Contents

Chapter 1

Extending linear mixed models to high-dimensional phenotypes

Part of this introduction might end up in the main intro and/or main intro parts in here; equally some terms like the standar linear model, RSS and such will be explained in the main intro already; will adapt once I have the main intro; also maybe the simulation bit should be its own small chapter, not sure. But irrespective of where things will end up, it's written down now

Many cohort studies today, ranging from studies in model organism such as yeast and arabidopsis thaliana to human, have rich, high dimensional datasets including molecular, morphological or imaging derived traits [**?????**]. However, these traits have often been analysed separately, partly for simplicity and partly because of a paucity of models suitable for the anlaysis of high-dimensional phenotype data. For few traits, a variety of multi-trait models have been developed which can be broadly grouped into three different classes: i) dimensionality reduction techniques, ii) meta-analysis approaches and iii) multivariate regression models (reviewed in [**??**]).

Dimensionality reduction techniques Dimensionality reduction methods in genotype-phenotype mapping seek to find a linear combination of the phenotypes $\mathbf{Y} \in \mathcal{R}^{N,P}$ into a lower dimensional space $\tilde{\mathbf{Y}} \in \mathcal{R}^{N,K}$:

$$\tilde{\mathbf{Y}} = b_1\mathbf{y}_1 + b_2\mathbf{y}_2 + \cdots + b_P\mathbf{y}_P \tag{1.1}$$

Two commonly employed dimensionality reduction methods are principal component analysis (PCA) and canonical correlation analyses (CCA). In PCA, the components of the new phenotype representation are called principal components (PC) and are the eigenvectors $\mathbf{W}$ of the empirical covariance matrix $\mathbf{Y}^T\mathbf{Y}$: $\mathbf{Y}^T\mathbf{Y} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T$. The eigenvalues Λ corresponding to the PCs are equivalent to the variance explained by their components. The transformation of the phenotype data into principal components leads to a projection where the highest amount of phenotypic variance explained lies in the first component, the second highest variance in the second component and so forth. The dimensionality reduction is achieved by using the first K principal components until the cumulative sum of the eigenvalues reaches a predefined threshold of total phenotypic variance that should be retained. These K principal components are used as proxy phenotypes for the association study. PCA as a dimensionality reduction technique has for instance been used in studies to find links between genotypes and facial features or obesity phenotypes [**???**]. Recently, Aschard and colleagues **??** demonstrated that simply focusing on the principal components with the highest variance might not exploit the full potential of using PCA for genetic association. They propose a model of combined PCA where the PCs are grouped based on the level of variance they explain. They show a power gain in detecting genetic associations compared to simple approaches of only testing the top few PCs.

While the PCA dimensionality reduction approach focus on the phenotype space and subsequent association with the genotypes, CCA finds the optimal transformation of $\mathbf{Y}$ into $\tilde{\mathbf{Y}}$ while simultaneously testing for the association with the genotypes. Originally proposed by Hotelling for any set of variables that remain invariant under internal linear transformation**?**, in quantitative genetics CCA seeks to maximise the canonical correlation $\hat{\rho}$ between $\tilde{\mathbf{Y}} = A^T\mathbf{Y}$ and a genotype $\mathbf{X} \in \mathcal{R}^{N,1}$: $\hat{\rho} = cor(\mathbf{X}, \tilde{\mathbf{Y}}) = \Sigma_{XY}A(A^T\Sigma_{YY}A\Sigma_{XX})^{-\frac{1}{2}}$. Σ_{YY}, Σ_{XX} and Σ_{XY} are the empirical sample covariance matrices of the phenotypes, the genotypes and cross-covariance of the phenotypes and genotypes, respectively. $\hat{\rho}$ is maximised by finding the squared root of the largest eigenvalue of the $\Sigma_{YY}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$ covariance matrix and the corresponding eigenvector $\mathbf{A}$ contains the coefficients for constructing $\tilde{\mathbf{Y}}$ [**?**]. CCA finds the transformation $\tilde{\mathbf{Y}}$ that explains the maximum amount of the covariation between the genotype and all traits. The significance of the correlation i.e. the genotype-phenotype association can be tested via Bartlett's likelihood ratio test with the null hypothesis of $\mathrm{H}_{0:}$ $\Sigma_{YX} = 0$ [**?**]. Ferreira and Purcell showed in simulations that CCA with multiple traits and one genetic marker $\mathbf{X}$ controls well for type I errors and has increased power compared to multivariate tests. In order to extend the CCA method to more than one marker, the genotypes also undergo a linear transformation: $\hat{\rho} = cor(B^T\mathbf{X}, A^T\mathbf{Y}) = B^T\Sigma_{XY}A(A^T\Sigma_{YY}AB^T\Sigma_{XX}B)^{-\frac{1}{2}}$ and the maximum $\hat{\rho}$ is found by solving for the largest eigenvalue of both $\mathbf{A}$ and $\mathbf{B}$. As the number of genotype markers in GWAS exceeds the number of samples, estimates of the covariance matrix X^TX become unreliable [**?**]. Several methods have been developed to circumvent this issue, making use of sparse matrices [**?**] or a priori grouping of the genotypes [**?**].

Meta-analysis approaches Meta-analysis approaches combine the simplicity of the univariate approaches with the advantages of the multivariate approach. For each phenotype, a univariate association study is conducted and the summary stastics of these tests combined. Many methods for combining the summary statistics [**????**] go back to the work by O'Brien [?], who proposed to use a linear combination of the observed test statistics for each univariate test $\mathbf{T} = (T_1, \ldots, T_P)^T$ as the new statistics to be evaluated for significance. $\mathbf{T}$ is asymptoctically normal distributed with mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P)^T$ and covariance matrix $\boldsymbol{\Sigma}$. O'Brien stastistic allows for testing the Null hypothesis $H_0 : \mu = 0$ against the alternative hypothesis of $H_1 : \mu_p \geq 0, p = 1, \ldots, P$ and is most powerful if $\mu_1 = \ldots = \mu_P$ [?]. It is defined as: $S = \mathbf{J}^T \boldsymbol{\Sigma} \mathbf{T}$, with $\mathbf{J} = (1, 1, \ldots, 1)^T$. The statistic has been modified in a number of studies, by adapting either the weighting matrix $\mathbf{J}$, the covariance matrix $\boldsymbol{\Sigma}$ or both. Xu and colleagues **?** optimised S to allow for testing against a general μ rather then for a case where $\mu_1 = \ldots = \mu_P$ by allowing for flexible, but restrained weights in $\mathbf{J}$. Similarily, Yang and colleagues **?** proposed non-uniform weights to reflect heterogeneity in the means and use a sample splitting and cross-validation approach to determine the optimal weights. While the previous two studies showed an increase in power for using the combined statistic, they either used a small marker set or small number of phenotypic traits. Bolormaa and colleagues showed that these power gains also hold for genotype to phenotype mapping of 32 traits across all genome-wide markers [?]. In their study, the weights of O'Briens proposal are substituted by the signed t-statistic.

Regression models There are a number of different regression models that allow for the multivariate analysis of phenotypes. Among them are graphical models, generalized estimation equations and frailty models, for which a summary of methods and application can be found in [**??**]. Here, I will focus on describing the development of multivariate linear regression models for genotype-phenotype mapping. The underlying models and their derivation have been described in Section **??**. Before the era of GWAS, QTL mapping in linkage experiments have demonstrated the increase in power when jointly analysing traits with common underlying genetics. Jiang and colleagues [?] proposed a multi-trait model where the phenotypes are jointly modeled as the sum of the fixed genetic effects of interest, fixed effects for genetic background variation and residual noise. They show that the joint analysis of traits can increase power to detect the underlying genetics and can increase the precision of the parameter estimates. The significance of the association is determined via a likelihood ratio test of the parameter estimates under the Null model where the fixed genetic effect is zero and the parameter estimates under the alternative model. The alternative model design depends on the underlying biological hypothesis regarding the effect of the genetic variant. Here, Jiang and colleagues differentiate hyptheses for a simple joint mapping of phenotypes, pleiotrophy and gene-environment interactions. Joint mapping does not make any assumptions about the underlying genetic architecture and simply tests if an association can be found when both traits are analysed jointly, i.e. the effect of the genetic variant is non-zero for at least one of the traits. This hypothesis can be extended in requiring that the effect on both traits is unequal to zero. In this case, the genetic variant is considered to be pleiotrophic. To test for gene-environment interaction, the different conditions a trait was studied in can be treated as different traits and be jointly mapped. If the effect size estimates of the genetic variant are not equal, the variant is considered to have environmental interactions. Methods developed thereafter often use the same underlying hypotheses for the mapping, but different techniques for the evaluation of the significance. For instance, two other groups developed methods for the joint analysis of traits based specifically on the residual sum of squares (RSS) matrix of the standard linear model (Equation **??**) estimated at each locus tested [**??**]. In the model proposed by Knott and Haley, the different properties and descriptors of the RSS are used to determine the significance of the QTL mapping. To test for pleiotropy for instance, the determinent of the RSS at the test locus is compared to the RSS of the null model of no association. In contrast, Korol and colleagues propose to use the RSS of the multi-trait mapping as a means for trait transformation and dimensionality reduction. The resulting one-dimensional trait per sample is fitted in a single-trait test for significance testing. While methods described so far have only used fixed genetic effects, Korte and colleagues **?** were the first to introduce a random genetic effect into the model. Based on the original model by Jiang, they substituted the fixed effect accounting for background genetics by a random effect, turning the multivariate linear model into a mulitvariate linear mixed model (Equation **??**). Based on these principals and method development for the efficient analysis of large cohort sizes, a number of publically available frameworks for the genome-wide mapping of a moderate number of traits via multivariate linear mixed models were developed. [**?????**].

Out of the differerent approaches described above, multivariate linear mixed models (LMMs) have the additional advantage that they can control for population structure (Section **??**). In LMMs, both the residual noise and the genetic backgound are modeled as random effects. For the random genetic effect, a complex covariance structure can be modeled such that family structure and relatedness captured in the genotypes can be exploited to model background genetic correlations in phenotype[**??**] (Equation **??**). The random effects are assumed to be composed of a sample-to-sample and trait-to-trait covariance component. The genetic sample covariance can be obtained from the data itself, e.g. using a genetic relationship matrix (GRM) or identity by descent, whereas the trait covariance terms need to be estimated from the observed data. Since the introduction of LMM for multi-trait GWAS studies, LMMs have been extended to handle the large sample sizes obtained in particular in human studies, allowing for cohort sizes of xxx individuals [**?**]. However nearly all current methods scale poorly in the number of traits (P) due to a bottle neck caused by estimating the trait covariance terms.

Together with Francesco Paolo Casale, a PhD student in Oliver Stegle's research group at the EBI, I developed a simple, but surprisingly effective heuristic which uses a linear mixed model with bootstrapping (LiMMBo) that allows for the analysis of datasets with a large number of phenotypic traits. In order to validate LiMMBo and show its applicability to high-dimensional phenotypes, I developed a phenotype simulation framework which allows for the simulation of well-defined phenotypes with different underlying genetic structures and noise distributions. This chapter describes this simulation strategy and the LiMMBo approach and its validity.

1.1 Data simulation

Often, a first step in new method development is to reverse the task of genotype to phenotype mapping. The latter is commonly realised by fitting a linear model to the phenotype measurements and treating the genotype and possible covariates as explanatory variables. In order to evaluate new methods, one needs to have a set of well-characterised genotypes and phenotypes to know the ground truth based on which comparisons of the model performance can be made. Based on the underlying linear association methods, the phenotypes are often simulated as a linear composition of different effect components.

For the methods development and selection process in this thesis (Section 1.2 and **??**), I needed well-characterised phenotypes for evaluation before applying the methods to real datasets and biological questions. In the following section, I will first describe the simulation of genotypes with different levels of population structure and relatedness, followed by a description of the phenotype simulation. The simulation strategies described in this section apply to all simulated datasets within this thesis.

1.1.1 Genotypes

Genotypes can either be generated by sampling from real data sets or by simulating SNPs anew. In the latter case, assuming bi-allelic SNPs, each SNP is simulated from a binomial distribution with two trials and probability equal to the given allele frequencies This simple approach, however, does not account for any LD structure in the genome. In contrast, sampling from a diverse set of real genotypes does not only facilitate retaining realistic LD structure in the genotypes, it also allows for the simple simulation of more defined population structure and relatedness within a cohort. As I wanted to evaluate the methods described in this thesis across different levels of relatedness and population structure, I chose to simulate the genotypes based on real genotype data from four European ancestry populations of the 1000 Genomes (1KG) Project (populations: CEU, FIN, GBR, TSI) [**?**], similar to strategies described in [**??**]. I simulated three genotype sets, each with 1,000 samples, that differed i) in the number of ancestors N from which the genotypes were chosen and ii) the subpopulations the ancestors were chosen from:

1. unrelatedPopStructure: unrelated individuals with prior assigment of ancestral population ($N = 10$)

2. unrelatedNoPopStructure: unrelated individuals with mixed ancestral population ($N = 10$)

3. relatedNoPopStructure: related individuals with mixed ancestral population ($N = 2$)

The number of ancestors sets the the level of relatedness within a cohort, where low numbers of N introduce relatedness among individuals, while high numbers of N lead to low levels of structure and relatedness. The choice of ancestrial population determines the level of subpopulation formation in the simulated genotypes: allowing for random selection of ancestors independent of the four subpopulations in the 1KG datasets yields low levels of population structure, whereas choosing ancestors based on their subpopulation gives rise to subpopulations in the simulated dataset as well. For the simulation, each newly synthesised individual is assigned to N ancestors from the original 1KG Project and their genome split into blocks of 1,000 SNPs. For each SNP block, an ancestor is chosen at random either from the whole dataset (NoPopstructure) or a subpopolution (Popstructure) and its genotype is copied to the individuals genome.

The level of structure and relatedness introduced by this simulation strategy can be visualised by examining the GRM and the principal components of the genotypes. The kinship as estimated via Equation **??** is a measure of relatedness between the individuals, while principal components reflect the genotypic variance in the data. The hierarchical clustering of the genetic relationship estimates and scatter plots of the first two principal components for each genotype set are shown in Figure 1.1. Samples cluster tightly based on their ancestrial subpopulations (Figure 1.1A), while there is no clustering and an even spread in the PC plot for the cohort of unrelated individuals with ancestors sampled across all subpopulations (Figure 1.1B). The cohort of related individuals shows less spread in the second principal component and higher individual genetic relationship estimates (Figure 1.1C).

1.1.2 Phenotypes

The simulation of phenotypes and their components can be described on two levels, either based on the biological meaning they reflect or the statistical type of the effect. In statistics, one distinguishes the fixed effects which are constant across individuals, from random effect which can vary (discussed in detail in [**?**]). On the biological level, we can classify the phenotypic components into genetic and non-genetic (noise) components. Commonly simulated phenotype components are fixed and random genetic effects and fixed, correlated and random noise effects (e.g.[**????**]).

Genetic fixed effects are the effects of interest in genetic association studies i.e. the SNPs that are significantly associated with a phenotype. Genetic effects that are not

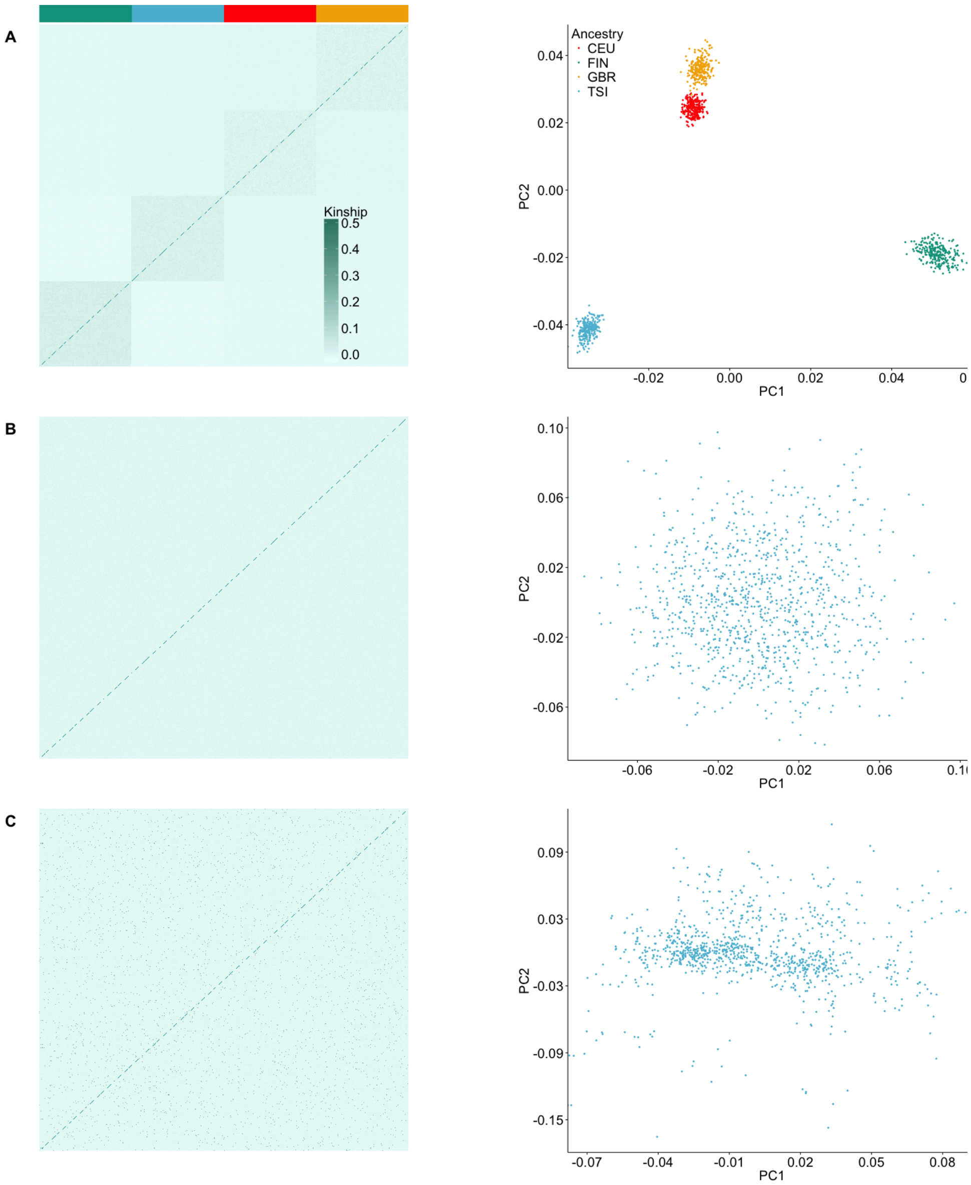


Figure 1.1: Genetic relationship matrices and principle components of three simulated European ancestry cohorts. The genotypes were simulated based in genotype data from four European ancestry populations (ancestry colour key in panel A). Depending on the choice and number of ancestors for the sampling of chromosomes to simulate an individual's genotype, cohorts with differing levels of population and relatedness structure will arise. The left column depicts the hierarchical clustering of the sample-to-sample genetic relationship coefficients (complete linkage clustering of euclidean distance between coefficients), the right column the first and second principal component (PC) of the sample genotypes for the three different cohorts: A. unrelated individuals, with population structure: $N = 10$, prior assigment to ancestral population; B. unrelated individuals, no population structure: $N = 10$, no prior assigment to ancestral population; C. related individuals, no population structure: $N = 2$, no prior assigment to ancestral population.

associated on a per-SNP basis but reflect underlying population structure and relatedness in a cohort are simulated as random genetic effects. These effects are based on genetic relationship estimates or IBD which can be derived from the samples' genotypes. Non-genetic effects are used to simulate environmental, experimental or noise effects. Fixed noise effects are use to simulate confounding variables or covariates in an analysis, such as sex, age, weight or disease status. When simulating such confounding structures, assumptions about their distribution have to be made and this choice depends on the specific biological effects that should be modeled. Common distribution are binomial (e.g. sex), normal or uniform distribution (e.g. weight, height) or categorical (e.g disease status). Random noise effects simulate any non-specified noise effects that could arise due to, for instance, experimental measurement error. Correlated noise effects are a type of random effect that can be used to simulate a phenotype component with a defined level of correlation between traits. For instance, such effects can reflect correlation structure decreasing in phenotypes with ordered or spatial components e.g. in imaging data.

In addition to the different sources of variation these components model, they can further differ in their effect distribution across the simulated traits and the proportion of the variance they explain out of the total phenotypic variance. The simulation strategy of these components, the effect distribution and the scaling of the components to a specifc proportion of variance are described below.

The phenotypes $\mathbf{Y} \in \mathcal{R}^{N,P}$ of N samples and P traits are generated as the sum of i) fixed genetic effects $\mathbf{U} \in \mathcal{R}^{N,P}$, ii) random genetic effects $\mathbf{G} \in \mathcal{R}^{N,P}$, iii) fixed noise effects $\mathbf{C} \in \mathcal{R}^{N,P}$, iv) random noise effects $\mathbf{\Psi} \in \mathcal{R}^{N,P}$ and v) correlated noise effects $\mathbf{T} \in \mathcal{R}^{N,P}$. For component i-iv, a certain percentage of their variance is shared across all traits (shared) and the remainder is independent (ind) across traits.

1. *Fixed genetic effects:* For the fixed genetic effects, S random SNPs for N samples are drawn from the simulated genotypes. From the S random SNPs, a proportion θ is selected to be causal across all traits. $\mathbf{U}^{\text{shared}} \in \mathcal{R}^{N,P}$ is simulated as the matrix product of this shared causal SNP matrix $\mathbf{X}^{\text{shared}} \in \mathcal{R}^{N,\theta\times S}$ and the shared effect size matrix $\mathbf{B}^{\text{shared}} \in \mathcal{R}^{\theta\times S,P}$. $\mathbf{B}^{\text{shared}}$ in turn is the matrix product of the two normally distributed vectors $b_s \in \mathcal{R}^{\theta\times S,1}$ and $b_p^T \in \mathcal{R}^{1,P}$. The remaining $(1-\theta)\times S$ SNPs are simulated to have an independent effect across a limited number of traits p^{ind}. To realise this structure, $\mathbf{B}^{\text{ind}} \in \mathcal{R}^{(1-\theta)\times S,P}$ is initialised with normally distributed entries. Subsequently, $1-p^{\text{ind}}$ traits are randomly selected and the row entries for $\mathbf{B}^{\text{ind}}$ at these traits set to zero. $\mathbf{U}^{\text{ind}} \in \mathcal{R}^{N,P}$ is the matrix product of $\mathbf{X}^{\text{ind}} \in \mathcal{R}^{N,(1-\theta)\times S}$ and $\mathbf{B}^{\text{ind}}$. The fixed genetic effect $\mathbf{U}$ is the sum of $\mathbf{U}^{\text{shared}}$ and $\mathbf{U}^{\text{ind}}$.

2. *Fixed noise effects:* The fixed noise effects $\mathbf{C}$ are based on K confounders $\mathbf{F} \in \mathcal{R}^{N,K}$, with a proportion γ being shared across all traits yielding the shared confounder matrix $\mathbf{F}^{\text{shared}} \in \mathcal{R}^{\gamma\times K,P}$. The proportion of $1-\gamma$ confounder that are independent make up the independent confounder matrix $\mathbf{F}^{\text{ind}} \in \mathcal{R}^{(1-\gamma)\times K,P}$. The distributions for each of the K confounders are independent and can be either normal, uniform,

binomial or categorical. The effect size matrices $\mathbf{A}^{\text{shared}} \in \mathcal{R}^{\gamma \times K,P}$ and $\mathbf{A}^{\text{ind}} \in \mathcal{R}^{(1-\gamma) \times K,P}$ were designed as described for the fixed genetic effects. The total fixed noise effect is then $\mathbf{C} = \mathbf{K}^{\text{shared}}\mathbf{A}^{\text{shared}} + \mathbf{K}^{\text{ind}}\mathbf{A}^{\text{ind}}$.

3. *Random genetic effects:* The random genetic effects $\mathbf{G} \in \mathcal{R}^{N,P}$ are modeled as a matrix-normally distributed random variable, defined by its mean $\mathbf{M} \in \mathcal{R}^{N,P}$, its column covariance $\mathbf{C} \in \mathcal{C}^{P,P}$ and its row covariance $\mathbf{D} \in \mathcal{R}^{N,N}$.

$$\mathbf{G} \sim \mathcal{MN}_{N,P}\left(\mathbf{M},\ \mathbf{D},\ \mathbf{C}\right) \tag{1.2}$$

The $N \times N$ genetic relationship matrix $\mathbf{R}$, estimated according to Equation **??** from the SNP genotypes (of the simulated samples) represents the row covariance $\mathbf{D}$. The structure of the trait-to-trait covariance $\mathbf{C}$ depends on the design of the covariance effect, which can be either shared or independent across traits. To construct $\mathbf{G}$ from shared and independent random genetic effects, assume a matrix-normally distributed random variable $\mathbf{Z}$ with $\mathbf{M} = 0$ and $\mathbf{D} = \mathbf{R}$:

$$\mathbf{Z} \sim \mathcal{MN}_{N,P}\left(\mathbf{0},\ \mathbf{R},\ \mathbf{C}\right) \tag{1.3}$$

$\mathbf{Z}$ can be expressed in terms of a multivariate normal distribution

$$\text{vec}(\mathbf{Z}) \sim \mathcal{N}_{N \times P}\left(\mathbf{0},\ \mathbf{C} \otimes \mathbf{R}\right). \tag{1.4}$$

With the cholesky decompositon of $\mathbf{K}$ and $\mathbf{C}$ into $\mathbf{E} = \mathbf{B}\mathbf{B}^T$ and $\mathbf{C} = \mathbf{A}\mathbf{A}^T$

$$\text{vec}(\mathbf{Z}) \sim \mathcal{N}_{N \times P}\left(\mathbf{0},\ \mathbf{A}\mathbf{A}^T \otimes \mathbf{B}\mathbf{B}^T\right), \tag{1.5}$$

which can be rearranged as with $\mathbf{I}$ as the identity matrix

$$\begin{gathered}\text{vec}(\mathbf{Z}) \sim \mathcal{N}_{N \times P}\left(\mathbf{0},\ (\mathbf{A} \otimes \mathbf{B})\mathbf{I}(\mathbf{A}^T \otimes \mathbf{B}^T)\right) \\ \text{vec}(\mathbf{Z}) \sim \mathcal{N}_{N \times P}\left(\mathbf{0},\ (\mathbf{A} \otimes \mathbf{B})\mathbf{I}(\mathbf{A} \otimes \mathbf{B})^T)\right).\end{gathered} \tag{1.6}$$

Using the property of a normally distributed random variable $\mathbf{Y}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

$$w\mathbf{Y} \sim \mathcal{N}\left(w\boldsymbol{\mu},\ w\boldsymbol{\Sigma}w^T\right), \tag{1.7}$$

we can let $\text{vec}(\mathbf{Z}) = (\mathbf{A} \otimes \mathbf{B})\text{vec}(\mathbf{Y})$ and $\mathbf{Y} \sim \mathcal{N}_{N \times P}\left(\mathbf{0},\ \mathbf{I}\right)$ such that

$$(\mathbf{A} \otimes \mathbf{B})\text{vec}(\mathbf{Y}) \sim \mathcal{N}_{N \times P}\left(\mathbf{0},\ (\mathbf{A} \otimes \mathbf{B})\mathbf{I}(\mathbf{A} \otimes \mathbf{B})^T\right) \tag{1.8}$$

Using [**?**]: Lemma 4.3.1, we get

$$(\mathbf{A} \otimes \mathbf{B})\text{vec}(\mathbf{Y}) = \text{vec}(\mathbf{B}\mathbf{Y}\mathbf{A}^T) \tag{1.9}$$

For the independent effect, $\mathbf{A}^{\text{ind}}$ is a diagonal matrix with normally distributed entries: $(\mathbf{A}^{\text{ind}})^T = \text{diag}(a_1, a_2, \ldots, a_P) \sim \mathcal{N}(0,\ 1)$, such that $\mathbf{G}^{\text{ind}} = \text{vec}(\mathbf{B}\mathbf{Y}(\mathbf{A}^{\text{ind}})^T)$. $\mathbf{A}^{\text{shared}}$ of the shared effect is a matrix of row rank one, with normally distributed entries in row 1 and zeros elsewhere: $a_{1,j} \sim \mathcal{N}(0,\ 1)$ and $a_{i\neq 1,j} = 0$ such that $\mathbf{G}^{\text{shared}} = \text{vec}(\mathbf{B}\mathbf{Y}(\mathbf{A}^{\text{shared}})^T)$. The total random genetic effect $\mathbf{G}$ is $\mathbf{G} = \mathbf{G}^{\text{shared}} + \mathbf{G}^{\text{ind}}$.

4. *Random noise effects:* The random noise effects $\mathbf{\Psi}$ are simulated as the sum of a shared and an independent random noise effect. The shared random effect $\mathbf{\Psi}^{\text{shared}}$ is simulated as $\text{vec}(\mathbf{\Psi}^{\text{shared}}) \sim \mathcal{N}(0,\ 1)$. The independent random effect $\mathbf{\Psi}^{\text{ind}}$ is simulated as the matrix product of two normally distributed vectors $\mathbf{a} \sim \mathcal{N}_{N\times 1}(0,\ 1)$ and $\mathbf{b} \sim \mathcal{N}_{P\times 1}(0,\ 1)$: $\mathbf{\Psi}^{\text{ind}} = \mathbf{a}\mathbf{b}^T$.

5. *Correlated noise effects:* Correlated noise effects are simulated as a multivariate normal distribution with a covariance matrix described by the trait-trait correlation. The trait-trait correlation matrix $\mathbf{C}$ is constructed as follows: traits of distance $d = 1$ (adjacent trait columns) will have the highest specified correlation r, traits with $d = 2$ have a correlation of r^2, up to traits with $d = (P-1)$ with a correlation of $r^{(P-1)})$, such that the correlation is highest at the first off-diagonal element and decreases exponentially by distance from the diagonal. The final correlated noise effect matrix is simulated as $\mathbf{T} \sim \mathcal{N}_{N\times P}(\mathbf{0},\ \mathbf{C})$.

Before combining the different components into the final phenotype, each component is rescaled by a factor a such that their average column variance explains x percent of the total variance. The scale factor a is derived as follows: Let X be a random variable with expected value $E[X] = \mu_x$ and variance $V[X] = E[(X - \mu_x)^2]$ and let $Y = aX$. Then

$$\begin{aligned} E[Y] &= a\mu_x \\ V[Y] &= E[(Y - \mu_y)^2] \\ V[Y] &= E[(aX - a\mu_x)^2] \\ &= a^2 E[(X - \mu_x)^2]. \end{aligned} \tag{1.10}$$

Hence, the scaling of a random variable by a leads to the scaling of its variance by a^2. To scale the phenotype components such that their average column variance $\overline{V_{col}} = \frac{V_1 + \ldots + V_p}{p}$ explains a specified percentage x of the total variance, choose the scaling factor a such that:

$$x = a^2 \times \overline{V_{col}}$$
$$a = \sqrt{\frac{x}{\overline{\overline{V_{col}}}}} \tag{1.11}$$

The final simulated phenotype is expressed as

$$\mathbf{Y} = \mathbf{U}^{\text{scaled}} + \mathbf{C}^{\text{scaled}} + \mathbf{G}^{\text{scaled}} + \mathbf{\Psi}^{\text{scaled}} + \mathbf{T}^{\text{scaled}}. \tag{1.12}$$

In Figure 1.2, I show an example of a simulated phenotype and its different components based on the simulation strategy described above. The phenotype consists of five traits for each of the 1,000 samples from a cohort of related individuals with no population structure. There are a total of ten causal SNPs and four covariates associated with the phenotype. In addition, it is composed of background genetic and noise effects as well as a correlated noise effect (correlation: 0.8). The total genetic variance accounts for 60% of the variance leaving 40% of variance explained by the noise terms.

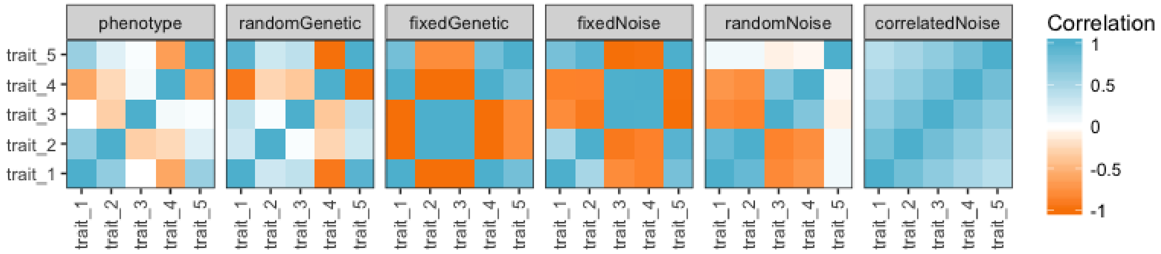


Figure 1.2: Phenotype simulation. Heatmaps of the trait-to-trait correlation (Pearson correlation) of a simulated phenotype and its five phenotype components: fixed (fixedGenetic) and random (randomGenetic) genetic effects and fixed (fixedNoise), random (randomNoise) and correlated (correlatedNoise) noise effects. The fixed noise effects consist of four independent components, two following a binomial and two following a normal distribution, the fixed genetic effect of ten causal SNPs. The highest correlation for the correlated noise effect was set at 0.8. Apart from the correlated noise component, each component was simulated with 80% of its variance shared across all traits, while the rest remained independent. The total genetic variance accounted to 60% leaving 40% of variance explained by the noise terms.

Developing new methods in quantitative genetics often requires simulated datasets with a well-characterised phenotype structure. Thereby, the number of phenotype components and their contribution to the final phenotypic variants depend on the task at hand. In order to provide a tool for phenotype simulation that is easily accessible and allows flexible simulation set-ups, I turned this simulation framework into the R package *PhenotypeSimulator*, which can be installed from the Comprehensive R Archive Network [?] describe in more detail?.

1.2 LiMMBo: Linear mixed modeling with bootstapping

Linear mixed models have become a workhorse in genetic assocation studies as they allow to control for complex sample-to-sample covariance structures that can reflect population structure and relatedness. LMM can broadly be grouped into two categories, based on the estimation of their random effect covariance terms. Exact methods estimate the covariance term anew for each SNP, while approximate methods rely on the assumption that the effect sizes of the fixed effects are sufficiently small [**??**] and an estimate of the covariance terms under the null is a good approximation. Hence, in these methods, the covariance terms are only estimated once under the null model of no fixed genetic effect and are then used as estimates in the genome-wide associations. Within these categories, one can further distinguish between methods only applicable as univariate test or tests that allow for multivariate testing. Table 1.1 summarizes commonly used frameworks and describes their computational complexity. Amongst the exact methods, FaST-LMM reduces the complexity best in terms of sample size by selecting the number of SNPs to use for the estimation of the GRM. However, it can only be applied in univariate analyses while MTMM and GEMMA extend to multivariate cases. BOLT-LMM scales best with increasing samples sizes in the group of approximate test, by directly using the genotypes and not computing or storing the GRM. All other methods have an upfront $O(N^3)$ operation for the eigendecomposition of the GRM. TASSEL reduces this complexity based on grouping of the samples and thereby effectively reducing the size of the GRM.

Eu-Ahsunthornwattana and colleagues **?** analysed several LMM frameworks including FaST-LMM, GEMMA and EMMAX with respect to their control for type I error and estimation of kinships and compared the results obtained from each method. They find that the results of most methods tested are in concordance and their performance similar in terms of power and calibration when applied to real and simulated data. In conclusion, they recommend to choose a framework based on complexity and data structure requirements.

With the generation of ever-increasing cohort sizes in genetic assocation studies, most LMM frameworks are optimised for the number of samples as described above for BOLT-LMM and TASSEL. While the remaining methods still have the upfront cubic computation of the GRM's eigendecomposition, subsequent steps have been adapted to scale linearly or quadratically with the number of samples for the majority of the applications. The optimisation in scaling for the sample dimenison comes as a trade-off for the scaling in the number of traits that can be analysed. From the multivariate methods listed above, the complexity in terms of trait number ranges from $O(P^4)$ to $O(P^7)$ which originates from the estimation of the trait-by-trait covariance component of the random effect and, in practice, limits these models to moderate trait numbers.

Table 1.1: Linear mixed model frameworks for genetic association studies. A list of popular LMM frameworks, grouped by their usage of covariance estimates when fitting the alternative model (method). P indicates the maximum trait sizes that the model can be applied to. Models with specific parameters are described in more detail in the text (FaST-LMM and TASSEL). N: number of samples; P: number of traits; S_c: number of SNPs used for singular value decomposition; c: compression factor with $c = \frac{N}{g}$ for g individuals per group; $t, t_1 and t_2$: average number of iterations needed to find parameter estimates. GRAMMAR-Gamma, FaST-LMM:t for Brenth's algorithm; GEMMA, GCTA, MTMM: t_1 for EM algorithm, t_2 for NR algorithm; BOLT-LMM: t for variational Bayes and conjugate gradients; TASSEL: t for ProcMixed algorithm in SAS; mtSet: t for LFBGS.

Method	Framework	LMM Complexity per SNP	P	Reference
exact	FaST-LMM	$O(NS_c^2 + N^2 + tN)$	1	[?]
	MTMM	$O(t_1N^3P^3 + t_2N^3P^7 + N^2P^2)$	2	[?]
	GEMMA	$O(N^3 + N^2P + t_1NP^2 + t_2NP^6)$	10	[?]
approximate	EMMAX	$O(N^3 + tN + N^2)$	1	[?]
	TASSEL	$O(\frac{1}{c^3}N^3)$	1	[?]
	GCTA	$O(t_1N^3P^3 + t_2N^3P^7)$	2	[?]
	GRAMMAR-Gamma	$O(N^3 + tN + N)$	1	[?]
	BOLT-LMM	$O(tN)$	1	[?]
	mtSet/LiMMix	$O(N^3 + N^2 + t(NP^2 + NP^4)$	$10 \sim 30$	[??]

To extend the range of LMMs for high-dimensional phenotype sets, I chose to build on an approximate model in order to avoid the repeated estimation of the trait covariance matrices. The multivariate LMM developed by Lippert, Casale and colleagues (mtSet) **??** is the only approximate model that is computationally efficient for 10-30 traits, depending on the sample and trait covariance structures. Within this framework, the genetic variance $\mathbf{G} \sim \mathcal{N}_{\mathcal{N}\times\mathcal{P}}(0, \mathbf{C_g} \otimes \mathbf{R}_N)$ and the noise variance component $\mathbf{\Psi} \sim \mathcal{N}_{\mathcal{N}\times\mathcal{P}}(0, \mathbf{C_n} \otimes \mathbf{I}_N)$ are estimated by fitting the null model of the mvLMM (Eq. **??**; omitting covariates for simplicity):

$$\mathbf{Y} \sim \mathcal{N}_{N\times P}(0, \mathbf{C_g} \otimes \mathbf{R} + \mathbf{C_n} \otimes \mathbf{I_N}) \tag{1.13}$$

The covariance structure of $\mathbf{G}$ and $\mathbf{\Psi}$ is described by the Kronecker product $\otimes$ of the genetic trait-to-trait covariance matrix $\mathbf{C_g}$ with the genetic sample-to-sample relationship matrix $\mathbf{R}$ and of the noise trait-to-trait covariance matrix $\mathbf{C_n}$ with the identity matrix $\mathbf{I_N}$, respectively (Fig. 1.3). $\mathbf{R}$ is estimated from the SNP genotypes of the samples and captures the kinship of the samples, while the noise sample-to-sample covariance is assumed to be constant. The variance decompositon (VD) of $\mathbf{Y}$ into $\mathbf{G}$ and $\mathbf{\Psi}$ is achieved by estimating $\mathbf{C_g}$ and $\mathbf{C_n}$ via restricted marginal likelihood (RML). The complexity of the VD is $O(N^2 + t(NP^2 + NP^4))$ with N the number of samples, P the number of traits, and t the number of iterations of Broyden's method for optimising the RML of the parameter estimates. From this equation, it becomes evident that as the number of traits increases, the complexity increases by a power of four and explains why this LMM set-up is not feasable for large trait sets. To overcome the bottleneck of estimating the trait-to-trait covariance matrices as a whole which is the reason for the complexity of P^4, I developed a simple heuristic that efficiently uses a linear mixed model bootstrapping (LiMMBo) approach to estimate $\mathbf{C_g}$ and $\mathbf{C_n}$.

1.2.1 Covariance estimation via bootstrapping

The key innovation of LiMMBo is to perform the variance decomposition on b bootstrap samples of s traits instead of on the whole dataset, and use those bootstrap samples to reconstruct the full $\mathbf{C_g}$ and $\mathbf{C_n}$ matrices (Fig. 1.3). In detail, from the total phenotype set with P traits, b subset of s traits are randomly selected. b depends on the overall trait size P and the sampling size s and is chosen such that each two traits are drawn together at least c times (default: 3). For each subset, the variance decomposition is estimated via the null model of the mvLMM, i.e. without the fixed genetic effect $\mathbf{x}$ (Eq. **??** and 1.13) and the $s \times s$ covariance matrices $\mathbf{C_g^s}$ and $\mathbf{C_n^s}$ recorded. For each trait pair, their covariance estimate is averaged over the number of times they were drawn. The challenge lies in combining the bootstrap results in such a way, that the resulting $\mathbf{C_g}$ and $\mathbf{C_n}$ matrices are true covariance matrices i.e. positive semi-definite and serve as good estimators of the true covariance matrices. This is achieved by fitting (least-squares estimate) the covariance estimates of the b subsets to the closest positive-semidefinite matrices via the Boyden-Fletcher-Goldfarb-Shanno algorithm (BFGS)[**?**], using the average estimates to initiate the matrices.

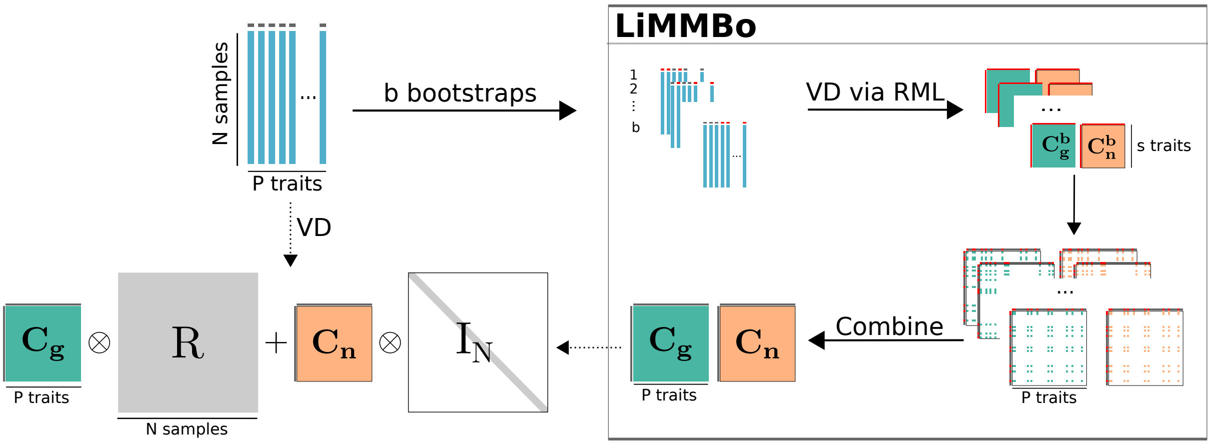


Figure 1.3: Variance decomposition. On the left-hand side, the phenotype set of P traits and N samples is decomposed into its $P \times P$ trait-to-trait covariances $\mathbf{C_g}$ and $\mathbf{C_n}$, based on the provided genetic sample-to-sample kinship estimate matrix $\mathbf{R}$. The noise sample-to-sample matrix is assumed to be constant (Identity matrix). Standardly, this is done by restricted maximum likelihood estimation of the null model of the mvLMM (Eq. 1.13). However, this direct variance decomposition (VD) via RML only works for moderate number of phenotype sizes. For higher trait-set sizes, LiMMBo serves as an alternative to the standard RML (right-hand side). Here, the phenotypes' variance components are estimated on b s-sized subsets of P which are subsequently combined into the overall $P \times P$ covariance matrices $\mathbf{C_g}$ and $\mathbf{C_n}$.

1.2.2 Data simulation

I simulated a number of different phenotype datasets to evaluate LiMMBo in terms of scalability, model calibration and power. The datasets differed in their overall trait size P, the percentage of variance explained by genetics h_2 (sum of fixed and random genetic effects) and the number of different phenotype components simulated to create the final phenotype. The phenotypes were simulated as described in Section 1.1, based on the parameters and parameter values described in Tables 1.2 and 1.3. Parameter values were chosen based on ....

Table 1.2: Parameters for phenotype simulation. The genetic and noise effects have both a random and fixed effect. For each, the total variance is the sum of their random and fixed effect variance and has to add to 1. Each fixed and random component has a certain percentage of its variance that is shared across traits, while the rest is independent.

		variance explained	shared	independent
genetic effects	total	h_2		
	fixed	h_2^s	θ	1-θ
	random	h_2^g	η	1-η
noise effects	total	(1-h_2)		
	fixed	(1-h_2)δ	γ	1-γ
	random	(1-h_2)(1-δ)	α	1-α

1.2.3 Scalability of LiMMBo

The complexitiy of the variance decomposition of the LMM framework that LiMMBo builds on is $O(N^2 + t(NP^2 + NP^4))$. The second term depends on the overall trait size and describes the complexity of estimating the trait-by-trait covariance matrices $\mathbf{C_g}$ and $\mathbf{C_n}$. By bootstrapping s-sized samples from the overall trait size, this complexity term changes to $bt(Ns^2 + Ns^4)$, with the covariance estimation carried out for b bootstraps. In addition to the estimation of the covariance terms, the overall complexity of LiMMBo also depends on the fitting the BFGS algorithm n times to the full traitset of size P. LiMMBo makes use of a cholesky decomposition of the matrices to be fitted, resulting in $\frac{1}{2}P(P+1)$ model parameters to be fitted for both $\mathbf{C_g}$ and $\mathbf{C_n}$ .Thus, the overall complexity of LiMMBo is $O(N^2+bt(Ns^2+Ns^4)+n(\frac{1}{2}P(P+1))$, which is the sum of the complexity of the bootstrap variance decompositions and the complexity of fitting the BFGS algorithm.

In order to assess and compare how LiMMBo scales, I performed VD both with LiMMBo and the standard RML approach on phenotypes with trait sizes ranging from 10 to 100 traits (parameters for phenotype simulation as described in Table 1.3, total of ten simulated datasets per setup). IFor phenotypes with 10 traits, the sampling datasize s was set to 5, otherwise $s = 10$. Fig. 1.4 shows the overall time taken by the standard

Table 1.3: Parameter values of simulated phenotypes for assessing scalability, calibration and power. The ‘genotype’ parameter specifies the simulated genotype cohort which was used to simulate fixed and random genetic effects (described in Section 1.1.1). P are the different trait set sizes that were simulated. The parameters that follow are described in Table 1.2 and specify the variance explained by each of the phenotype components. A variance explained equals zero means that this component was not simulated and corresponding non-applicable variance terms are designated with ‘-’.

	Parameter values	
Parameter	Power	Calibration
Genotypes	relatedNoPopstructure	relatedNoPopstructure unrelatedNoPopstructure unrelatedPopstructure
P	10, 50, 100	10, 20, …, 100
h_2^s	0.48, 0.3, 0.12	0
h_2^g	0.32, 0.2, 0.08	0.8, 0.5, 0.2
h_2	0.8, 0.5, 0.2	0.8, 0.5, 0.2
$(1\text{-}h_2)\delta$	0.08, 0.2, 0.32	0
$(1\text{-}h_2)(1\text{-}\delta)$	0.12, 0.3, 0.48	0.2, 0.5, 0.8
$(1\text{-}h_2)$	0.2, 0.5, 0.8	0.2, 0.5, 0.8
θ	0.6	-
η	0.8	0.8
γ	0.6	-
α	0.8	0.8

RML approach, LiMMBo and its two main components, the bootstrapping and the combination of the bootstrap results. The majority of the run time of LiMMBo is taken by the VD of the bootstrapped subsets, which accounts for at least 85% (70 traits) and on average 97% of the total runtime. As a comparison, the time taken by the standard RML approach quickly exceeds the time of LiMMBo and becomes unfeasable for more than 30 traits.

I implemented LiMMBo as a python module where the individual variance decomposition estimations can be distributed across multiple cores on a single computer or multiple nodes on a computing network via the *Parallel Python Software* **?**, allowing for the parallelisation of the most time-consuming step. LiMMBo can be freely accessed via my github/PMB github/python repo?.

1.2.4 LiMMBo yields covariance estimates consistent with RML estimates for moderate trait numbers

I evaluated the suitability of LiMMBo for covariance estimation of $\mathbf{C_g}$ and $\mathbf{C_n}$ on simulated datasets with different strength of background genetic effects. I choose to simulate traitset sizes between ten and thirty traits, as this is the regime where the standard RML approach is still feasible and allows for a comparison of the two methods. I simulated phenotype sets composed of background genetic effects $\mathbf{G}$ and noise effects $\mathbf{\Psi}$ only, omitting any specific genetic effects (additional parameters as described in Table 1.3) and estimated these variance components subsequently with LiMMBo and standard RML. Variance estimation on simulated datasets allows for the comparison of the estimated covariance matrices to the true covariance matrices based on which the phenotypes were simulated. By computing the root mean squared deviation (RMSD) between the true and estimated covariance matrices from both methods, I obtain a measure that is directly comparable and is independent of the traitset size:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n}(C_{\mathrm{true}} - C_{\mathrm{estimate}})^2}{n}} \tag{1.14}$$

Fig. 1.5 shows the comparison of both standard RML and LiMMBo-derived covariance matrices compared to the simulated, true covariance matrices. Both methods provide consistent estimates across trait sizes with little difference between the methods.

1.2.5 mtGWAS with LiMMBo-derived covariance matrices are well calibrated across all phenotype sizes

One key aspect in statistical method development is to ensure that the method is well-calibrated under the null model. Apart from gaining knowledge about the genetic and

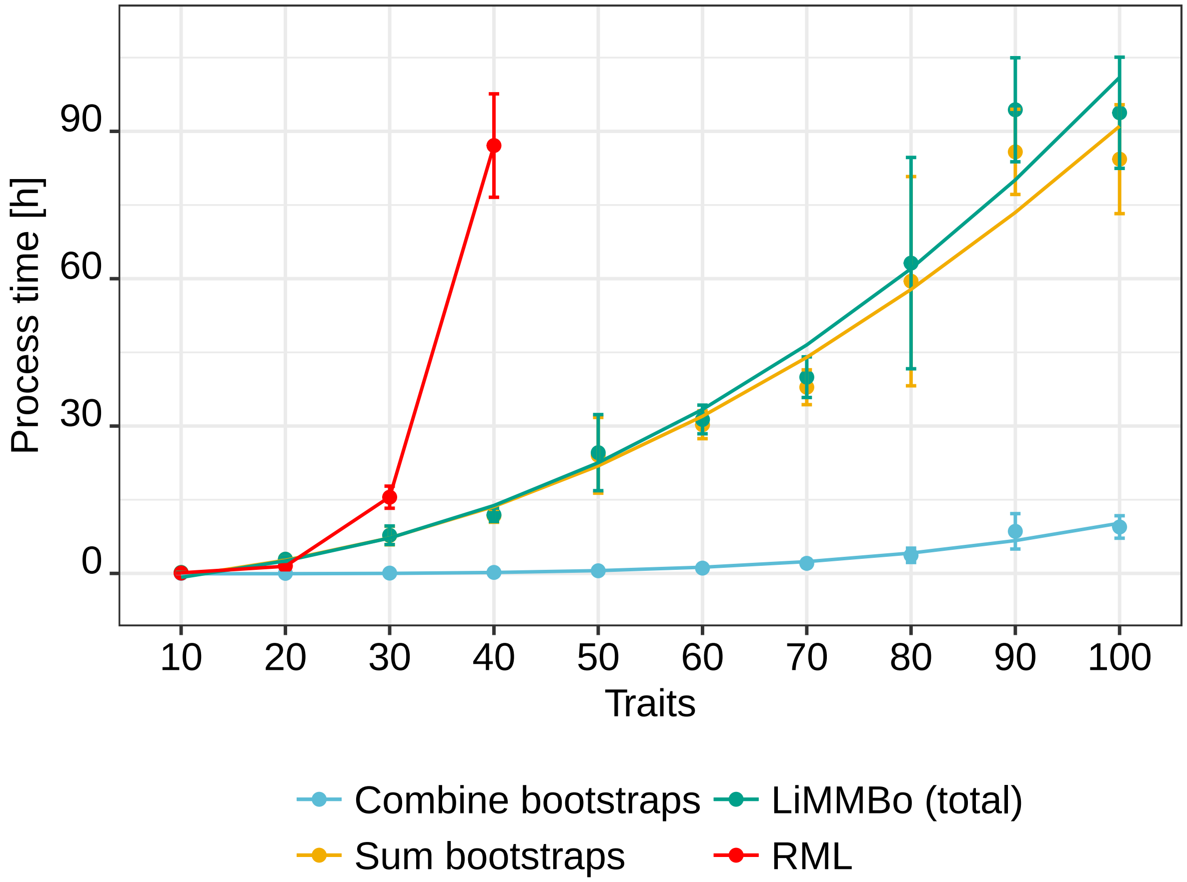


Figure 1.4: Scalability of LiMMBo compared to standard RML. Empirical run times for LiMMBo and the standard RML approach on three simulated datasets per phenotype size, with $N = 1,000$ individuals each and different amount of variance explained by the genetic background signal (0.2, 0.5, 0.8). The boxplots summarise the results across the different set-ups. Lines were fitted for the bootstrapping step (orange): $n(s^2 + s^4)$; the combination of the bootstrapping (blue): $\frac{1}{2}P(P+1)$ and their combined runtime (turquoise): $n(s^2+s^4)+\frac{1}{2}P(P+1)$. b: number of boostraps, s: bootstrap size, P: phenotype size. The majority of the runtime is required for the bootstrapping. The runtime for the standard RML results (red) are only depicted up to $P = 40$ when they already exceed the runtimes for $P = 100$ in the LiMMBo approach. The complexity of the RML and LiMMBo algorithm (or their reseptive parts) were used for fitting the lines to the observed data: sum across boostraps: $bt(Ns^2 + Ns^4)$; combining of bootstraps: $n(\frac{1}{2}P(P+1))$. total runtime of LiMMBo: $O(N^2+bt(Ns^2+Ns^4)+n(\frac{1}{2}P(P+1))$; RML: $O(N^2 + t(NP^2 + NP^4))$. Need to doublecheck with Paolo, since the line fit for the RML is perfect when assuming $O(N^2 + t(NP^2 + NP^6))$ but doesn't work when fitting to the power of 4.

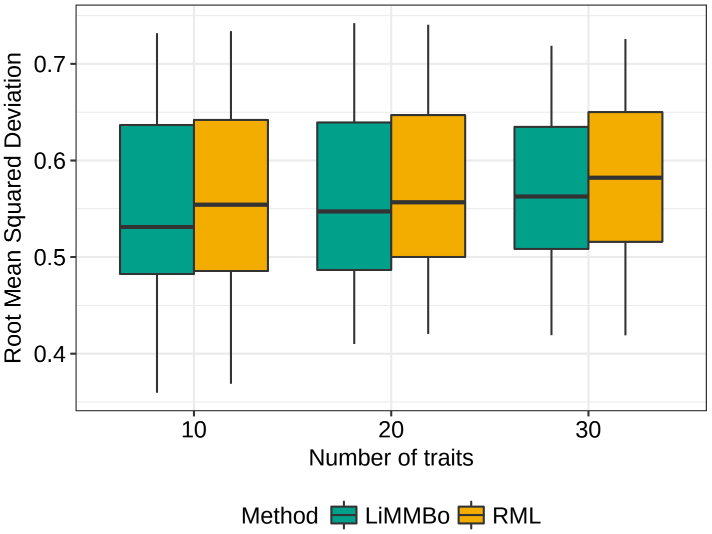


Figure 1.5: Comparison of trait-by-trait covariance estimates derived from standard RML and LiMMBo. For moderate trait set sizes ranging from 10 to 30 traits, phenotypes with different percentage of variance explained by genetics were simulated. The genetic and noise trait-to-trait covariance matrices $\mathbf{C_g}$ and $\mathbf{C_n}$ were then estimated both via LiMMBo and standard RML. These estimates were compared to the true (simulated) covariance matrix by computing their root mean squated deviation (RMSD; equation 1.14). The boxplots summarize the RMSD across different variance levels. Across all trait sizes, LiMMBo yields covariance estimates with RMSD consistent to the RML approach.

noise trait-to-trait covariance structure of a phenotype, variance decompostion into different random effect components yields estimates that can be supplied as known parameters to approximate mvLMM methods and multi-trait GWAS (mtGWAS). As introduced by Jiang and colleagues **?** and adapted by Korte and colleagues **?**, there are different model designs for mvLMM, depending on the underlying biological hypothesis regarding the effect of the genetic variant. In the most simple case, one can test if the genetic variant has an effect on any of the traits P (any effect test) i.e. the effect size of the fixed effect is unequal to zero for at least one trait : $H_\mathrm{A} : \boldsymbol{\beta} \neq \mathbf{0}_P$. In this P-degrees of freedom (df) test, the corresponding null hypthesis of no association is that the effect size of the fixed effect is equal to zero: $H_0 : \boldsymbol{\beta} = \mathbf{0}_P$. In the common effect model, the variant has the same effect size across all traits ($\boldsymbol{\beta} = \mathbf{1}_P\beta$) and is tested for significance in a 1 df model versus the null hypothesis of no association ($\boldsymbol{\beta} = \mathbf{0}_P$). A more complicated model allows to test for specific effects of the variant on a given trait p. This can be tested with a 1 df test where a model containing a common effect across all traits and a specific effect for trait p is compared against the common effect model. Both, for the calibration and power analysis I chose to apply an any effect test.

In order to test if LiMMBo-derived covariance estimates yield well calibrated test statistics, I simulated phenotype sets composed of random genetic and noise effects only with 10, 20, 30, 50 and 100 traits and parameters Table 1.3. For trait sizes of up to thirty traits, I compared the calibration of mtGWAS for LiMMBo- and RML-derived covariance matrices. As shown in Figure 1.6, both methods show sufficient calibration across all phenotype sizes and variance explained by genetics.

For higher trait sizes, I compared the calibration of mtGWAS using a mvLMM to a simple multi-variate linear model (mvLM, Eq. ??). The mvLM does not require the variance decompostion into different random effects, i.e. avoids the computational bottleneck, but simply uses pricipal components of the genotypes as fixed effects to adjust for population structure. I assessed the calibration of the mvLM and mvLMM in mtGWAS by testing all simulated SNPs against the different phenotypes and subsequently estimating the type I error rates. For each mtGWAS, I counted the number of tests that exceeded a given threshold divided by the overall number of tests conducted, hence the number of genome-wide SNPs. I tested for calibration at two significance thresholds, 5E-5 and 5E-8. The latter is the common GWAS significance threshold, which is based on the number of independent variants in the genome. It was first proposed in the scope of the HapMap project **?** who found about 150 independent, common variants per 500 kb region. At a significance threshold of $p < 0.05$ and extrapolated to the genome of ~ 3.3 Gb, by Bonferroni correction this yields $0.05/(\frac{150}{500\text{kb}} \times 3.3\text{Gb} = 5.05 \times 10^-8$. This estimate was later confirmed in a study using different methods for estimating the number of independent variants [**?**]. Tab. 1.4 shows the type I error estimates for both mtGWAS approaches. The mvLMM performs well across all trait sizes and thresholds of significance. In contrast, the mvLM is poorly calibrated and clearly demonstrates the difficulty of adjusting for population structure via fixed effects in highly structured populations.

Table 1.4: Type I error estimates for mtGWAS. Type I errors estimates for mvLMM and mvLM across all genome-wide SNPs for three trait set sizes assesed at two different levels of significance. For the mvLMM, covariance estimates were derived via LiMMBo. In the mvLM, population structure was adjusted for via the first ten PCs of the genotype data. The mvLMM controls well for Type I errors at both thresholds, while the mvLM leads to inflated test statistics.

		Type I Error estimates	
Traits	Significance level	mvLM	mvLMM
10	5.00E-05	2.17E-03	4.51E-05
	5.00E-08	2.32E-05	<5E-08
50	5.00E-05	1.09E-02	1.93E-05
	5.00E-08	2.08E-04	<5E-08
100	5.00E-05	3.17E-02	2.28E-05
	5.00E-08	1.01E-03	4.56E-08

1.2.6 Multi-trait genotype to phenotype mapping increases power for high-dimensional phenotypes

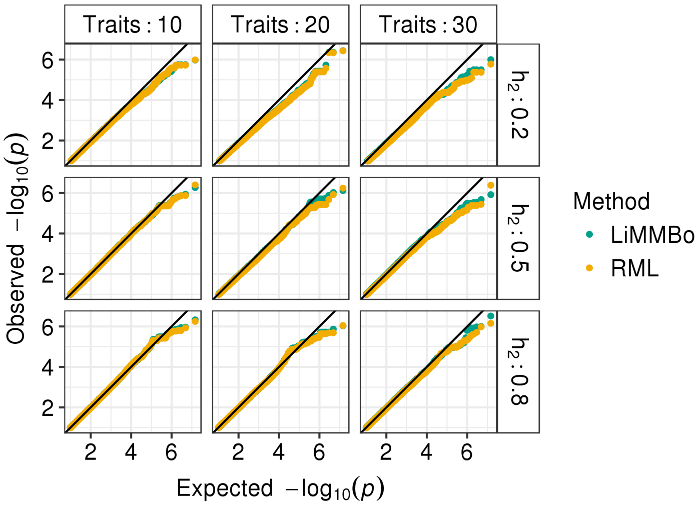


Figure 1.6: Calibration of mtGWAS based on covariance estimates from standard RML and LiMMBo. Phenotypes ranging from 10 to 30 traits with different percentage of variance explained by genetics were simulated. The genetic and noise trait-to-trait covariance matrices $\mathbf{C_g}$ and $\mathbf{C_n}$ were then estimated both via LiMMBo and standard RML and used as estimates for mvLMM across all genome-wide SNPs. Both methods show a unifrom distribution of observed p-values across all phenotype sizes and variance explained by genetics

Chapter 2

LiMMBo applied to multitrait GWAS in *Saccharomyces cerevisiae*

The often large number of phenotypes measured in cohort studies offers the possibilty to test LiMMBo on a real world dataset. Amongst the publically available studies, such as flowering, defense and developmental phenotypes in arabidopsis thaliana [?] or human blood metabolites [?], I found the dataset of 46 quantitiative traits in yeast generated and analysed in the study by Bloom and colleagues [?] most suitable for several reasons. The study investigated the growth of a yeast F2 cross on several different substrates. First, the genetic architecture of an F2 cross is highly structured, making it an ideal test scenario for a linear mixed model capable for adjusting and profiting from population structure in the sample. Second, the measured phenotypic traits have a broad spectrum of correlation, with highly related phenotypes for metabolically similar compounds to very low correlation for certain chemicals. At the same time, the phenotypic measurements are all obtained by measuring the growth size of the colonies, hence, the variable type und unit does not change across the phenotypes. Lastly, the generation and quality control of the data was well described and easily accessible in a user-friendly format. However, as with many studies where multiple measurements per sample are obtained, not all samples were fully phenotyped.

Linear mixed models and methods based theron such as LiMMBo require full phenotyping of each sample, i.e. the model cannot deal with missing values. In order to understand how to deal with missing values in the dataset, it is important to have an understanding of the underlying process generating the missing data [?]. In general, one can distinguish between three processes, missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR) [?]. Their formal definitions are based on the data $X \in R^{n,p}$, the binary indicator matrix $M \in R^{n,p}$ and ϕ, the (unknown) parameter of the missing data process, i.e. the parameter of the conditional distribution g_ϕ of M given X. n is the number of observations and p the number of observed variables. M is an indicator whether an observation is missing $m = 1$ or not $m = 0$. The data X can formally be grouped into $X = X_{obs} + X_{miss}$, where X_{obs} and X_{miss} are the observed and missing parts of the data, respectively. Data are MAR if the distribution of missingness only depends on X_{obs}

$$g_{\phi}(M|X, \phi) = g_{\phi}(M|X_{\text{obs}}, \phi), \forall X_{\text{miss}}, \phi. \tag{2.1}$$

If the the distribution is also independent of X_{obs},

$$g_{\phi}(M|X, \phi) = g_{\phi}(M|\phi), \forall X, \phi, \tag{2.2}$$

then the data is MCAR. If on the other hand, the distribution of missingness is dependent on X_{miss}, hence

$$g_{\phi}(M|X, \phi) = g_{\phi}(M|(X_{\text{obs}}, X_{\text{miss}}\phi), \forall X, \phi, \tag{2.3}$$

then the data is MNAR. To illustrate these cases, consider an example where there are n colonies of yeast and one wants to automatically detect the size and the density of each colony with a suitable instrument. If the instrument fails with a constant probabilty ϕ for any colony independent of the measurement, then the pattern of missing values in the data is MCAR. If the probability that the density measurement is missing changes with the value of the size measurement, but is not dependent on the density of colonies with the same size, then the data are MAR. In contrast, data are MNAR if the probability of obtaining a density measurement depends on the density of colonies with the same size.

In practice, detecting the missing data mechanism often proves difficult. Testing fo MCAR can be done via statistical tests [?], but distinguishing between MAR and MNAR cannot be achieved formally as this would require knowledge of the missing values [??]. However, there are visualisation tools that provide diagnostic plots and approximate measures available which can help make assumptions about the missingness mechanism [??].

When analysing datasets with missing data, there are four general approaches to choose from: i) methods simply based on the complete data only ii) methods on complete data with weighting procedures, iii) model-based and iv) imputation-based procedures. In the first class, incompletely recorded samples are simply excluded, which is the most easy to implement method, but is inefficient and can lead to major bias, especially if the data is MNAR [?]. Weighting procedures also exclude incompletely sampled data, but apply a weighting to the recorded samples, where the weights attempt to adjust for the missing data as if it were part of the sample design. Model-based procedures define a model for the observed data and base inference and parameter estimates on the likelihood or posterior distribution of that model. The last class of methods, imputation-based approaches estimate the missing values based on the observed values and the then completed dataset can analysed by standard methods (an extensive review of the different methods can be found in [?]). The precise usage of the methods and underlying assumptions will dependent on the missing data mechanism.

I found the imputation approach most applicable for dealing with the missing phenotype values in the yeast dataset as they were simple to apply, did not lead to a decreased sample size and possible loss in power (as method i) and did not require recasting the model underlying LiMMBo (as would have been required for method iii). There are vast number of imputation methods available, that can be categorised by both the method for imputation and the number of times the missing values are imputated. Methods include simple mean prediction, where the missing data for a given variable is replaced by the mean of all known values of that variable and derivations thereof such as KNN or FKM which use the mean of the k-nearest neighbours to replace the missing values [??]. Instead of imputing based on the mean, i.e. the center of a distribution, other strategies use random draws from a predictive distribution of plausible values of the missing value. The predictive distribution is conditioned on the observed data when creating the predictive distribution. These techniques can then be used to either impute one value for each missing item (single imputation) or more than one value to account for imputation uncertainty (multiple imputation) [?]. For complex datasets, multiple imputation have emerged as the method of choice [??]. To impute the missing phenotype data in the yeast study, I choose MICE, a method based multivariate imputations by chained equations [?], which generates imputations for multivariate data by Gibbs sampling.

In the following chapter, I will first describe the data processing and imputation strategy for the yeast phenotypes. I will then show the results of applying LiMMBo and subsequent multi-trait GWAS to the dataset and compare the association to the association obtained from single-trait GWAS. Finally, I will explore the benefits of jointly modelling large numbers of traits in genetic studies.

2.1 Data set and imputation

The dataset consists of phenotype and genotype data of 1,008 prototrophic haploid *Saccharomyces cerevisiae* segregants derived from a cross between a laboratory strain and a wine strain strain. It contains 11,623 unique genotypic markers obtained via short-read sequencing for all 1,008 segregants (no missing genotypes). For phenotyping, segregants were grown on agar plates under 46 different conditions, including different temperatures, pH and nutrient addition (see labels in Figure 2.3). The phenotypes were definded as end-point colony size normalized relative to growth on control medium. For the remainder of this chapter, a trait is defined as the normalised growth size in one conditions. Out of the 1,008 segregants, 303 segregants were phenotyped for all 46 traits.

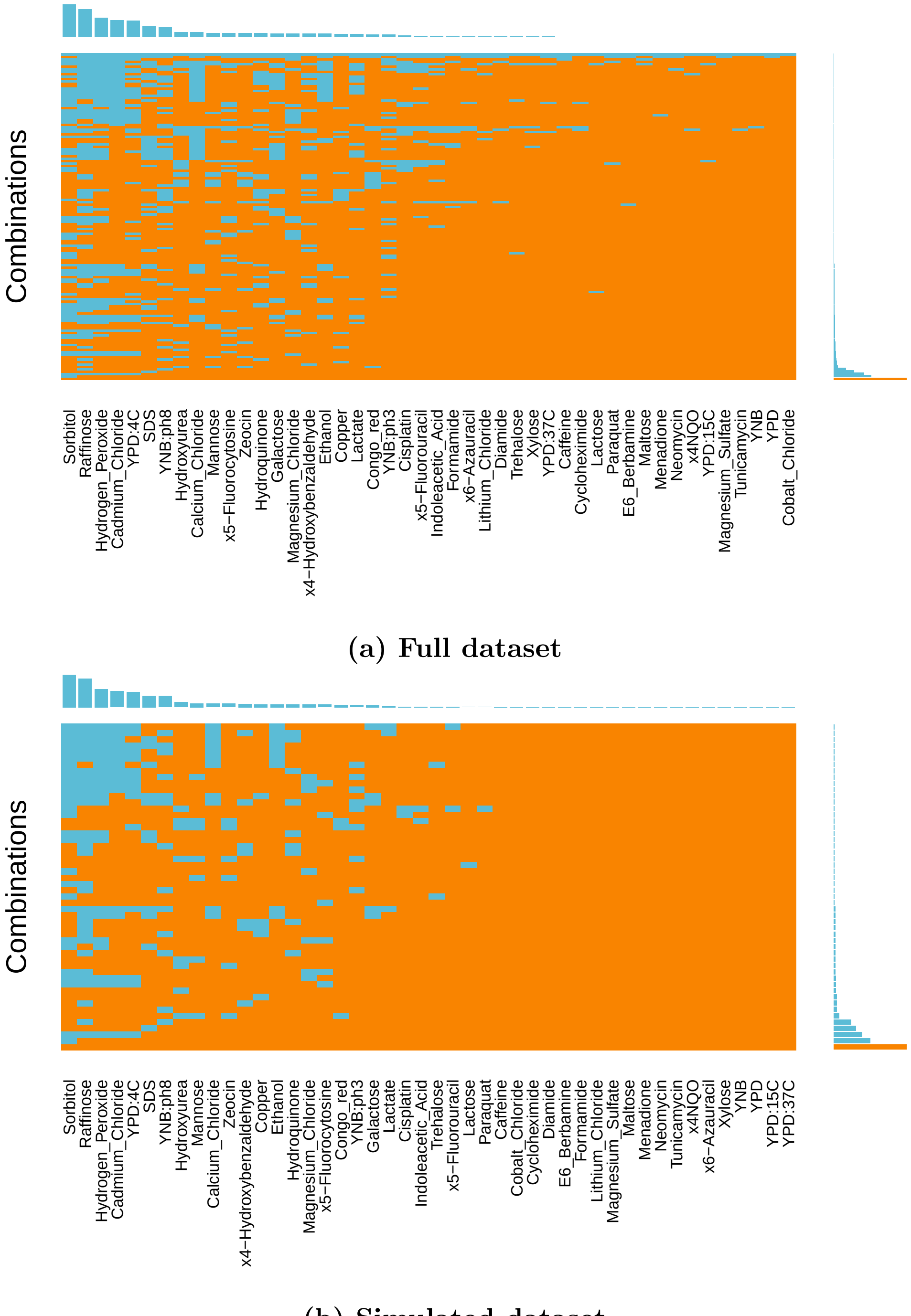
2.1.1 Missing data mechanism

In order to gain an understanding of the dataset, I first looked at the frequencies and distribution of missing values. There are 135 different combinations of missing values across the samples and the missing phenotypes are not evenly distributed (Figure 2.1a). Some traits such as cobalt chloride are present for almost all samples while others such as sorbitol or raffinose are lacking in more than a third of the samples. I used Little's global test for MCAR to analyse wether these observed data patterns can be acocunted for through a MCAR mechanism. It tests the null hypothesis that the data is MCAR [**??**], which can in this case be rejected with a p-value of 2E-34 (based on a χ^2 dsitribution, $\chi^2 = 5,902$, $df = 4,63$).

Determining if data is MAR or MNAR cannot be tested for formally and relies on approximate measures and assumptions based on the experimental procedures [**???**]. Garson suggests to use significance tests of missingness **?**. If it can be demonstrated that one or more variables in the dataset are significantly correlated with missing values, missingness may be predictable, which is the requirement for imputing MAR data. In order to test for predictable missingness, I created an indicator matrix for the phenotype matrix, where observed values where encoded as 0 and missing values as 1. For each of the 46 traits in the dataset I correlated the observed values across all samples with each column of the indicator matrix, i.e. the missingness patterns per trait. If all values were observed for a given trait i.e. all values in the incator matrix in this column were equal to 0, then the correlation between the trait and the missingness was set to NA. Figure 2.2 shows these correlation patterns between the phenotypes and the missing values per trait. The p-values of the correlations were adjusted for multiple testing via Benjamini and Hochbergs method [**?**] and only significant correlations are depicted ($FDR < 0.2$. For traits like cobalt chloride and magnesium sulfate where little data is missing, many entries are NA. Overall, fo a number of traits and missingness patterns, I found sufficient evidence for predictable missingness and MAR assumptions for further analyses were considered valid. Most importantly, for data with MAR, the missing data mechanism is ignorable for maximum likelihood based methods and no further adjustments for the mechansims have to be made in the modeling [**??**]. Thus, the MAR assumption of missingness in the yeast data allows for imputation via the likelihood-based method of multiple imputation.

2.1.2 Imputation via MICE

Before imputing the missing values in the dataset, I wanted to understand which missing trait values can be reliable imputed and find the best parameter settings for the imputation. In order to do this, I needed a fully phenotyped dataset with the same structure as the yeast dataset, where missing values could be introduced, imputed and subsequently compared to the true values. I chose a simple approach and used the subset of 303 fully



(a) Full dataset

(b) Simulated dataset

Figure 2.1: Frequencies and distributions of missing values in the yeast phenotype data. In both panels, the aggregation plot (middle) depicts all existing combinations of missing (blue) and non-missing (orange) values in the traits. The bar chart on its right shows the frequencies of occurrence of the different combinations. The histogram on the top shows the frequency of missing values for each trait. (R Package: *VIM* [?]). (a) The full dataset contains normalised colony sizes for growth in 46 different conditions of 1,008 genotyped yeast segregants. 306 segregants are fully genotyped (bar chart, orange bar). (b) Fully-phenotyped dataset of 306 segreagants with simulated missing values based on the observed missingness pattern for the entire pool of 1,008 segregants.

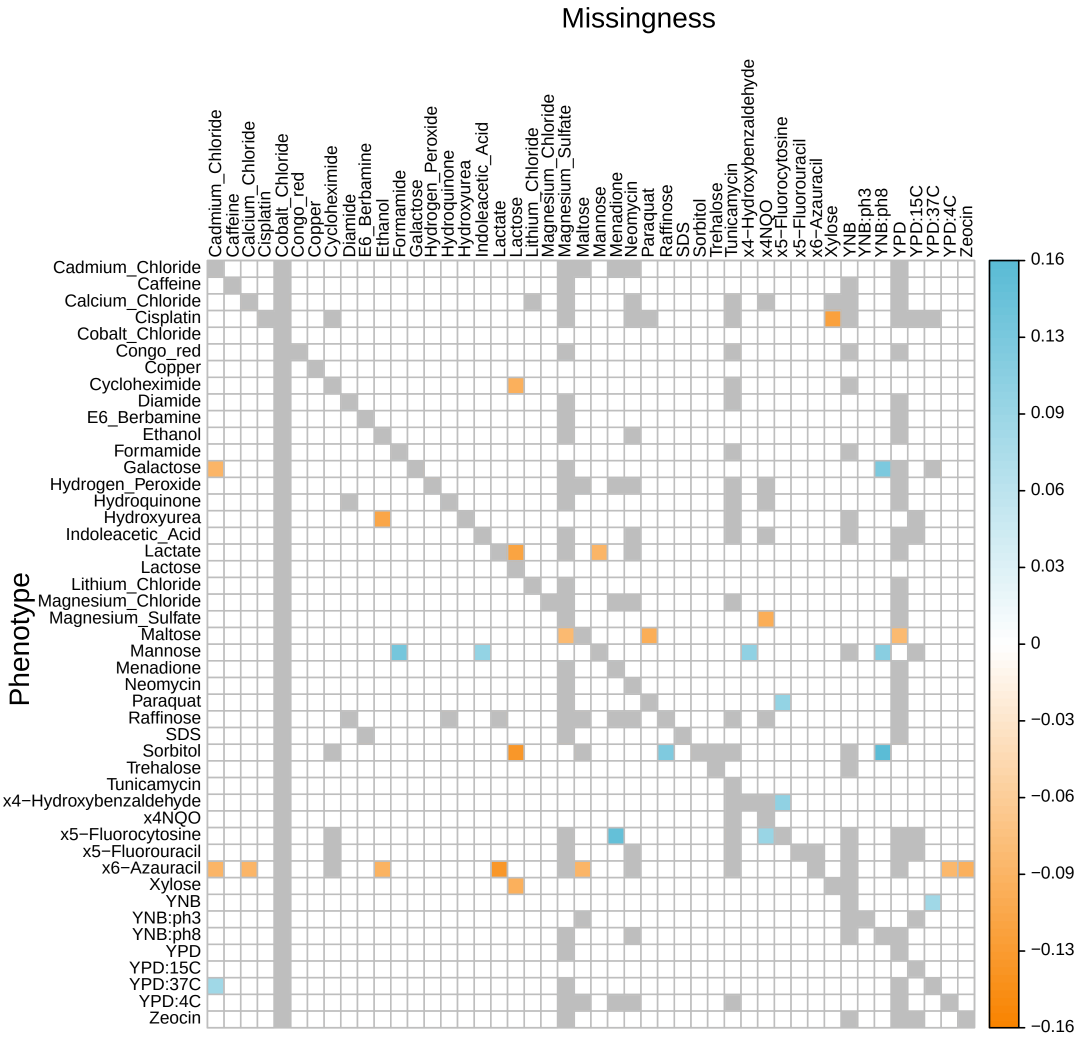


Figure 2.2: Correlations of observed phenotypes with missing data values. For each trait, Spearman's rank correlation coefficient ρ was computed with each column of the indicator matrix of the phenotypes, containing 0 for observed values and 1 for missing values. The p-values of the correlations were adjusted for multiple testing according to Benjamini and Hochberg's method [**?**]. The strength and the direction of significant correlations ($FDR < 0.2$) are depicted above., with the original phenotypes in rows and the indicator matrix of the phenotypes across columns. Unsignificant correlations are left blank. Grey squares indicate NA, i.e. columns in the indicator matrix for which no traits were missing when correlated with the observed values for a given trait.(R Package: *corrplot*).

phenotyped samples and introduced missing values with a similar pattern of missingness as observed in the original dataset. The results for the real (Figure 2.1a) and simulated (Figure 2.1b) dataset are similar in terms of frequencies and combination of missing/non-missing traits .

I used this simulated dataset as input to the imputation framework based on multiple imputation by chain equations (MICE) [**?**].

MICE. MICE belongs to the general class of multiple imputation frameworks, where several imputed versions of the dataset are generated. The imputations are done separately for each variable. The imputed values are chosen from plausible values drawn from a distribution that is specific for that variable, in this case for each trait. This distribution is derived from the dataset $X \in R^{n,p}$ itself, with $X = (X_{\text{miss}}, X_{\text{obs}})$ (the missing and the observed parts of the data), the binary indicator matrix for missingness $M \in R^{n,p}$ and a set of predictor variables Z. The MICE algorithm is usually divided into four steps [**????**]:

1. Specify the posterior predictive density $p(X_{\text{miss}}|Z, M)$ given the non-response mechanism $p(M|X)$ and the complete data model pX).

2. Draw imputations from this density to produce m complete data sets.

3. Perform m complete-data analyses on each completed data matrix.

4. Pool the m analyses results into final point and variance estimates.

Garson describes the possibilty of switching step 3 and 4 **?**, where the multiple imputations are first pooled and the subsequent analyses run on the pooled estimates. Employing this approach allows me to obtain reliable imputation estimates while having to estimate the variance components via LiMMBo only once. As described in the previous chapter, LiMMBo strongly reduces the computation time for the variance decomposition, but it is still the time-consuming factor in the analysis.
The two main choices in applying MICE for imputation have to be made in step 1: the form of the imputation model and the choice of predictor variables.

Imputation model. From the different imputation models available (examples described in [**?**], I found multivariate data predictive mean matching (PMM) a fast and sensible imputation model. PMM is a semi-parametric method which preserves non-linear relations in the data [**??**]. In brief, PMM finds the mean and covariance of the multivariate distribution X with missing values (often simply based on the complete cases). Subsequently, for each incomplete sample it predicts the missing values X_{miss} based on X_{obs} and the provided predictor variables Z. In addtion, values of the complete samples for the same set of X_{miss} are predicted. The predicted values of the incomplete sample are than matched to the predicted values of the complete samples and the closest match is chosen. The imputed values for the incomplete sample are set to those of the closest match[**?**].

Predictor variables. As many predictor variables as possible should be included in the imputation to obtain the least amount of bias and maximal certainty about the predictions [**?**]. In addition, Schafer showed that using this strategy makes MAR assumptions more plausible [**?**]. However, not all predictors will be relevant and the choice of predictors can be done on a per-variable level. In order to select suitable predictors for each trait, I first computed the pairwise Spearman correlation coefficient ρ for all traits across the 303 fully-phenotyped segregants. Some of the traits like cadmium chloride or neomycin show very little correlation to any of the other traits, while many of the traits based on growth on different carbohydrate resources form a large cluster of moderate to strong correlation (Figure 2.3). I tested several sets of predictor variables, either using all traits as predictors or chosing predictors based on the pairwise ρ of the traits. For each trait, I included predictors that showed a correlation higher than a predefined threshold ($\rho = \{0.1, 0.2, 0.3\}$. In addition, I restricted the predictors to traits that had been measured in at least 20% of the samples in the dataset. This excluded cadmium chloride (0.21% missing), hydrogen peroxide (0.24%), raffinose (0.34%), sorbitol (0.41%) and YPD:4C (0.20%) as predictor variables, but did not exclude them from being imputed.

Further parameters for MICE are the number of multiple iterations m (set to $m = 20$) and the number of iterations $maxit$ (set to $maxit = 30$). For each predictor set-up, I initiated MICE with the same seed for the random number generator to ensure comparability. After imputation, I evaluated the goodness of the imputation by computing the Spearman correlation of the imputed values (averaged across iterations m) to the experimentally observed ones (Figure 2.4). Traits where the imputed values correlated to the original ones by more then 95% in at least one of the predictor set-ups were retained in the analysis. For five traits (cadmium chloride, hydrogen peroxide, raffinose, YNB:ph8, YPD:4C), no suitable predictors could be determined and these were excluded from further analyses (Figure 2.4, red labels). For each trait, I chose the predictor scheme that yielded the highest correlation between the imputed and observed data for the imputation of missing values in the full dataset. Missing values were imputed in segregants that were phenotyped for at least 80% of the traits. The final dataset contained 981 segregants with phenotypes for 41 traits each.

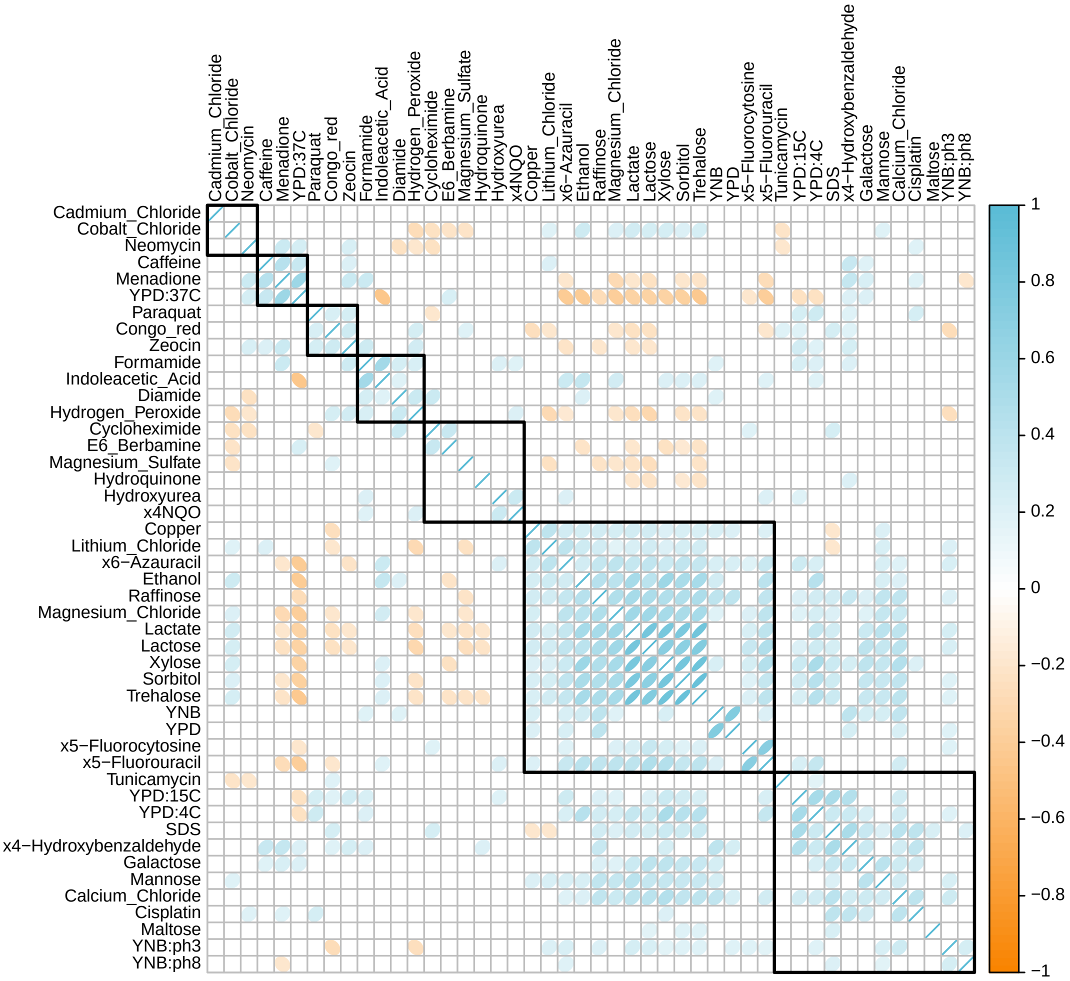


Figure 2.3: Pair-wise correlations of 46 growth traits in *Saccharomyces cerevisiae*. For each trait pair, Spearman's rank correlation coefficient ρ and the p-values of the correlation were computed. The p-values were adjusted for multiple testing according to Benjamini and Hochberg's method [**?**]. The strength and the direction of significant correlations ($p < 0.05$) are depicted above. Unsignificant correlations are left blank. The traits are clustered based on complete-linkage clustering of $(1 - \rho)$ as distance measurement (R Package: *corrplot*).

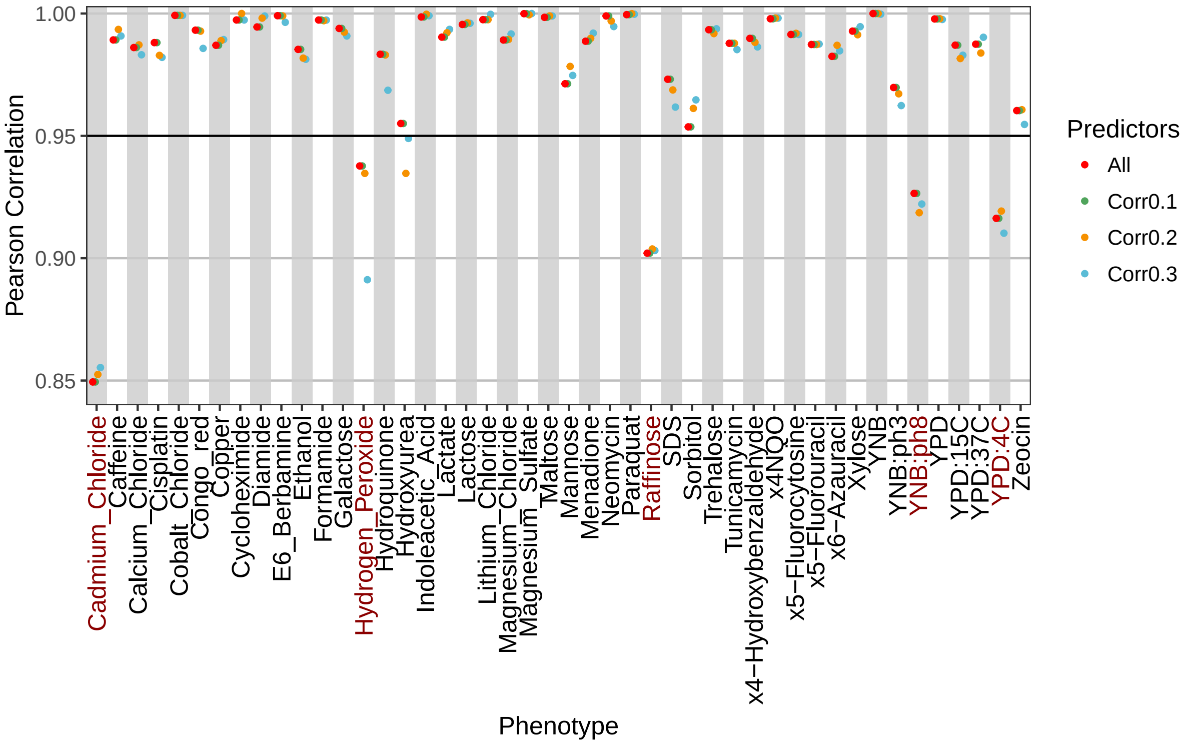


Figure 2.4: Correlation between imputed and experimentally observed trait values. In the subset of 306 fully phenotyped samples, missing values were introduced and subsequently imputed via MICE. Different predictor sets were tested based on Spearman's rank correlation coefficient: traits were considered predictors if their correlation with the target trait was greater than a given threshold. For each predictor setup (all traits as predictors and predictors passing the corrleation threshold $\rho = \{0.1, 0.2, 0.3\}$, $m = 20$ multiple imputations and $maxit = 30$ iterations of MICE were conducted. The goodness of the imputation was evaluated by computing the correlation of the imputed values (averaged across iterations m) to the experimentally observed ones. Traits with at least one correlation greater than the 0.95 (black vertical line) were retained in the dataset. For traits labeled in red, the imputation was considered to be unreliable and the traits were excluded from further analyses (R Package: *mice* [?]).

2.2 Multi-trait GWAS

In order to show the utility of LiMMBo for joint high-dimensional phenotype analyses and demonstrate the advantages over single trait approaches, I analysed the imputed dataset both with single-trait and multi-trait GWAS.

2.2.1 LiMMBo increases power in detecting genetic associations

For both analyses, I used a LMM where the sample-by-sample component of the random genetic effect is based on the genetic relationship matrix. To obtain an estimate of the genetic relationship matrix, I first pruned the genome-wide SNPs (11,623) for SNPs that are in LD with $r^2 > 0.2$ within a window of 3kb. As the dataset is based on an F2 cross, LD structure estimation is not straight-forward and this window size is a simple estimate derived from a study on the population genomics of domestic and wild yeasts [?]. The LD pruning reduced the SNP set for GRM estimation to 4,105 SNPs. The GRM was estimated via plink **?**, which follows the method introcuded by Yang and colleagues [?].

The first step in the mtGWAS is the trait-by-trait covariance estimation via LiMMBo. After xx bootstraps of ten traits each, every trait-trait combination was sampled and its covariance estimated at least three times. Together with the subsequent combination of the bootstrapped covariance estimates, the runtime was xxx. The combined trait-trait covariance estimates $\mathbf{C_g}$ and $\mathbf{C_n}$ were then used as input estimates for the second step in the mtGWAS, the mvLMM (Eq. **??**) across all genome-wide SNPs. I used a mvLMM with a trait-design matrix corresponding to the any effect test, i.e. testing for an effect of the SNP on any of the traits compared to the null hypothesis of no association (Section **??**).

For the stGWAS, the trait-by-trait components of the random effects will be point estimates (σ_g and $sigma_n$) derived within the LMM and do not require *a priori* estimation. The stGWAS is based on a univariate LMM (Eq. **??**) per SNP, where each trait is mapped individually. To account for the number of univariate tests, the pvalues obtained from the stGWAS were adjusted for multiple testing by the effective number of conducted tests M_{eff}. M_{eff} was introduced by Galwey and colleagues **?** and adjusts for multiple testing in a manner similar to the Bonferroni method [?]. However, it is less conservative, as it does not adjust for total number of tests, but the estimated, effective number of tests, taking correlation between the variables and tests into account:

$$M_{eff} = \frac{(\sum_{i=1}^{M} \sqrt{\lambda_i})^2}{\sum_{i=1}^{M} \lambda_i}, \tag{2.4}$$

where λ are the eigenvalues of the phenotypes' correlation matrix. To adjust for multiple testing in the stGWAS, the pvalues are multiplied with M_{eff} and set to 1 if the multiplication leads to values greater than one. M_{eff} for the 41 growth traits was estimated to be 33.

In order to assess the significance of the single-trait and multi-trait analyses, I followed approaches of previous association studies in yeast crosses [???], where permuations are used to estimate empirical significance levels. With a conservative, theoretical significance threshold of $p_\mathrm{t}10^{-5}$ at most one SNP is expected to be false positive with a total of $s = 11,623$ SNPs . To find the empircal FDR corresponding to this threshold, I generated $p = 50$ permutations of the genotypes and fitted the LMM against these permutations. The pvalues obtained from these analyses were then combined and sorted in increasing order. The pvalue observed at position $p_\mathrm{t} \times s \times p$ is used as the empirical FDR. For the mtGWAS this threshold is at $\mathrm{FDR}_{\mathrm{mtGWAS}} = 1.2e - 05$ lower than for the stGWAS $\mathrm{FDR}_{\mathrm{stGWAS}} = 8.6e - 06$.

Fig. 2.6 shows the manhattan plot of the multi-trait and single-trait GWAS. On several chromsomomes, mtGWAS peaks (blue) are observed whereas no stGWAS peaks (orange; minimum p-value per SNP across all 41 stGWAS, adjusted for multiple testing) can be detected. This results demonstrates the increase in power on real data when jointly modeling the traits, confirming the results obtained from the theoretical power analysis (Section ??).

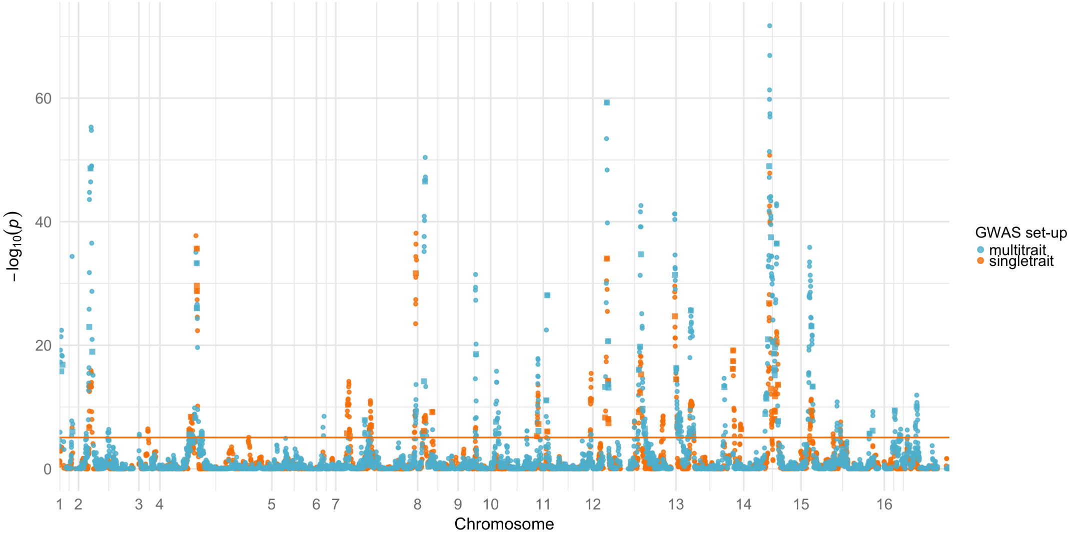


Figure 2.5: Manhattan plot of pvalues from stGWAS and mtGWAS. The stGWAS pvalues were adjusted for multiple testing by the effective number of tests ($M_{eff} = 33$ and only the minimum adjusted p-values across all 41 traits per SNP are shown. The significance line is drawn at the empirical $\mathrm{FDR}_{\mathrm{stGWAS}} = 8.6e - 06$.

To quantify the increase in power, I counted the number of loci detected above the permutation-based significant thresholds for both the stGWAS and the mtGWAS. However for a fair comparison of the two methods, I also needed to account for linked loci, with the long LD structure present in the F2 cross potentially merging nearby signals. As for the estimation of the GRM, I used *plink* for LD pruning of the SNPs, chosing a strict threshold of $r^2 > 0.2$ and increasing LD window sizes ranging from 3 to 100kb. The maximal LD window of 100kb covers between 6% (chromosome 4) and 43% (chromosome 1) of total chromosome length (yeast genome assembly: ScerevisaeR64-1-1 []). Table 2.1 shows that the increase in power is present from narrow to broad LD pruning, with on average 29% more significant loci in mtGWAS.

Table 2.1: Comparison of significant loci in stGWAS and mtGWAS. In column 'All SNPs', the absolute number of SNPs beyond the significance threshold for multitait and singletrait GWAS as well as their ratio (multitrait:singletrait) are depicted. In order to limit the potential bias in the counting of the loci, introduced by different degrees of linkage disequilibrium (LD) for different loci, the genome-wide SNPs were LD pruned and the ratio of significant SNPs determined for five different LD window sizes. The maximal LD window covering between 6% (chromosome 4) and 43% (chromosome 1) of total chromosome length.

	All SNPs	LD pruned with $r^2 > 0.2$				
		3kb	10kb	30kb	50kb	100kb
NrSNPs	11623	4105	1028	264	161	107
multitrait	1132	384	101	24	15	9
singletrait	695	275	72	20	13	7
multitrait:singletrait	1.63	1.4	1.4	1.2	1.15	1.29

2.2.2

As well as providing an increase in power, the mtGWAS inherently provides effect size estimates across all phenotypes for a particular locus, allowing a richer exploration of the underlying biology. To analyse the relationship between traits and SNPs based on their effect size estimates, I filtered the genome-wide SNPs for SNPs that fell within a gene body (yeast genome assembly: ScerevisaeR64-1-1). I pruned these 8,135 SNPs for SNPs in LD with $r^2 > 0.2$ and within a 3kb window (1,412 SNPs). Lastly, I filtered for SNPs passing the $\text{FDR}_{\text{mtGWAS}} = 1.2e - 05$ yielding 210 SNPs across 15 out of the 16 yeast chromosomes. Chromsome V is the only chromosome without significantly associated SNPs for the mtGWAS (Figure 2.6).

To find groups of SNPs and traits with similar effect size estimates, I clustered the effect size estimates of these SNPs both across traits and SNPs. I used the hierarchical clustering algorithm *pvclust* that provides bootstrap-based p-values as a measure for the stability of a given cluster **?**. The clustering was based on the Pearson correlation coefficients, with 50,000 bootstraps for traits and 10,000 for SNPs. Clusters with $p < 0.05$ were considered significant. A heatmap effect size estimates and the clustering results is depicted in Figure **??**. Ignoring the clustering for a first impression of the results, one can clearly see that most SNPs have significant non-zero effects in more than one trait. Furthermore some traits have contributions from across the genome, many of which are xenobiotic growth conditions e.g. zeocin **?** and neomycin **?**. Turning the attention to the clustering, Figure **??** shows that the clusters are driven by specific combinations of loci and traits, and would be hard to achieve from a single trait analysis.

There are a number of significant clusters for traits (Figure **??**, coloured row dendrogram), including classically linked carbon metabolism sources (lactose, lactate and ethanol), and other clusters which there is literature support for. For example, I found a study showing gene expression changes for genes involved in DNA replication upon treatment with hydroxyurea and 4-nitroquinoline-l-oxide (x4NQO) **?**, two substances that form stable cluster. Supporting the cluster of trehalose and sorbitol is a previous results, which demonstrated these sugars to have synergistic effects on viability in yeast **?**. For other clusters, such as SDS and Hydroxybenzaldehyde or magnesium sulfate and berbamine I was unable to find literature support but this could be a candidate clustering of these growth phenotypes for further investigation.

For the clustering across SNPs, I discovered 31 stable clusters (Figure **??**, coloured column dendrogram), many of which represent linked loci. However, there are nine clusters (Figure **??**B, grey boxes) spanning multiple chromosomes, and many clusters with disjoint regions across a chromosome. Some SNP clusters have suggestive common annotation, such as c*cluster a* which has two members of the nuclear pore complex, and *cluster b* which has a common set of vesicle associated genes (Fig. 2.6B, labeled boxes). The small size of clusters inhibited any systematic gene ontology based enrichment, but the ability to explore both multiple traits and multiple loci from the mtGWAS provides stimulating hypothesis generation.

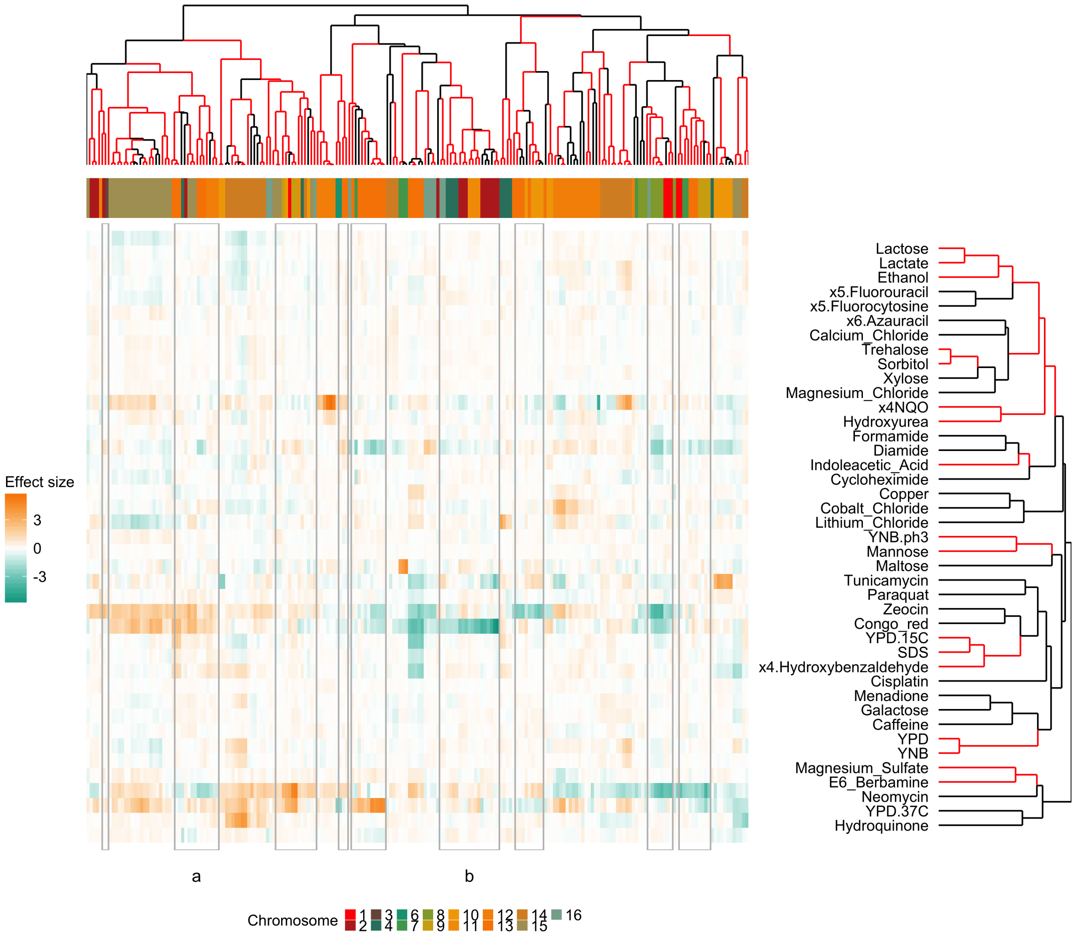


Figure 2.6: mtGWAS effects size estimates. Effect size estimates of LD-pruned (3kb window, $r^2 > 0.2$), significant SNPs located within a gene were clustered by loci and traits (both hierarchical, average-linkage clustering of Pearson correlation coefficients). Stable clusters (pvclust $p < 0.05$) are marked in orange. Grey boxes indicate stable SNP clusters spread across at least two chromosomes. a and b label two clusters for which suggestive common annotation was found, for details see test.