a,b=1}^{P} \sum_{i,j=1}^{N} \left(X^T \otimes I^T\right)_{ab:ij} M_{ab:ij} \\
&= \sum_{a,b=1}^{P} \sum_{i,j=1}^{N} \left(X^T_{ab} I_{ij}\right) M_{ab:ij} \\
&= \sum_{a,b=1}^{P} \left(X^T_{ab}\right) \mathrm{tr}\left(M_{ab:..}\right) \\
&= \mathrm{tr}\left(X \mathrm{tr}_P\left[M\right]\right) \qquad \text{(A.3)}
\end{aligned}$$

The proof of (A.2) is symmetric.

□

Rank-one matrix extraction from the partial trace

A similar extraction rule can be derived for rank-one matrices.

Lemma 3.

$$tr_P\left(uv^T\left[I \otimes X\right]\right) = U^T X^T V \qquad \text{(A.4)}$$

Proof. Define $U = \mathrm{vec}^{-1}(u)$, where vec^{-1} maps the NP vector u to an $N \times P$ matrix U by filling it column-wise. Define the ith multi-index

$$[i] := (i-1) * N + 0 : (N-1)$$

Then $u_{[i]} = U_{,i}$, and

$$\begin{aligned}
&\mathrm{tr}_P\left(uv^T\left[I \otimes X\right]\right)_{ij} = \mathrm{tr}\left(u_{[i]} v^T_{[j]} X\right) = \mathrm{tr}\left(U_{,i} V^T_{,j} X\right) = V^T_{,j} X U_{,i} \implies \\
&\mathrm{tr}_P\left(uv^T\left[I \otimes X\right]\right) = \left(V^T X U\right)^T = U^T X^T V
\end{aligned}$$

□

A.2 Proof of Proposition 3: the Jeffreys prior for matrix factorization

We derive the Jeffreys prior [Jeffreys, 1946] for matrix factorization models with general N, M and P below.

Proposition 3. *Let*

$$Y \sim \mathcal{MN}(S\beta, I, I) \tag{A.5}$$

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 [Nakajima and Sugiyama, 2011], we first show that the flat prior on $U := S\beta$ 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, 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:

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

This leads to expected second derivatives

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

and so the Fisher information is

$$\mathcal{I}\left(\mathrm{vec}\left(S\right),\mathrm{vec}\left(\beta\right)\right) = \begin{pmatrix} (\beta\beta^T)\otimes I_N & \beta \otimes S \\ \beta^T \otimes S^T & I_P \otimes (S^T S) \end{pmatrix} \tag{A.6}$$

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 $U := (U_\beta \otimes U_S) \oplus (V_\beta \otimes V_S)$, where $\oplus$ is the Cartesian product or direct sum:

$$U^T \mathcal{I} U = \begin{pmatrix} D_\beta D_\beta^T \otimes I_N & D_\beta \otimes D_S \\ D_\beta^T \otimes D_S^T & I_P \otimes D_S^T D_S \end{pmatrix} =: \mathcal{I}'$$

Define $\Lambda_\beta = D_\beta D_\beta^T$ and $\Lambda_S = D_S^T D_S$ and let $\lambda_i^S = (\Lambda_S)_{ii}$, $\lambda_i^\beta = (\Lambda_\beta)_{ii}$. Then

the eigenvalues of $\mathcal{I}$ are roots of the characteristic polynomial:

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

As in Appendix 1 of [Nakajima and Sugiyama, 2011], I take the Jeffreys prior proportional to the square root of the product of non-zero eigenvalues of the Fisher information.

□

A.3 Variational characterization of the nuclear norm

Theorem 1 (Well known)**.**

$$
||X||_* = \frac{1}{2} \left(\min_{AB^T = X} ||A||_F^2 + ||B||_F^2 \right) = \min_{AB^T = X} ||A||_F ||B||_F \tag{A.7}
$$

Proof. I prove

$$\frac{1}{2}\left(\min_{AB^T=X}||A||_F^2+||B||_F^2\right)\overset{(1)}{\leq}||X||_*\overset{(2)}{\leq}\min_{AB^T=X}||A||_F||B||_F\overset{(3)}{\leq}\frac{1}{2}\left(\min_{AB^T=X}||A||_F^2+||B||_F^2\right)$$

one inequality at a time.

Write the singular value decomposition $X=UDV^T$.

(1): Take $A=UD^{1/2}$ and $B=VD^{1/2}$. Then

$$\begin{aligned}||A||_F^2+||B||_F^2&=||D^{1/2}||_F^2+||D^{1/2}||_F^2 && \text{(unitary invariance)}\\&=2\sum_i d_i=2||X||_*\end{aligned}$$

(2): Let $AB^T=X$. Then

$$\begin{aligned}||X||_*=||AB^T||_*=||\sigma(AB^T)||_1&\leq\langle\sigma(A),\sigma(B)\rangle && \text{(by (A.8))}\\&\leq||\sigma(A)||_2||\sigma(B)||_2 && \text{(Cauchy-Schwarz)}\\&=||A||_F||B||_F\end{aligned}$$

(3): For any real numbers a and b,

$$0\leq(a-b)^2=(a^2+b^2)-2ab\implies ab\leq\frac{1}{2}(a^2+b^2)$$

and so

$$||A||_F||B||_F\leq\frac{1}{2}\left(||A||_F^2+||B||_F^2\right)$$

for all A and B and thus, in particular,

$$\min_{AB^T=X}||A||_F||B||_F\leq\frac{1}{2}\left(\min_{AB^T=X}||A||_F^2+||B||_F^2\right)$$

□

This uses another well-known result for which I could not find a proof:

Lemma 4.

$$\sum_i \sigma_i(AB^T) \leq \sum_i \sigma_i(A)\sigma_i(B) \tag{A.8}$$

Proof. Write the SVDs $A = U_A D_A V_A^T$ and $U_B D_B V_B^T$, and define the orthogonal matrix $R := V_A^T V_B$. Then

$$\begin{aligned}
\sum_i \sigma_i(AB^T) &= \sum_i \sigma_i(U_A^T A B^T U_B) && \text{(unitary invariance)} \\
&= \sum_i \sigma_i(D_A R D_B) \\
&= \sum_i |D_A^i D_B^i R_{ii}| && \text{(cylic property of the trace)} \\
&\leq \sum_i D_A^i D_B^i && (\|R\|_\infty \leq 1) \\
&= \sum_i \sigma_i(A)\sigma_i(B)
\end{aligned}$$

Moreover, the inequality is tight if and only if all $R_{ii} = \pm 1$ for all i; that is, the inequality is tight if and only if A and B share right singular vectors. □

A.4 Analytic penalized MLEs for Gaussian covariance estimation

Frobenius penalization

For some sample covariance S, the objective is

$$\min_{X \succeq 0} \mathcal{O} := -\log|X| + \operatorname{tr}(SX) + \frac{\rho}{2}||X||_F^2$$

Let $X = QDQ^T$ be an eigendecomposition of X, so the first order condition is

$$\begin{gathered}
\nabla_X \mathcal{O} = -X^{-1} + S + \rho X = 0 \iff \\
-I + (Q^T S Q)D + \rho D^2 = 0
\end{gathered}$$

Since I and D are diagonal, this immediately implies $(Q^T S Q)$ is diagonal or, in other words, that X and S share eigenvectors (though this fact is often cited without proof, we give it here for completeness). Up to permutation of the eigenvalues, this uniquely defines Q. We now assume the eigenvectors of X and S are sorted in the same order; this comes without loss of generality as the order of the eigenvalues d remains free.

All that remains is to optimize D, the eigenvalues of X:

$$\nabla_X \mathcal{O} = 0 \iff d_i^* = \frac{-d_i^S \pm \sqrt{(d_i^S)^2 + 4\rho}}{2\rho}$$

where d_S are the eigenvalues of S[1]. But since $d_i^* \geq 0$, the $\pm$ must simply be a $+$, and so

$$d_i^* = \frac{\sqrt{(d_i^S)^2 + 4\rho} - d_i^S}{2\rho} \tag{A.9}$$

which is non-negative for $\rho \geq 0$, as desired.

Variable-independence constraint

The independence constraint is easily incorporated into the optimization problem:

$$\begin{aligned}
\min_{X \succeq 0} \mathcal{O} &:= -\log|X| + \operatorname{tr}(SX) + \mathbb{I}\{X \in \mathcal{D}\} \implies \\
\min_{x_p \geq 0} \mathcal{O} &:= -\sum_p \log x_p + \sum_p x_p S_{pp} \implies \\
x_p^* &= S_{pp}^{-1}
\end{aligned}$$

As expected, then, the penalized MLE precision matrix–diag(x^*)–is just the diagonal part of the sample precision.

[1] In general, the index i on the left-hand side need not be the same as the index i on the right but, rather, the i-th index of some permutation vector–the equality occurs because we assumed above that the eigenvectors of S and X are sorted in the same order.

A.5 Conjugate gradient descent to circumvent matrix inversion

Conjugate gradient descent can be used to solve the following problem for any A and y:

$$\min_x \|Ax - y\|^2$$

This convex problem has a unique optimum at $x = A^{-1}y$, and thus the conjugate gradient method converges to $A^{-1}y$. As the conjugate gradient iterations will require A only via operations of the form Az, this can be an efficient way to solve a linear system when inverting A is dramatically more expensive than multiplying with A.

Conjugate gradient descent for the above optimization iteratively performs the following updates:

$$\begin{aligned}
x_{k+1} &= x_k + \frac{||r_k||_2^2}{p_k^T q_k} p_k && \text{(update)} \\
r_{k+1} &= r_k - \frac{||r_k||_2^2}{p_k^T q_k} q_k && \text{(residual)} \\
p_{k+1} &= r_{k+1} + \frac{||r_{k+1}||_2^2}{||r_k||_2^2} p_k && \text{(next step)} \\
q_{k+1} &= A p_{k+1}
\end{aligned}$$

The first three steps consist of cheap $O(|y|)$ operations: additions and inner products of length-$|y|$ vectors. The total cost is thus dominated by the fourth step, which evaluates Ap_k. Though this is already cheaper than inverting A as written ($O(|y|^2)$ instead of $O(|y|^3)$), the real advantage is problem specific: in the above setting, the latent algebraic structure of A^{-1} is hard to utilize but the simplification is immediate when using A.

As described in Section 4.3.5, this can be directly used to evaluate $\Lambda_y y$ by taking $A = \Sigma_y$.

A.6 Variational Bayes overview

Variational Bayes (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 (less reliably) covariances and marginal likelihoods.

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

$$\log P(D) \quad = \quad F(Q) + D_{KL}(Q||P(\cdot|D)) \tag{A.10}$$

where D_{KL} is the Kullback-Leibler 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.

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:

$$Q_i \leftarrow \arg\max_{Q_i'} F(Q_i', Q_{-i}) \tag{A.11}$$

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

$$\log Q_i \leftarrow \arg\max_{Q_i} F(Q_i, Q_{-i}) \equiv \mathbb{E}_{\theta_{-i} \sim Q_{-i}} \Big(\log P(D, \theta) \Big) \tag{A.12}$$

The precise form of each Q_i is determined automatically by the variational equation (A.12), not specified *a priori*. 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 (A.11), which in general optimizes Q_i over a function space, reduces to optimizing $\tilde{\theta}_i$, which is a parsimonious representation of Q_i.

Bibliography

Abdollahi Arpanahi, R, A Pakdel, A Nejati Javaremi, M Moradi Shahrbabak, G Morota, B D Valente, A Kranis, G J M Rosa, and D Gianola (2014). Dissection of additive genetic variability for quantitative traits in chickens using SNP markers. *Journal of Animal Breeding and Genetics* 131.3, pp. 183–193.

Abecasis, Goncalo R, Lon R Cardon, William O Cookson, Pak C Sham, and Stacey S Cherny (2001). Association analysis in a variance components framework. *Genetic Epidemiology* 21 Suppl 1, S341–6.

Abney, Mark (2015). Permutation testing in the presence of polygenic variation. *Genetic Epidemiology* 39.4, pp. 249–258.

Abney, Mark, Carole Ober, and Mary Sara McPeek (2002). Quantitative-Trait Homozygosity and Association Mapping and Empirical Genomewide Significance in Large, Complex Pedigrees: Fasting Serum-Insulin Level in the Hutterites. *The American Journal of Human Genetics* 70.4, pp. 920–934.

Aerts, Diederik and Ingrid Daubechies (1978). Physical justification for using the tensor product to describe two quantum systems as one joint system. *Helv Phys Acta*.

Alexandrov, Vadim, Dani Brunner, Liliana B Menalled, Andrea Kudwa, Judy Watson-Johnson, Matthew Mazzella, Ian Russell, Melinda C Ruiz, Justin Torello, Emily Sabath, Ana Sanchez, Miguel Gomez, Igor Filipov, Kimberly Cox, Mei Kwan, Afshin Ghavami, Sylvie Ramboz, Brenda Lager, Vanessa C Wheeler, Jeff Aaronson, Jim Rosinski, James F Gusella, Marcy E MacDonald, David Howland, and Seung Kwak (2016). Large-scale phenome analysis defines a behavioral signature for Huntington's disease genotype in mice. *Nature Biotechnology* 34.8, pp. 838–844.

Allen, Genevera I and Robert Tibshirani (2010). Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics* 4.2, pp. 764–790.

Almasy, Laura and John Blangero (1998). Multipoint quantitative-trait linkage analysis in general pedigrees. *The American Journal of Human Genetics* 62.5, pp. 1198–1211.

Andrews, Nancy C (2009). Genes determining blood cell traits. *Nature Genetics* 41.11, pp. 1161–1162.

Aschard, Hugues, Bjarni J Vilhjálmsson, Nicolas Greliche, Pierre-Emmanuel Morange, David-Alexandre Tregouet, and Peter Kraft (2014). Maximizing the Power of Principal-Component Analysis of Correlated Phenotypes in Genome-wide Association Studies. *The American Journal of Human Genetics* 94.5, pp. 662–676.

Aschard, Hugues, Bjarni Vilhjalmsson, Chirag Patel, David Skurnik, Jimmy Yu, Brian Wolpin, Peter Kraft, and Noah Zaitlen (2016). Playing Musical Chairs in Big Data to Reveal Variables Associations. *BioRxiv*.

Aulchenko, Yurii S, Dirk-Jan de Koning, and Chris Haley (2007). Genomewide Rapid Association Using Mixed Model and Regression: A Fast and Simple Method For Genomewide Pedigree-Based Quantitative Trait Loci Association Analysis. *Genetics*.

Bach, Francis R (2013). Sharp analysis of low-rank kernel matrix approximations. *COLT*.

Bach, Francis R and Michael I Jordan (2003). Kernel independent component analysis. *The Journal of Machine Learning Research*.

Banerjee, Onureena, Laurent El Ghaoui, and Alexandre d'Aspremont (2008). Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. *The Journal of Machine Learning Research* 9, pp. 485–516.

Bernier, Raphael, Christelle Golzio, Bo Xiong, Holly A Stessman, Bradley P Coe, Osnat Penn, Kali Witherspoon, Jennifer Gerdts, Carl Baker, Anneke T Vulto-van Silfhout, et al. (2014). Disruptive CHD8 Mutations Define a Subtype of Autism Early in Development. *Cell* 158.2, pp. 263–276.

Bhatia, Gaurav, Alexander Gusev, Po-Ru Loh, Bjarni J Vilhjálmsson, Stephan Ripke, Schizophrenia Working Group Psychiatric Genomics C, Shaun Purcell, Eli Stahl, Mark Daly, Teresa R de Candia, Kenneth S Kendler, Michael C O'Donovan, Sang Hong Lee, Naomi R Wray, Benjamin M Neale, Matthew C Keller, Noah A Zaitlen, Bogdan Pasaniuc, Jian Yang, and Alkes L Price (2015). Haplotypes of common SNPs can explain missing heritability of complex diseases. *BioRxiv*, pp. 1–25.

Bjerrum, O W, O J Bjerrum, and N Borregaard (1987). Beta 2-microglobulin in neutrophils: an intragranular protein. *Journal of immunology (Baltimore, Md. : 1950)* 138.11, pp. 3913–3917.

Bloom, Joshua S, Ian M Ehrenreich, Wesley T Loo, Thúy-Lan Võ Lite, and Leonid Kruglyak (2013). Finding the sources of missing heritability in a yeast cross. *Nature*, pp. 1–6.

Bruining, Hilgo, Marinus JC Eijkemans, Martien JH Kas, Sarah R Curran, Jacob AS Vorstman, and Patrick F Bolton (2014). Behavioral signatures related to genetic disorders in autism. *Molecular Autism* 5.1, p. 1.

Bulik-Sullivan, Brendan, Hilary K Finucane, Verneri Anttila, Alexander Gusev, Felix R Day, Po-Ru Loh, ReproGen Consortium, Psychiatric Genomics Consortium, Genetic Consortium for Anorexia Nervosa of the Wellcome Trust Case Control Consortium 3, Laramie Duncan, John R B Perry, Nick Patterson, Elise B Robinson, Mark J Daly, Alkes L Price, and Benjamin M Neale (2015a). An atlas of genetic correlations across human diseases and traits. *Nature Genetics* 47.11, pp. 1236–1241.

Bulik-Sullivan, Brendan K, Po-Ru Loh, Hilary K Finucane, Stephan Ripke, Jian Yang, Schizophrenia Working Group of the Psychiatric Genomics Consortium, Nick Patterson, Mark J Daly, Alkes L Price, and Benjamin M Neale (2015b). LD Score regression distinguishes confounding from polygenicity in genome-wide association studies. *Nature Genetics* 47.3, pp. 291–295.

Buuren, Stef van and Karin Groothuis-Oudshoorn (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Computational and Graphical Statistics*.

Campos, Gustavo de los and Daniel Gianola (2007). Factor analysis models for structuring covariance matrices of additive genetic effects: a Bayesian implementation. 39.5, pp. 481–494.

Campos, Gustavo de los, Daniel Gianola, and David B Allison (2010). Predicting genetic predisposition in humans: the promise of whole-genome markers. *Nature Reviews Genetics* 11.12, pp. 880–886.

Campos, Gustavo de los, Daniel Sorensen, and Daniel Gianola (2015). Genomic Heritability: What Is It? *PLoS Genetics* 11.5, e1005048.

Candes, Emmanuel and Terrence Tao (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*.

Candès, Emmanuel J and Benjamin Recht (2009). Exact Matrix Completion via Convex Optimization. *Foundations of Computational mathematics* 9.6, pp. 717–772.

Candès, Emmanuel J and Terence Tao (2010). The Power of Convex Relaxation: Near-Optimal Matrix Completion. *IEEE Transactions on Information Theory* 56.5, pp. 2053–2080.

Caruana, Rich, Steve Lawrence, and C Lee Giles (2000). Overfitting in Neural Nets - Backpropagation, Conjugate Gradient, and Early Stopping. *NIPS*.

Casale, Francesco Paolo, Barbara Rakitsch, Christoph Lippert, and Oliver Stegle (2015). Efficient set tests for the genetic analysis of correlated traits. *Nature* 12.8, pp. 755–758.

Cebamanos, L, A Gray, I Stewart, and A Tenesa (2014). Regional Heritability Advanced Complex Trait Analysis for GPU and Traditional Parallel Architectures. *Bioinformatics* 30.8, btt754–1179.

Chen, Xi, Han Liu, and Jaime G Carbonell (2012). Structured Sparse Canonical Correlation Analysis. *AISTATS*.

Chi, Eric C, Genevera I Allen, Hua Zhou, Omid Kohannim, Kenneth Lange, and Paul M Thompson (2013). "Imaging genetics via sparse canonical correlation analysis". *2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI 2013)*. IEEE, pp. 740–743.

Choi, Yun-Hee, Rafiqul Chowdhury, and Balakumar Swaminathan (2014). Prediction of hypertension based on the genetic analysis of longitudinal phenotypes: a comparison of different modeling approaches for the binary trait of hypertension. *BMC Proceedings* 8.1, p. 1.

Cotsapas, Chris, Benjamin F Voight, Elizabeth Rossin, Kasper Lage, Benjamin M Neale, Chris Wallace, Gonçalo R Abecasis, Jeffrey C Barrett, Timothy Behrens, Judy Cho, Philip L De Jager, James T Elder, Robert R Graham, Peter Gregersen, Lars Klareskog, Katherine A Siminovitch, David A van Heel, Cisca Wijmenga, Jane Worthington, John A Todd, David A Hafler, Stephen S Rich, Mark J Daly, and on behalf of the FOCiS Network of Consortia (2011). Pervasive Sharing of Genetic Effects in Autoimmune Disease. *PLoS Genetics* 7.8, e1002254–8.

Cristianini, Nello, John Shawe-Taylor, and Huma Lodhi (2002). Latent Semantic Kernels. *Journal of Intelligent Information Systems* 18.2-3.

Cross-Disorder Group of the Psychiatric Genomics Consortium, Sang Hong Lee, Stephan Ripke, Benjamin M Neale, Stephen V Faraone, Shaun M Purcell, Roy H Perlis, Bryan J Mowry, Anita Thapar, Michael E Goddard, et al. (2013). Genetic relationship between five psychiatric disorders estimated from genome-wide SNPs. *Nature Genetics* 45.9, pp. 984–994.

Cui, Xiaoqi, Qiuying Sha, Shuanglin Zhang, and Huann-Sheng Chen (2009). A combinatorial approach for detecting gene-gene interaction using multiple traits of Genetic Analysis Workshop 16 rheumatoid arthritis data. *BMC Proceedings* 3.7, p. 1.

Dahl, Andrew, Victoria Hore, Valentina Iotchkova, and Jonathan Marchini (2013). Network inference in matrix-variate Gaussian models with non-independent noise. *arXiv.org*.

Dahl, Andrew, Valentina Iotchkova, Amelie Baud, Åsa Johansson, Ulf Gyllensten, Nicole Soranzo, Richard Mott, Andreas Kranis, and Jonathan Marchini (2016). A multiple-phenotype imputation method for genetic studies. *Nature Genetics* 48.4, pp. 466–472.

Dahl, David B (2009). xtable: Export Tables to LaTeX or HTML. *R package*.

Dawid, A P (1981). Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika* 68.1, pp. 265–274.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

Denny, Joshua C, Marylyn D Ritchie, Melissa A Basford, Jill M Pulley, Lisa Bastarache, Kristin Brown-Gentry, Deede Wang, Dan R Masys, Dan M Roden, and Dana C Crawford (2010). PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene-disease associations. *Bioinformatics* 26.9, pp. 1205–1210.

Denny, Joshua C, Dana C Crawford, Marylyn D Ritchie, Suzette J Bielinski, Melissa A Basford, Yuki Bradford, High Seng Chai, Lisa Bastarache, Rebecca Zuvich, Peggy Peissig, et al. (2011). Variants Near FOXE1 Are Associated with Hypothyroidism and Other Thyroid Conditions: Using Electronic Medical Records for Genome- and Phenome-wide Studies. *The American Journal of Human Genetics* 89.4, pp. 529–542.

Denny, Joshua C, Lisa Bastarache, Marylyn D Ritchie, Robert J Carroll, Raquel Zink, Jonathan D Mosley, Julie R Field, Jill M Pulley, Andrea H Ramirez, Erica Bowton, et al. (2013). Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. *Nature Biotechnology* 31.12, pp. 1102–1111.

Drineas, Petros and Michael W Mahoney (2005). On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *Journal of Machine Learning Research* 6.Dec, pp. 2153–2175.

Du, Lei, Heng Huang, Jingwen Yan, Sungeun Kim, Shannon L Risacher, Mark Inlow, Jason H Moore, Andrew J Saykin, Li Shen, and Alzheimer's Disease Neuroimaging Initiative (2016). Structured sparse canonical correlation analysis for brain imaging genetics: an improved GraphNet method. *Bioinformatics* 32.10, pp. 1544–1551.

Dueck, Delbert, Q D Morris, and B J Frey (2005). Multi-way clustering of microarray data using probabilistic sparse matrix factorization. *Bioinformatics* 21.Suppl 1, pp. i144–i151.

Eddelbuettel, Dirk (2013). *Seamless R and C++ Integration with Rcpp*. New York, NY: Springer New York.

Eddelbuettel, Dirk and Romain François (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*.

Engelhardt, Barbara E and Matthew Stephens (2010). Analysis of Population Structure: A Unifying Framework and Novel Methods Based on Sparse Factor Analysis. *PLoS Genetics* 6.9, e1001117–12.

Falconer, Douglas S and Trudy FC Mackay (1996). *Introduction to quantitative genetics — Clc*. Harlow: Addison Wesley Longman.

Ferreira, Manuel A R and Shaun M Purcell (2008). A multivariate test of association. *Bioinformatics* 25.1, pp. 132–133.

Fine, Shai and Katya Scheinberg (2001). Efficient SVM Training Using Low-Rank Kernel Representations. *Journal of Machine Learning Research* 2.Dec, pp. 243–264.

Finucane, Hilary K, Brendan Bulik-Sullivan, Alexander Gusev, Gosia Trynka, Yakir Reshef, Po-Ru Loh, Verneri Anttila, Han Xu, Chongzhi Zang, Kyle Farh, Stephan Ripke, Felix R Day, Shaun Purcell, Eli Stahl, Sara Lindström, John R B Perry, Yukinori Okada, Soumya Raychaudhuri, Mark J Daly, Nick Patterson, Benjamin M Neale, and Alkes L Price (2015). Partitioning heritability by functional annotation using genome-wide association summary statistics. *Nature Genetics*.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9.3, pp. 432–441.

Furlotte, Nicholas A, David Heckerman, and Christoph Lippert (2014). Quantifying the uncertainty in heritability. *Journal of human genetics* 59.5, pp. 269–275.

Fusi, Nicolo, Christoph Lippert, Neil D Lawrence, and Oliver Stegle (2014). Warped linear mixed models for the genetic analysis of transformed phenotypes. *Nature communications* 5, p. 4890.

Galinsky, Kevin J, Gaurav Bhatia, Po-Ru Loh, Stoyan Georgiev, Sayan Mukherjee, Nick J Patterson, and Alkes L Price (2016). Fast Principal-Component Analysis Reveals Convergent Evolution of ADH1B in Europe and East Asia. *American journal of human genetics* 98.3, pp. 456–472.

Ganesh, Santhi K, Neil A Zakai, Frank J A van Rooij, Nicole Soranzo, Albert V Smith, Michael A Nalls, Ming-Huei Chen, Anna Köttgen, Nicole L Glazer, Abbas Dehghan, et al. (2009). Multiple loci influence erythrocyte phenotypes in the CHARGE Consortium. *Nature Genetics* 41.11, pp. 1191–1198.

Ge, Tian, Thomas E Nichols, Phil H Lee, Avram J Holmes, Joshua L Roffman, Randy L Buckner, Mert R Sabuncu, and Jordan W Smoller (2015). Massively expedited genome-wide heritability analysis (MEGHA). *Proceedings of the National Academy of Sciences* 112.8, pp. 2479–2484.

Ge, Tian, Chia-Yen Chen, Benjamin M Neale, Mert R Sabuncu, and Jordan W Smoller (2016). Phenome-wide Heritability Analysis of the UK Biobank. *BioRxiv*.

Gianola, Daniel (2013). Priors in whole-genome regression: the bayesian alphabet returns. *Genetics* 194.3, pp. 573–596.

Gianola, Daniel, Gustavo de los Campos, William G Hill, Eduardo Manfredi, and Rohan Fernando (2009). Additive genetic variability and the Bayesian alphabet. *Genetics* 183.1, pp. 347–363.

Gianola, Daniel, Gustavo de los Campos, Miguel A Toro, Hugo Naya, Chris-Carolin Schön, and Daniel Sorensen (2015). Do Molecular Markers Inform About Pleiotropy? *Genetics* 201.1, pp. 23–29.

Giordano, Ryan, Tamara Broderick, and Michael I Jordan (2015). Linear Response Methods for Accurate Covariance Estimates from Mean Field Variational Bayes. *NIPS*.

Glicksberg, Benjamin S, Li Li, Marcus A Badgeley, Khader Shameer, Roman Kosoy, Noam D Beckmann, Nam Pho, Jörg Hakenberg, Meng Ma, Kristin L Ayers, Gabriel E Hoffman, Shuyu Dan Li, Eric E Schadt, Chirag J Patel, Rong Chen, and Joel T Dudley (2016). Comparative analyses of population-scale phenomic data in electronic medical records reveal race-specific disease networks. *Bioinformatics* 32.12, pp. i101–i110.

Goddard, Mike (2009). Genomic selection: prediction of accuracy and maximisation of long term response. *Genetica* 136.2, pp. 245–257.

Golan, David, Eric S Lander, and Saharon Rosset (2014). Measuring missing heritability: inferring the contribution of common variants. *Proceedings of the National Academy of Sciences of the United States of America* 111.49, E5272–81.

Golan, David and Saharon Rosset (2011). Accurate estimation of heritability in genome wide studies using random effects models. *Bioinformatics* 27.13, pp. i317–23.

Golan, David, Saharon Rosset, and Eric S Lander (2015). Reply to Lee: Downward bias in heritability estimation is not due to simplified linkage equilibrium SNP simulation. *Proceedings of the National Academy of Sciences of the United States of America* 112.40, E5452–3.

Granot-Attas, Shira, Hilla Knobler, and Ari Elson (2007). Protein Tyrosine Phosphatases in Osteoclasts. *Critical Reviews^{TM} in Eukaryotic Gene Expression* 17.1, pp. 49–72.

Gray, A, I Stewart, and A Tenesa (2012). Advanced complex trait analysis. *Bioinformatics* 28.23, pp. 3134–3136.

Grey, ADNJ de (2010). Commentary on some recent theses relevant to combating aging: February 2010. *Rejuvenation Research*.

Guan, Yue and Jennifer G Dy (2009). Sparse Probabilistic Principal Component Analysis. *AISTATS*.

Hai, Rong, Lei Zhang, Yufang Pei, LanJuan Zhao, Shu Ran, YingYing Han, XueZhen Zhu, Hui Shen, Qing Tian, and HongWen Deng (2012). Bivariate genome-wide association study suggests that the DARC gene influences lean

body mass and age at menarche. *Science China Life Sciences* 55.6, pp. 516–520.

Harbrecht, Helmut, Michael Peters, and Reinhold Schneider (2012). On the low-rank approximation by the pivoted Cholesky decomposition. *Applied Numerical Mathematics* 62.4.

Hastie, Trevor, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, and David Botstein (1999). *Imputing missing data for gene expression arrays*. Tech. rep.

Hastie, Trevor, Rahul Mazumder, Jason D Lee, and Reza Zadeh (2015). Matrix completion and low-rank SVD via fast alternating least squares. *Journal of Machine Learning Research*.

Hayeck, Tristan, Noah A Zaitlen, Po-Ru Loh, Samuela Pollack, Alexander Gusev, Nick Patterson, and Alkes Price (2016). Mixed Model Association with Family-Biased Case-Control Ascertainment. *BioRxiv*, p. 046995.

Hayeck, Tristan J, Noah A Zaitlen, Po-Ru Loh, Bjarni Vilhjalmsson, Samuela Pollack, Alexander Gusev, Jian Yang, Guo-Bo Chen, Michael E Goddard, Peter M Visscher, Nick Patterson, and Alkes L Price (2015). Mixed Model with Correction for Case-Control Ascertainment Increases Association Power. *The American Journal of Human Genetics* 96.5, pp. 720–730.

Hayes, Ben J, Peter M Visscher, and Michael E Goddard (2009). Increased accuracy of artificial selection by using the realized relationship matrix. *Genetics Research* 91.1, pp. 47–60.

Heckerman, David, Deepti Gurdasani, Carl Kadie, Cristina Pomilla, Tommy Carstensen, Hilary Martin, Kenneth Ekoru, Rebecca N Nsubuga, Gerald Ssenyomo, Anatoli Kamali, Pontiano Kaleebu, Christian Widmer, and Manjinder S Sandhu (2016). Linear mixed model for heritability estimation that explicitly addresses environmental variation. *Proceedings of the National Academy of Sciences* 113.27, pp. 7377–7382.

Henderson, Charles R (1975). Comparison of Alternative Sire Evaluation Methods. *Journal of Animal Science* 41.3, pp. 760–770.

Hore, Victoria, Andrew Dahl, Ana Vinuela, Alfonso Buil, Julian C Knight, Mark I McCarthy, and Jonathan Marchini (2015). Tensor decomposition for multi-tissue gene expression. *MLCB*.

Hormozdiari, Farhad, Eun Yong Kang, Michael Bilow, Eyal Ben-David, Chris Vulpe, Stela McLachlan, Aldons J Lusis, Buhm Han, and Eleazar Eskin (2016). Imputing Phenotypes for Genome-wide Association Studies. *The American Journal of Human Genetics* 99.1, pp. 89–103.

Howie, Bryan N, Peter Donnelly, and Jonathan Marchini (2009). A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genetics* 5.6, e1000529.

Hsu, Yi-Hsiang, M Carola Zillikens, Scott G Wilson, Charles R Farber, Serkalem Demissie, Nicole Soranzo, Estelle N Bianchi, Elin Grundberg, Liming Liang, J Brent Richards, Karol Estrada, Yanhua Zhou, Atila van Nas, Miriam F Moffatt, Guangju Zhai, Albert Hofman, Joyce B van Meurs, Huibert A P Pols, Roger I Price, Olle Nilsson, Tomi Pastinen, L Adrienne Cupples, Aldons J Lusis, Eric E Schadt, Serge Ferrari, André G Uitterlinden, Fernando Rivadeneira, Timothy D Spector, David Karasik, and Douglas P Kiel (2010). An Integration of Genome-Wide Association Study and Gene Expression Profiling to Prioritize the Discovery of Novel Susceptibility Loci for Osteoporosis-Related Traits. *PLoS Genetics* 6.6, e1000977.

Huffman, Jennifer E, Ana Knežević, Veronique Vitart, Jayesh Kattla, Barbara Adamczyk, Mislav Novokmet, Wilmar Igl, Maja Pučić, Lina Zgaga, Åsa Johannson, Irma Redžić, Olga Gornik, Tatijana Zemunik, Ozren Polašek, Ivana Kolčić, Marina Pehlić, Carolien A M Koeleman, Susan Campbell, Sarah H Wild, Nicholas D Hastie, Harry Campbell, Ulf Gyllensten, Manfred Wuhrer, James F Wilson, Caroline Hayward, Igor Rudan, Pauline M Rudd, Alan F Wright, and Gordan Lauc (2011). Polymorphisms in B3GAT1, SLC9A9 and MGAT5 are associated with variation within the human plasma N-glycome of 3533 European adults. *Human Molecular Genetics* 20.24, ddr414–5011.

Ilin, Alexander and Tapani Raiko (2010). Practical Approaches to Principal Component Analysis in the Presence of Missing Values. *Journal of Machine Learning Research* 11.Jul, pp. 1957–2000.

Jeffreys, Harold (1946). "An invariant form for the prior probability in estimation problems". *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pp. 453–461.

Joetham, Anthony, Katsuyuki Takeda, Nobuaki Miyahara, Shigeki Matsubara, Hiroshi Ohnishi, Toshiyuki Koya, Azzeddine Dakhama, and Erwin W Gelfand (2007). Activation of naturally occurring lung CD4(+)CD25(+) regulatory T cells requires CD8 and MHC I interaction. *Proceedings of the National Academy of Sciences* 104.38, pp. 15057–15062.

Kalaitzis, Alfredo, John Lafferty, Neil Lawrence, and Shuheng Zhou (2013). "The Bigraphical Lasso". *Proceedings of the th International Conference on Machine Learning*, pp. 1229–1237.

Kamatani, Yoichiro, Koichi Matsuda, Yukinori Okada, Michiaki Kubo, Naoya Hosono, Yataro Daigo, Yusuke Nakamura, and Naoyuki Kamatani (2010). Genome-wide association study of hematological and biochemical traits in a Japanese population. *Nature Genetics* 42.3, pp. 210–215.

Kang, Hyun Min, Noah A Zaitlen, Claire M Wade, Andrew Kirby, David Heckerman, Mark J Daly, and Eleazar Eskin (2008). Efficient control of population

structure in model organism association mapping. *Genetics* 178.3, pp. 1709–1723.

Kang, Hyun Min, Jae Hoon Sul, Susan K Service, Noah A Zaitlen, Sit-yee Kong, Nelson B Freimer, Chiara Sabatti, and Eleazar Eskin (2010). Variance component model to account for sample structure in genome-wide association studies. *Nature Genetics* 42.4, pp. 348–354.

Kim, Yong-Deok and Seungjin Choi (2013). Variational Bayesian View of Weighted Trace Norm Regularization for Matrix Factorization. *IEEE Signal Processing Letters* 20.3, pp. 261–264.

Klei, Lambertus, Diana Luca, Bernie Devlin, and Kathryn Roeder (2008). Pleiotropy and principal components of heritability combine to increase power for association analysis. *Genetic Epidemiology* 32.1, pp. 9–19.

Korte, Arthur, Bjarni J Vilhjálmsson, Vincent Segura, Alexander Platt, Quan Long, and Magnus Nordborg (2012). A mixed-model approach for genome-wide association studies of correlated traits in structured populations. *Nature Genetics* 44.9, pp. 1066–1071.

Kostem, Emrah and Eleazar Eskin (2013). Improving the Accuracy and Efficiency of Partitioning Heritability into the Contributions of Genomic Regions. *The American Journal of Human Genetics* 92.4, pp. 558–564.

Kumar, Siddharth Krishna, Marcus W Feldman, David H Rehkopf, and Shripad Tuljapurkar (2016a). Limitations of GCTA as a solution to the missing heritability problem. *Proceedings of the National Academy of Sciences* 113.1, E61–E70.

Kumar, Siddharth Krishna, Marcus W Feldman, David H Rehkopf, and Shripad Tuljapurkar (2016b). Response to Commentary on "Limitations of GCTA as a solution to the missing heritability problem". *BioRxiv*, p. 039594.

Lauc, Gordan, 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 (2010). Genomics Meets Glycomics—The First GWAS Study of Human N-Glycome Identifies HNF1α as a Master Regulator of Plasma Protein Fucosylation. *PLoS Genetics* 6.12, e1001256–14.

Lauritzen, Steffen L (1996). *Graphical Models*. Clarendon Press.

Lawson, Daniel John, Garrett Hellenthal, Simon Myers, and Daniel Falush (2012). Inference of Population Structure using Dense Haplotype Data. *PLoS Genetics* 8.1, e1002453.

Lee, Sang Hong (2015). Implications of simplified linkage equilibrium SNP simulation. *Proceedings of the National Academy of Sciences of the United States of America* 112.40, E5449–51.

Lee, Sang Hong, Naomi R Wray, Michael E Goddard, and Peter M Visscher (2011). Estimating missing heritability for disease from genome-wide association studies. *American journal of human genetics* 88.3, pp. 294–305.

Lee, Sang Hong, Teresa R DeCandia, Stephan Ripke, Jian Yang, The Schizophrenia Psychiatric Genome-Wide Association Study Consortium PGC-SCZ, The International Schizophrenia Consortium ISC, The Molecular Genetics of Schizophrenia Collaboration MGS, Patrick F Sullivan, Michael E Goddard, Matthew C Keller, Peter M Visscher, and Naomi R Wray (2012a). Estimating the proportion of variation in susceptibility to schizophrenia captured by common SNPs. *Nature Genetics* 44.3, pp. 247–250.

Lee, Sang Hong, Denise Harold, Dale R Nyholt, Michael E Goddard, Krina T Zondervan, Julie Williams, Grant W Montgomery, Naomi R Wray, and Peter M Visscher (2012b). Estimation and partitioning of polygenic variation captured by common SNPs for Alzheimer's disease, multiple sclerosis and endometriosis. *Human Molecular Genetics* 22.4, dds491–841.

Lee, Sang Hong, Jian Yang, Guo-Bo Chen, Stephan Ripke, Eli A Stahl, Christina M Hultman, Pamela Sklar, Peter M Visscher, Patrick F Sullivan, Michael E Goddard, and Naomi R Wray (2013). Estimation of SNP Heritability from Dense Genotype Data. *The American Journal of Human Genetics* 93.6, pp. 1151–1155.

Legarra, A and I Misztal (2008). Technical Note: Computing Strategies in Genome-Wide Selection. *Journal of Dairy Science* 91.1, pp. 360–366.

Leng, Chenlei and Cheng Yong Tang (2012). Sparse Matrix Graphical Models. *Journal of the American Statistical Association* 107.499, pp. 1187–1200.

Li, Li, Wei-Yi Cheng, Benjamin S Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P Bottinger, and Joel T Dudley (2015). Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science Translational Medicine* 7.311, 311ra174–311ra174.

Lim, Yew J and Yee Whye Teh (2007). "Variational Bayesian approach to movie rating prediction". *Proceedings of KDD cup and workshop*.

Lin, Dongdong, Vince D Calhoun, and Yu-Ping Wang (2014). Correspondence between fMRI and SNP data by group sparse canonical correlation analysis. *Medical image analysis* 18.6, pp. 891–902.

Lippert, Christoph, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman (2011). FaST linear mixed models for genome-wide association studies. *Nature* 8.10, pp. 833–835.

Lippert, Christoph, Gerald Quon, Eun Yong Kang, Carl M Kadie, Jennifer Listgarten, and David Heckerman (2013). The benefits of selecting phenotype-specific variants for applications of mixed models in genomics. *Scientific reports* 3, p. 1815.

Lippert, Christoph, Francesco P Casale, Barbara Rakitsch, and Oliver Stegle (2014). LIMIX: genetic analysis of multiple traits. *BioRxiv*.

Listgarten, Jennifer, Christoph Lippert, and David Heckerman (2013). FaST-LMM-Select for addressing confounding from spatial structure and rare variants. *Nature Genetics* 45.5, pp. 470–471.

Listgarten, Jennifer, Christoph Lippert, Carl M Kadie, Robert I Davidson, Eleazar Eskin, and David Heckerman (2012). Improved linear mixed models for genome-wide association studies. *Nature* 9.6, pp. 525–526.

Listgarten, Jennifer, Christoph Lippert, Eun Yong Kang, Jing Xiang, Carl M Kadie, and David Heckerman (2013). A powerful and efficient set test for genetic markers that handles confounders. *Bioinformatics* 29.12, pp. 1526–1533.

Little, Roderick J A and Donald B Rubin (1987). *Statistical Analysis With Missing Data*. John Wiley & Sons.

Liu, Dehua, Tengfei Zhou, Hui Qian, Congfu Xu, and Zhihua Zhang (2013). "A Nearly Unbiased Matrix Completion Approach". *ECML PKDD 2013: Proceedings, Part II, of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 8189*. Zhejiang University. Berlin, Heidelberg: Springer-Verlag New York, Inc., pp. 210–225.

Loh, Po-Ru, Gaurav Bhatia, Alexander Gusev, Hilary K Finucane, Brendan K Bulik-Sullivan, Samuela J Pollack, Schizophrenia Working Group of Psychiatric Genomics Consortium, Teresa R de Candia, Sang Hong Lee, Naomi R Wray, Kenneth S Kendler, Michael C O'Donovan, Benjamin M Neale, Nick Patterson, and Alkes L Price (2015a). Contrasting genetic architectures of schizophrenia and other complex diseases using fast variance-components analysis. *Nature Genetics* 47.12, pp. 1385–1392.

Loh, Po-Ru, George Tucker, Brendan K Bulik-Sullivan, Bjarni J Vilhjálmsson, Hilary K Finucane, Rany M Salem, Daniel I Chasman, Paul M Ridker, Benjamin M Neale, Bonnie Berger, Nick Patterson, and Alkes L Price (2015b). Efficient Bayesian mixed-model analysis increases association power in large cohorts. *Nature Genetics* 47.3, pp. 284–290.

Ma, Shiqian, Lingzhou Xue, and Hui Zou (2013). Alternating Direction Methods for Latent Variable Gaussian Graphical Model Selection. *Neural computation* 25.8, pp. 2172–2198.

Mackay, Ian J, Pauline Bansept-Basler, Toby Barber, Alison R Bentley, James Cockram, Nick Gosman, Andy J Greenland, Richard Horsnell, Rhian Howells, Donal M O'Sullivan, Gemma A Rose, and Phil J Howell (2014). An eight-

parent multiparent advanced generation inter-cross population for winter-sown wheat: creation, properties, and validation. *G3: Genes—Genomes—Genetics* 4.9, pp. 1603–1610.

Mairal, Julien, Francis Bach, Jean Ponce, and Guillermo Sapiro (2010). Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research* 11.Jan, pp. 19–60.

Mandò, C, S Tabano, P Colapietro, P Pileri, F Colleoni, L Avagliano, P Doi, G Bulfamante, M Miozzo, and I Cetin (2011). Transferrin receptor gene and protein expression and localization in human IUGR and normal term placentas. *Placenta* 32.1, pp. 44–50.

Marchini, Jonathan and Bryan Howie (2010). Genotype imputation for genome-wide association studies. *Nature Reviews Genetics* 11.7, pp. 499–511.

Marchini, Jonathan, Bryan Howie, Simon Myers, Gil McVean, and Peter Donnelly (2007). A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics* 39.7, pp. 906–913.

Marttinen, Pekka, Matti Pirinen, Antti-Pekka Sarin, Jussi Gillberg, Johannes Kettunen, Ida Surakka, Antti J Kangas, Pasi Soininen, Paul O'Reilly, Marika Kaakinen, Mika Kähönen, Terho Lehtimäki, Mika Ala-Korpela, Olli T Raitakari, Veikko Salomaa, Marjo-Riitta Jarvelin, Samuli Ripatti, and Samuel Kaski (2014). Assessing multivariate gene-metabolome associations with rare variants using Bayesian reduced rank regression. *Bioinformatics* 30.14, btu140–2034.

Marx, Vivien (2015). Human phenotyping on a population scale. *Nature Methods* 12.8, pp. 711–714.

Mathieson, Iain and Gil McVean (2012). Differential confounding of rare and common variants in spatially structured populations. *Nature Genetics* 44.3, pp. 243–246.

Mathieson, Iain and Gil McVean (2013). Reply to: "FaST-LMM-Select for addressing confounding from spatial structure and rare variants". *Nature Genetics* 45.5, pp. 471–471.

Mazumder, Rahul and Trevor Hastie (2012). The graphical lasso: New insights and alternatives. *Electronic Journal of Statistics* 6.0, pp. 2125–2149.

Mazumder, Rahul, Trevor Hastie, and Robert Tibshirani (2010). Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *Journal of Machine Learning Research* 11.Aug, pp. 2287–2322.

Meuwissen, Theo H E, Ben J Hayes, and Michael E Goddard (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics* 157.4, pp. 1819–1829.

Meuwissen, Theo HE, Jorgen Odegard, Ina Andersen-Ranberg, and Eli Grindflek (2014). On the distance of genetic relationships and the accuracy of genomic prediction in pig breeding. *Genetics Selection Evolution* 46.1, p. 1.

Meyer, Karin and Mark Kirkpatrick (2010). Better Estimates of Genetic Covariance Matrices by "Bending" Using Penalized Maximum Likelihood. *Genetics* 185.3, pp. 1097–1110.

Mott, Richar, Christopher J Talbot, Maria G Turri, Allan C Collins, and Jonathan Flint (2000). A method for fine mapping quantitative trait loci in outbred animal stocks. *Proceedings of the National Academy of Sciences* 97.23, pp. 12649–12654.

Mu, Cun, Yuqian Zhang, John Wright, and Donald Goldfarb (2014). Scalable Robust Matrix Recovery - Frank-Wolfe Meets Proximal Methods. *CoRR*.

Muñoz, Patricio R, Marcio F R Resende, Salvador A Gezan, Marcos Deon Vilela Resende, Gustavo de los Campos, Matias Kirst, Dudley Huber, and Gary F Peter (2014). Unraveling additive from nonadditive effects using genomic relationship matrices. *Genetics* 198.4, pp. 1759–1768.

Nakajima, Shinchi, R Tomioka, and Masashi Sugiyama (2015). Condition for perfect dimensionality recovery by variational bayesian PCA. *Journal of Machine Learning Research*.

Nakajima, Shinichi and Masashi Sugiyama (2011). Theoretical Analysis of Bayesian Matrix Factorization. *The Journal of Machine Learning Research* 12, pp. 1185–1224.

Nakajima, Shinichi, Ryota Tomioka, Masashi Sugiyama, and S Derin Babacan (2012). Perfect Dimensionality Recovery by Variational Bayesian PCA. *NIPS*.

Nakajima, Shinichi, Masashi Sugiyama, S Derin Babacan, and Ryota Tomioka (2013). Global Analytic Solution of Fully-observed Variational Bayesian Matrix Factorization. *Journal of Machine Learning Research* 14.Jan, pp. 1–37.

O'Reilly, Paul F, Clive J Hoggart, Yotsawat Pomyen, Federico C F Calboli, Paul Elliott, Marjo-Riitta Jarvelin, and Lachlan J M Coin (2012). MultiPhen: Joint Model of Multiple Phenotypes Can Increase Discovery in GWAS. *PLoS ONE* 7.5, e34861–12.

Park, Sung Hee, Ji Young Lee, and Sangsoo Kim (2011). A methodology for multivariate phenotype-based genome-wide association studies to mine pleiotropic genes. *BMC Systems Biology* 5.2, p. 1.

Patterson, Nick, Alkes L Price, and David Reich (2006). Population Structure and Eigenanalysis. *PLoS Genetics* 2.12, e190–20.

Piccolo, Stephen R, Ryan P Abo, Kristina Allen-Brady, Nicola J Camp, Stacey Knight, Jeffrey L Anderson, and Benjamin D Horne (2009). Evaluation of genetic risk scores for lipid levels using genome-wide markers in the Framingham Heart Study. *BMC Proceedings* 3.Suppl 7, S46.

Pickrell, Joseph K, Tomaz Berisa, Jimmy Z Liu, Laure Segurel, Joyce Y Tung, and David A Hinds (2016). Detection and interpretation of shared genetic influences on 42 human traits. *Nature Genetics* 48.7, pp. 709–717.

Pirinen, Matti, Peter Donnelly, and Chris C A Spencer (2013). Efficient computation with a linear mixed model on large-scale data sets with applications to genetic studies. *The Annals of Applied Statistics* 7.1, pp. 369–390.

Price, Alkes L, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics* 38.8, pp. 904–909.

Pritchard, Jonathan K, Matthew Stephens, and Peter Donnelly (2000). Inference of Population Structure Using Multilocus Genotype Data. *Genetics* 155.2, pp. 945–959.

Pritchard, Jonathan K, Matthew Stephens, Noah A Rosenberg, and Peter Donnelly (2000). Association Mapping in Structured Populations. *The American Journal of Human Genetics* 67.1, pp. 170–181.

Rakitsch, Barbara, Christoph Lippert, Karsten M Borgwardt, and Oliver Stegle (2013). It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. *NIPS*, pp. 1466–1474.

Rangkasenee, Noppawan, Eduard Murani, Ronald M Brunner, Karl Schellander, Mehmet Ulas Cinar, Henning Luther, Andreas Hofer, Monika Stoll, Anika Witten, Siriluck Ponsuksili, and Klaus Wimmers (2013). Genome-Wide Association Identifies TBX5 as Candidate Gene for Osteochondrosis Providing a Functional Link to Cartilage Perfusion as Initial Factor. *Frontiers in genetics* 4.

Recht, Benjamin (2011). A Simpler Approach to Matrix Completion. *The Journal of Machine Learning Research* 12, pp. 3413–3430.

Recht, Benjamin, Weiyu Xu, and Babak Hassibi (2008). Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. *2008 47th IEEE Conference on Decision and Control*, pp. 3065–3070.

Rubin, Donald B (1996). Multiple Imputation After 18 years. *Journal of the American Statistical Association* 91.434, pp. 473–489.

Rudi, Alessandro, Raffaello Camoriano, and Lorenzo Rosasco (2015). Less is more: Nyström computational regularization. *Advances in Neural Information Processing Systems*.

Runcie, Daniel E and Sayan Mukherjee (2013). Dissecting High-Dimensional Phenotypes with Bayesian Sparse Factor Analysis of Genetic Covariance Matrices. *Genetics* 194.3, pp. 753–767.

Sabatti, Chiara, Susan K Service, Anna-Liisa Hartikainen, Anneli Pouta, Samuli Ripatti, Jae Brodsky, Chris G Jones, Noah A Zaitlen, Teppo Varilo, Marika Kaakinen, Ulla Sovio, Aimo Ruokonen, Jaana Laitinen, Eveliina Jakkula, Lachlan Coin, Clive Hoggart, Andrew Collins, Hannu Turunen, Stacey Gabriel, Paul Elliot, Mark I McCarthy, Mark J Daly, Marjo-Riitta Jarvelin, Nelson B Freimer, and Leena Peltonen (2008). Genome-wide association analysis of

metabolic traits in a birth cohort from a founder population. *Nature Genetics* 41.1, pp. 35–46.

Salakhutdinov, Ruslan and Andriy Mnih (2008). *Bayesian probabilistic matrix factorization using Markov chain Monte Carlo*. New York, New York, USA: ACM.

Scutari, Marco, Phil Howell, David J Balding, and Ian Mackay (2014). Multiple Quantitative Trait Analysis Using Bayesian Networks. *Genetics* 198.1, pp. 129–137.

Segura, Vincent, Bjarni J Vilhjálmsson, Alexander Platt, Arthur Korte, Ümit Seren, Quan Long, and Magnus Nordborg (2012). An efficient multi-locus mixed-model approach for genome-wide association studies in structured populations. *Nature Genetics* 44.7, pp. 825–830.

Sequencing, Rat Genome and Mapping Consortium (2013). Combined sequence-based and genetic mapping analysis of complex traits in outbred rats. *Nature Genetics* 45.7, pp. 767–775.

Shimizu, Shohei, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, and Kenneth Bollen (2011). DirectLiNGAM - A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model. *Journal of Machine Learning Research*.

Shin, So-Youn, Eric B Fauman, Ann-Kristin Petersen, Jan Krumsiek, Rita Santos, Jie Huang, Matthias Arnold, Idil Erte, Vincenzo Forgetta, Tsun-Po Yang, et al. (2014). An atlas of genetic influences on human blood metabolites. *Nature Genetics* 46.6, pp. 543–550.

Sivakumaran, Shanya, Felix Agakov, Evropi Theodoratou, James G Prendergast, Lina Zgaga, Teri Manolio, Igor Rudan, Paul McKeigue, James F Wilson, and Harry Campbell (2011). Abundant Pleiotropy in Human Complex Diseases and Traits. *The American Journal of Human Genetics* 89.5, pp. 607–618.

Sluis, Sophie van der, Danielle Posthuma, and Conor V Dolan (2013). TATES: efficient multivariate genotype-phenotype analysis for genome-wide association studies. *PLoS Genetics* 9.1, e1003235.

Smola, Alex J and Peter Bartlett (2001). Sparse greedy Gaussian process regression. *Advances in Neural Information Processing Systems*.

Solovieff, Nadia, Chris Cotsapas, Phil H Lee, Shaun M Purcell, and Jordan W Smoller (2013). Pleiotropy in complex traits: challenges and strategies. *Nature Reviews Genetics* 14.7, pp. 483–495.

Soranzo, Nicole, Tim D Spector, Massimo Mangino, Brigitte Kühnel, Augusto Rendon, Alexander Teumer, Christina Willenborg, Benjamin Wright, Li Chen, Mingyao Li, et al. (2009). A genome-wide meta-analysis identifies 22 loci associated with eight hematological parameters in the HaemGen consortium. *Nature Genetics* 41.11, pp. 1182–1190.

Speed, Doug and David J Balding (2014). MultiBLUP: improved SNP-based prediction for complex traits. *Genome Research* 24.9, pp. 1550–1557.

Speed, Doug, Gibran Hemani, Michael R Johnson, and David J Balding (2012). Improved Heritability Estimation from Genome-wide SNPs. *The American Journal of Human Genetics* 91.6, pp. 1011–1021.

Srebro, Nathan (2004). "Learning with matrix factorizations". PhD thesis. Massachusetts Institute of Technology: Massachusetts Institute of Technology.

Stegle, Oliver, Leopold Parts, Richard Durbin, and John Winn (2010). 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.

Stegle, Oliver, Christoph Lippert, Joris M Mooij, Neil D Lawrence, and Karsten M Borgwardt (2011). Efficient inference in matrix-variate Gaussian models with iid observation noise. *NIPS*.

Steinsaltz, David, Andrew Dahl, and Kenneth W Wachter (2016). Statistical Properties of Simple Random-Effects Models for Genetic Heritability. *In Preparation*.

Stephens, Matthew (2013). A unified framework for association analysis with multiple related phenotypes. *PLoS ONE* 8.7, e65245.

Suhre, Karsten, So-Youn Shin, Ann-Kristin Petersen, Robert P Mohney, David Meredith, Brigitte Wägele, Elisabeth Altmaier, CARDIoGRAM, Panos Deloukas, Jeanette Erdmann, et al. (2011). Human metabolic individuality in biomedical and pharmaceutical research. *Nature* 477.7362, pp. 54–60.

Sul, Jae Hoon and Eleazar Eskin (2013). Mixed models can correct for population structure for genomic regions under selection. *Nature Reviews Genetics* 14.4, pp. 300–300.

Svishcheva, Gulnara R, Tatiana I Axenovich, Nadezhda M Belonogova, Cornelia M van Duijn, and Yurii S Aulchenko (2012). Rapid variance components-based method for whole-genome association analysis. *Nature Genetics* 44.10, pp. 1166–1170.

Tarleton, R L, B H Koller, A Latour, and M Postan (1992). Susceptibility of beta 2-microglobulin-deficient mice to Trypanosoma cruzi infection. *Nature* 356.6367, pp. 338–340.

Tibshirani, Robert (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 1403, pp. 54–62.

Troyanskaya, Olga, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics* 17.6, pp. 520–525.

Tucker, George, Alkes L Price, and Bonnie Berger (2014). Improving the power of GWAS and avoiding confounding from population stratification with PC-Select. *Genetics* 197.3, pp. 1045–1049.

Tucker, George, Po-Ru Loh, Iona M MacLeod, Ben J Hayes, Michael E Goddard, Bonnie Berger, and Alkes L Price (2015). Two-Variance-Component Model Improves Genetic Prediction in Family Datasets. *The American Journal of Human Genetics* 97.5, pp. 677–690.

Van Essen, David C, Stephen M Smith, Deanna M Barch, Timothy E J Behrens, Essa Yacoub, and Kamil Ugurbil (2013). The WU-Minn Human Connectome Project: An overview. *NeuroImage* 80, pp. 62–79.

Visscher, Peter M and Michael E Goddard (2015). A general unified framework to assess the sampling variance of heritability estimates using pedigree or marker-based relationships. *Genetics* 199.1, pp. 223–232.

Visscher, Peter M and Jian Yang (2016). A plethora of pleiotropy across complex traits. *Nature Genetics* 48.7, pp. 707–708.

Visscher, Peter M, Sarah E Medland, Manuel A R Ferreira, Katherine I Morley, Gu Zhu, Belinda K Cornes, Grant W Montgomery, and Nicholas G Martin (2006). Assumption-Free Estimation of Heritability from Genome-Wide Identity-by-Descent Sharing between Full Siblings. *PLoS Genetics* 2.3, e41.

Visscher, Peter M, Gibran Hemani, Anna A E Vinkhuyzen, Guo-Bo Chen, Sang Hong Lee, Naomi R Wray, Michael E Goddard, and Jian Yang (2014). Statistical power to detect genetic (co)variance of complex traits using SNP data in unrelated samples. *PLoS Genetics* 10.4, e1004269.

Wakefield, J (2009). Bayes factors for genome-wide association studies: comparison with P-values. *Genetic Epidemiology*.

Weir, Bruce S, Amy D Anderson, and Amanda B Hepler (2006). Genetic relatedness analysis: modern data and new challenges. *Nature Reviews Genetics* 7.10, pp. 771–780.

Widmer, Christian, Christoph Lippert, Omer Weissbrod, Nicolo Fusi, Carl Kadie, Robert Davidson, Jennifer Listgarten, and David Heckerman (2014). Further improvements to linear mixed models for genome-wide association studies. *Scientific reports* 4, p. 6874.

Williams, Christopher K I and Matthias W Seeger (2000). Using the Nyström Method to Speed Up Kernel Machines. *NIPS*.

Wray, Naomi R, Jian Yang, Ben J Hayes, Alkes L Price, Michael E Goddard, and Peter M Visscher (2013). Pitfalls of predicting complex traits from SNPs. *Nature Reviews Genetics* 14.7, pp. 507–515.

Wright, Stephen J (1999). Modified Cholesky Factorizations in Interior-Point Algorithms for Linear Programming. *SIAM Journal on Optimization* 9.4, pp. 1159–1191.

Yamamoto, Guilherme L, Wagner A R Baratela, Tatiana F Almeida, Monize Lazar, Clara L Afonso, Maria K Oyamada, Lisa Suzuki, Luiz A N Oliveira, Ester S Ramos, Chong A Kim, Maria Rita Passos-Bueno, and Débora R Bertola (2014). Mutations in PCYT1A Cause Spondylometaphyseal Dysplasia with Cone-Rod Dystrophy. *The American Journal of Human Genetics* 94.1, pp. 113–119.

Yang, Jian, Beben Benyamin, Brian P McEvoy, Scott Gordon, Anjali K Henders, Dale R Nyholt, Pamela A Madden, Andrew C Heath, Nicholas G Martin, Grant W Montgomery, Michael E Goddard, and Peter M Visscher (2010a). Common SNPs explain a large proportion of the heritability for human height. *Nature Genetics* 42.7, pp. 565–569.

Yang, Jian, Sang Hong Lee, Michael E Goddard, and Peter M Visscher (2011). GCTA: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*.

Yang, Jian, Taeheon Lee, Jaemin Kim, Myeong-Chan Cho, Bok-Ghee Han, Jong-Young Lee, Hyun-Jeong Lee, Seoae Cho, and Heebal Kim (2013). Ubiquitous polygenicity of human complex traits: genome-wide analysis of 49 traits in Koreans. *PLoS Genetics* 9.3, e1003355.

Yang, Jian, Noah A Zaitlen, Michael E Goddard, Peter M Visscher, and Alkes L Price (2014). Advantages and pitfalls in the application of mixed-model association methods. *Nature Genetics* 46.2, pp. 100–106.

Yang, Jian, Sang Hong Lee, Naomi R Wray, Michael E Goddard, and Peter M Visscher (2016). Commentary on "Limitations of GCTA as a solution to the missing heritability problem". *BioRxiv*, p. 036574.

Yang, Qiong, Hongsheng Wu, Chao-Yu Guo, and Caroline S Fox (2010b). Analyze multivariate phenotypes in genetic association studies by combining univariate association tests. *Genetic Epidemiology* 34.5, pp. 444–454.

Yu, Jianming, Gael Pressoir, 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, and Edward S Buckler (2006). A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature Genetics* 38.2, pp. 203–208.

Yuan, Ming and Yi Lin (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika* 94.1, pp. 19–35.

Zaitlen, Noah and Peter Kraft (2012). Heritability in the genome-wide association era. *Human Genetics* 131.10, pp. 1655–1664.

Zaitlen, Noah, Peter Kraft, Nick Patterson, Bogdan Pasaniuc, Gaurav Bhatia, Samuela Pollack, and Alkes L Price (2013). Using extended genealogy to estimate components of heritability for 23 quantitative and dichotomous traits. *PLoS Genetics* 9.5, e1003520.

Zeileis, Achim, Kurt Hornik, Alex Smola, and Alexandros Karatzoglou (2004). kernlab - An S4 Package for Kernel Methods in R. *Journal of Statistical Software*.

Zhang, Zhiwu, Elhan Ersoz, Chao-Qiang Lai, Rory J Todhunter, Hemant K Tiwari, Michael A Gore, Peter J Bradbury, Jianming Yu, Donna K Arnett, Jose M Ordovas, and Edward S Buckler (2010). Mixed linear model approach adapted for genome-wide association studies. *Nature Genetics* 42.4, pp. 355–360.

Zhao, Keyan, María José Aranzana, Sung Kim, Clare Lister, Chikako Shindo, Chunlao Tang, Christopher Toomajian, Honggang Zheng, Caroline Dean, Paul Marjoram, and Magnus Nordborg (2007). An Arabidopsis Example of Association Mapping in Structured Samples. *PLoS Genetics* 3.1, e4.

Zhou, Xiang, Peter Carbonetto, and Matthew Stephens (2013). Polygenic Modeling with Bayesian Sparse Linear Mixed Models. *PLoS Genetics* 9.2, e1003264.

Zhou, Xiang and Matthew Stephens (2012). Genome-wide efficient mixed-model analysis for association studies. *Nature Genetics* 44.7, pp. 821–824.

Zhou, Xiang and Matthew Stephens (2014). Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nature Methods* 11.4, pp. 407–409.

Zhu, Xiang and Matthew Stephens (2016). Bayesian large-scale multiple regression with summary statistics from genome-wide association studies. *BioRxiv*.

Zhu, Zhihong, Andrew Bakshi, Anna A E Vinkhuyzen, Gibran Hemani, Sang Hong Lee, Ilja M Nolte, Jana V Van Vliet-Ostaptchouk, Harold Snieder, LifeLines Cohort Study, Tõnu Esko, Lili Milani, Reedik Mägi, Andres Metspalu, William G Hill, Bruce S Weir, Michael E Goddard, Peter M Visscher, and Jian Yang (2015). Dominance genetic variation contributes little to the missing heritability for human complex traits. *American journal of human genetics* 96.3, pp. 377–385.

Zwiernik, Piotr, Caroline Uhler, and Donald Richards (2014). Maximum Likelihood Estimation for Linear Gaussian Covariance Models. *arXiv.org*. arXiv: 1408.5604.