# Modeling the polygenic architecture of complex traits

**Dissertation**

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Barbara Rakitsch

aus Moosburg an der Isar

Tübingen

2014

| | |
|---|---|
| Tag der mündlichen Qualifikation: | 19.11.2014 |
| Dekan: | Prof. Dr. Wolfgang Rosenstiel |
| 1. Berichterstatter: | Prof. Dr. Karsten Borgwardt |
| 2. Berichterstatter: | Prof. Dr. Bertram Müller-Myhsok |

# Zusammenfassung

Die Genomforschung ist innerhalb der letzten Jahre stark gewachsen. Fortschritte in der Sequenzierungstechnologie haben zu einer wahren Flut von genomweiten Daten geführt, die es uns ermöglichen, die genetische Architektur von komplexen Phänotypen detaillierter als jemals zuvor zu untersuchen. Selbst die modernsten Analysemethoden stoßen jedoch an ihre Grenzen, wenn die Effektgrößen zwischen den Markern zu stark schwanken, Störfaktoren die Analyse erschweren, oder die Abhängigkeiten zwischen verwandten Phänotypen ignoriert werden. Das Ziel dieser Arbeit ist es, mehrere Methoden zu entwickeln, die diese Herausforderungen effizient bewältigen können.

Unser erster Beitrag ist der LMM-Lasso, ein Hybrid-Modell, das die Vorteile von Variablenselektion mit linearen gemischten Modellen verbindet. Dafür zerlegt er die phänotypische Varianz in zwei Komponenten: die erste besteht aus individuellen genetischen Effekten. Die zweite aus Effekten, die entweder durch Störfaktoren hervorgerufen werden oder zwar genetischer Natur sind, sich aber nicht auf individuelle Marker zurückführen lassen. Der Vorteil unseres Modells ist zum einen, dass die selektierten Koeffizienten leichter zu interpretieren sind als bei etablierte Standardverfahren und zum anderem diese auch an Vorhersagegenauigkeit übertroffen werden.

Der zweite Beitrag beschreibt eine kritische Evaluierung verschiedener Lasso-Methoden, die *a-priori* bekannte strukturelle Informationen über die genetische Marker und den untersuchten Phänotypen benutzen. Wir bewerten die verschiedenen Ansätze auf Grund ihrer Vorhersagegenauigkeit auf simulierten Daten und auf Genexpressionsdaten in Hefe. Beide Experimente zeigen, dass Strukturinformationen nur dann helfen, wenn ihre Annahmen gerechtfertigt sind – sobald die Annahmen verletzt sind, hat die Zuhilfenahme der Strukturinformation den gegenteiligen Effekt. Um dem vorzubeugen, schlagen wir in unserem nächstem Beitrag vor, die Struktur zwischen den Phänotypen aus den Daten zu lernen.

Im dritten Beitrag stellen wir ein effizientes Rechenverfahren für Multi-Task Gauss-Prozesse auf, das sowohl die genetische Verwandtschaft zwischen den Phäno-

typen als auch die Verwandtschaft der Residuen lernt. Unser Inferenzverfahren zeichnet sich durch einen verminderten Laufzeit- und Speicherbedarf aus und ermöglicht uns damit, die gemeinsame Heritabilität von Phänotypen auf großen Datensätzen zu untersuchen. Das Kapitel wird durch zwei Versuchsstudien vervollständigt; einer genomweiten Assoziationsstudie von *Arabidopsis thaliana* und einer Genexpressionsanalyse in Hefe, die bestätigen dass die neue Methode bessere Vorhersagen liefert.

Die Vorteile der gemeinsamen Modellierung von Variablenselektion und Störfaktoren, sowie von Multi-Task Learning, werden in all unseren Versuchsreihen deutlich. Während sich unsere Experimente vor allem auf Anwendungen aus dem Bereich der Genomik konzentrieren, sind die von uns entwickelten Methoden jedoch allgemeingültig und können auch in anderen Feldern Anwendung finden.

# Abstract

A series of rapid changes has affected the field of genomics within the last few years. Advances in sequencing technology have led to an explosion of genotype and phenotype data, which allows us to explore the genetic architecture of complex traits at a finer scale than ever before. However, current approaches often fall short if the effect sizes of the markers are heterogenous, confounding factors harm the analysis, or dependencies between related phenotypes are erroneously ignored. While confounding factors may cause spurious associations between markers and the phenotype of interest, assuming homogenous effect sizes of all markers and independent phenotypes can lead to a loss of power in detecting markers with weak effect sizes. The aim of this thesis is to develop more accurate statistical methods that address these shortcomings while at the same time retain efficient computations.

Our first contribution is the LMM-Lasso, a hybrid model that combines the advantages of sparse linear models and linear mixed models. Our model dissects the phenotypic variability into components that either result from (1) individual genetic effects or (2) effects that are either caused by confounding, such as population structure, or by genetic effects that are too small to be traced back to single markers. Besides better interpretability of the selected markers, our method yields significantly more accurate phenotype predictions than standard sparse linear models.

Secondly, we provide a critical assessment of different Lasso methods that incorporate input and output structure that is known *a-priori*. Our results on simulated and gene expression level data in yeast indicate that methods that do not incorporate structural information work better than methods that make incorrect assumptions about the data. In order to avoid the use of incorrect prior knowledge, we suggest in our next contribution to rather learn the output structure from the data.

Thirdly, we present an efficient inference scheme for multi-task Gaussian processes that learns the genetic relatedness between the phenotypes, as well as the relatedness between the residuals. Our reformulation reduces the runtime and memory requirement significantly, making it possible to analyze the cross-heritability of large numbers of phenotypes and sample cohorts. We demonstrate the practical

use of our model in genome-wide association studies in *Arabidopsis thaliana* and an expression quantitate trait loci (eQTL) study in yeast.

Our experiments highlight the importance of feature selection, confounder correction and multi-task learning for applications in the field of genomics. Moreover, the methods we developed are general, and can thus also be used in other application domains.

# Acknowledgements

Above all, I would like to thank my supervisor Prof. Karsten Borgwardt for giving me guidance and feedback throughout my PhD. His encouragement and support helped me not only to accomplish this thesis, but also helped me grow on a personal level.

Prof. Bernhard Schölkopf and Prof. Detlef Weigel let me be a member of both of their departments, the Max Planck Institute for Intelligent Systems and the Institute for Developmental Biology. I enjoyed the continuous exchange of ideas with members of both fields and am thankful for this opportunity as well as for their scientific advice.

I am thankful to Prof. Bertram Müller-Myhsok for being the second examiner of this thesis. In addition, I also want to thank Prof. Daniel Huson for being willed to stand in as a first examiner in case needed.

I am also sincerely grateful to Oliver Stegle and Christoph Lippert for taking me under their wing. Both have been a great source of inspiration for me and our discussions have led to close collaborations. Connected to this, I would also like to thank Christoph Lippert and David Heckerman for offering me an enriching internship at Microsoft Research last summer.

I also want to thank my other colleagues in the AGKB group for scientific and non-scientific discussions. I enjoyed working with the old members of the group (Theofanis Karaletsos, Nino Shervashidze, Chloe Azencott, Limin Li, Niklas Kasenburg, Aasa Feragen) as much as with our new members (Damian Roqueiro, Felipe Llinares, Carl-Johann Simon-Gabriel). My special thanks goes to Dominik Grimm who has become a close friend over the last years. Furthermore, I would like to thank Paolo Casale, Philip Hennig, Recep Colak and Jonas Müller for collaborations and fruitful discussions, and Sonja Rümelin, Oliver Stegle, Meike Sprecher and Dominik Grimm for proof-reading parts of the thesis.

In addition, I want to thank the administrative and technical team, in particular Sebastian Stark, for their assistance. I acknowledge the Max Planck Society (Max Planck Gesellschaft) and the German Research Foundation (Deutsche Forschungsgemeinschaft) for funding.

Meike Sprecher deserves my greatest gratitude for having shared a flat with me during my PhD and the ups and downs that came with it; in this time, she always had an open ear for my sorrows and a warm soup for my soul.

Last, but not least I would like to thank my parents for their unconditional belief in me.

# Contents

# Chapter 1

# Introduction

Since the first draft of the human genome was sequenced in 2001 (Lander et al., 2001), tremendous progress has been made, allowing us to study the human genome at an unprecedented level of detail today. Large consortia, such as the HapMap Project (Frazer et al., 2007) or the 1000 Genomes Project (Abecasis et al., 2012), have helped in identifying positions in the genome that differ between different individuals, so-called single nucleotide polymorphisms (SNPs). In genome-wide association studies, SNPs are used as markers to detect associations between the trait under observation and a specific region of the genome by correlating genetic differences with phenotypic profiles in large cohorts of related or unrelated individuals. The phenotype can in principle be any quantifiable characteristics. Both, continuous or discrete measure are frequently being considered. For instance, if we are interested in finding markers, which are associated with a quantitative trait, we can test whether people carrying the one variant tend to have a larger phenotypic value than people carrying the other variant. If the phenotype is qualitative, such as carrying a particular disease, we can test whether one of the variants is more frequent in people that carry the disease. In contrast to traditional linkage analysis, which build upon pedigree information to identify regions in the genome that co-segregate with the disease of interest (Ott et al., 2011), association studies are easier to conduct (Nordborg and Weigel, 2008) and more powerful in detecting genes that only have a weak effect on the phenotype (Risch and Merikangas, 1996).

In the last seven years, genome-wide association studies have amongst others yielded valuable insights into the genetic architecture of global-level traits in plants (Atwell et al., 2010) and mice (Flint and Eskin, 2012), as well as the risks for common human diseases, such as inflammatory bowel disease (Khor et al., 2011), major depression (Kohli et al., 2011) and type 2 diabetes (Scott et al., 2007). So far, more

than 2000 robust associations with more than 300 complex diseases and traits have been reported (Manolio, 2013). In spite of these successes, our understanding of these phenotypes is far from being complete and we are just beginning to unravel the biological mechanisms underlying them. In particular, the associations uncovered to date only account for a small proportion of the phenotypic variances and the effect sizes of the individual markers are small (Maher, 2008). This can be shown for the instance of height, which has about 180 known associated variants, but they only account for $\sim 10\%$ of the phenotypic variance, although traditional methods estimate that the heritability of height is around 80% (Lango Allen et al., 2010; Visscher et al., 2008). Originally, heritability is estimated by comparing the resemblance of relatives by regressing the mean parental phenotype against its offspring. This measures the so-called narrow-sense heritability $h^2$ that is based on additive genetic effects only. It does not include the effects of other genetic factors, such as dominance effects, gene-gene interactions and gene-environment interactions (Visscher et al., 2008). The gap between $h^2$ and the heritability estimate $h^2_{\text{GWAS}}$, based on all significant markers only, is commonly referred to as *missing heritability*, and has received attention over the last few years (Manolio et al., 2009; Zaitlen and Kraft, 2012; Bloom et al., 2013). Many different explanations have been proposed, including structural variations, rare variants, and the joint contribution of many small additive effects (Eichler et al., 2010). In addition, epistatic effects can lead to an overestimate of the narrow-sense heritability (Zuk et al., 2012). Examples of these include amongst others a duplication of the *APP* locus, which causes autosomal dominant early-onset Alzheimer disease (Rovelet-Lecrux et al., 2006), and an epistatic interaction between HLA-B*51 and ERAP1, which contributes to the disease susceptibility for Behçet's disease (Kirino et al., 2013).

**Polygenic architecture** A major source of missing heritability can be contributed to common SNPs that have effect sizes too small for being able to be detected. Park et al. (2010) used data from existing genome-wide association studies to estimate the number of associated loci and the distribution of their effect sizes, to show that most complex phenotypes are likely to be controlled by thousands of susceptibility loci with effect sizes that are too small to be detected by current sample sizes. Purcell et al. (2009) and Stahl et al. (2012) computed an additive polygenic risk score based on SNPs that have a $p$-value below a certain significance threshold. By varying that threshold, they could show that including markers with a $p$-value above the genome-wide significance level improves the predictive performance on an independent test dataset, confirming the importance of yet undiscovered associations. At the same time, Yang et al. (2010) used a linear mixed model approach to show that much

larger parts of the phenotypic variance can be explained by rather using all common SNPs jointly than using the subset of significant SNPs only.

The consequences we can draw from that are twofold: First, there is hidden information contained in the SNP data that current methods cannot trace down to the single SNP level. Albeit larger sample sizes will help to gain additional power, it is unlikely that all risk variants can be revealed (Park et al., 2010).

Second, using SNP data only, the heritability estimates of Yang et al. (2010) provide an upper bound for what predictions with a linear model can achieve. Luckily, the development of new predictive models is still an area of active research (de los Campos et al., 2010). Most of the approaches are based on linear models, but they differ in the assumptions they make about the effect sizes. To give an example: Using a Gaussian prior leads to homogenous shrinkage across all SNPs, while sparse priors assign a larger weight to few markers and set the remaining weights to zero (Meuwissen et al., 2001). It depends on the genetic architecture of the phenotype at hand which prior should be chosen. If the effect sizes of the markers differ little and many variants contribute to the phenotype, the gaussian prior is well suited. However, if there are few causal variants that have a large effete on the phenotype, sparse priors are to be preferred. In practice, we rarely know which of the two scenarios we will encounter which is why we require approaches that can automatically decide between the two. For instance, Bloom et al. (2013) analyzed growth in yeast under multiple conditions and observed a various degree of trait complexity: the number of quantitative trait loci (QTL) and their effect sizes differed substantially between the traits.

**Confounding** The search of associated variants is often hindered by hidden factors that have an association with the phenotype as well as with the markers. If they are not corrected for, they can lead to spurious associations between the markers and the phenotype. One of the main sources of confounding in genome-wide association studies is population structure (Marchini et al., 2004). Consider therefor having a dataset consisting of two subpopulations. Both have a different genetic background and the disease is more prevalent in one of the two subpopulations. Then all SNPs, that are differentiated between the subpopulations, have an association with the phenotype. Lander and Schork (1994) have come up with an example illustrating this: Consider the phenotype "eating with chopsticks" for people living in the San Francisco area. The population of San Francisco harbors many people of East-Asian origin, people who are arguably better trained in eating with chopsticks. Variants that are more common in East Asian populations, such as the human leukocyte antigen complex, which plays an important role in immunology, would then appear

to be associated with the phenotype "eating with chopsticks", albeit there is no causal relationship between the chopstick skills of a person and its immune system.

In practice, population structure is difficult to avoid and even in a seemingly stratified sample the extent of hidden structure cannot be ignored (Newman et al., 2001). Models that account for the presence of such structure are routinely applied and have been shown to greatly reduce the impact of population stratification. For instance, EIGENSTRAT builds on the idea of extracting the major axes of population differentiation using a PCA decomposition of the genotype data (Price et al., 2006), and subsequently including them into the model as additional covariates. Linear mixed models (Kang et al., 2008, 2010; Lippert et al., 2011; Zhou and Stephens, 2012) provide for a more fine-grained control by modeling the contribution of population structure as a random effect, correcting also effectively for family structure and cryptic relatedness. The relatedness between the individuals is thereby estimated from the SNP data, by basically counting how many alleles the individuals share. The more alleles they share, the more similar the individuals are. In heritability estimation, the similarity between the samples is estimated in a similar fashion, albeit with a different intention: It is counted how many alleles the individuals share to get a proxy of how many *causal* alleles they have in common.

Listgarten et al. (2012) were one of the first that connected confounding in GWAS and heritability estimation, by proposing that parts of the confounding is due to shared genetic factors. In consequence, they propose to only include those SNPs in the background model, that are associated with the phenotype. While it has been shown that conditioning on a subset of relevant markers increases power if no population structure is present (Lippert et al., 2013), often all markers are needed to correct for population structure (Yang et al., 2014). In practice, it is common to have datasets that are subject to polygenic effects and population structure, demanding new methods that allow for both.

**From one to many: multidimensional phenotypes** Pleiotropy describes the effect if a variant or a gene is associated with several phenotypes (Mackay et al., 2009). The phenotypes can hereby be either distinct phenotypes (Lee et al., 2012) or the same phenotype measured under different conditions (Gagneur et al., 2013). The occurrences of pleiotropy are ubiquitous: Conservative estimates have shown that at least 16.9% of the genes and 4.6% of the SNPs that are associated with complex human diseases have pleiotropic effects (Sivakumaran et al., 2011). The protein *PTPN22* is, for instance, associated with several immune-related diseases, resulting in a higher probability that they share the same genetic pathway (Solovieff et al., 2013). While it is important to know which phenotypes share a common effect,

it is equally interesting to understand which variants only affect the phenotype under specific conditions. Amongst others, prominent examples can be found in the field of pharmacogenetics, in which the effectiveness of a drug can be harmed if a patient is carrying a specific mutation (Hunter, 2005). The risk of colorectal adenoma is for instance reduced for regular aspirin users that carry the slow allele of the enzyme UGT1A6, which is responsible for impaired aspirin metabolism (Bigler et al., 2001; Chan et al., 2005).

Going from one variant to multiple variants, we can also extend the heritability concept to multiple phenotypes. A way of doing that is examining the genetic cross-correlations between different phenotypes (Price et al., 2011; Vattikuti et al., 2012). Unfortunately, due to computational issues most analysis have been restricted to at most a handful of phenotypes so far. It will be interesting to see how these concepts can be carried over to high-dimensional phenotypes, such as gene expression levels. Genes that are co-expressed can be clustered into groups then, which are presumably involved in the same biological process. So far, existing algorithms that assemble co-expression networks, have mostly been based on empirical correlations only making them vulnerable to confounding factors (Mackay et al., 2009). Using the genetic correlations instead will allow us to study gene networks at a much finer level of detail, helping us to identify complex co-expression patterns between genes.

## 1.1 Contributions of this thesis

It has been the goal of this thesis to develop scalable algorithms that help to unravel the complex relationships between genotypes and phenotypes.

In the beginning of the genome-wide association era, most association tests were univariate tests, testing one marker at a time while ignoring all other effects. While this is computationally less demanding, these methods fall short if the genetic architecture of the phenotype is polygenic, i.e. many loci contribute jointly to the phenotypic variability. Instead, multivariate methods have the capacity to explicitly model the additive effects of multiple markers, leading to an increase in power for detecting weak effects. There exists a large body of work to address this issue, ranging from sparse models (Li et al., 2011; Hoggart et al., 2008; Carbonetto and Stephens, 2012) to variance component models (Kang et al., 2008, 2010; Yang et al., 2010). These algorithms differ substantially in the assumptions they make about the effect sizes, demanding a new class of models that requires less stringent assumptions and is capable to automatically decide which genetic architecture fits best. At the same time, it is unclear how to combine the benefits of multivariate sparse modeling with confounder correction, as population stratification is common.

While most analyses today are carried out on the single trait level, many datasets contain measurements of multiple correlated phenotypes. Body mass index, percent fat mass and waist circumference are, for example, different measurements that are all used to describe human obesity. All three of them are strongly correlated to each other (Shriner, 2012). By modeling them jointly, we cannot only increase the power to detect effects, but also gain insights into their interplay. However, most current approaches make either assumptions that are too simplistic, such that the noise is independent between the measured phenotypes (Stegle et al., 2011), or they do not scale up to dataset sizes of interest (Korte et al., 2012).

### 1.1.1 Confounder correction for Lasso methods

In Chapter 3, we propose the LMM-Lasso, a hybrid model that combines the advantages of linear mixed models and sparse regression models. Linear mixed model are often used to correct for confounding factors in genome-wide association studies, but also for heritability estimation and phenotype prediction, assuming that the causal variants can be best approximated by using all markers. Sparse models, in contrast, assume that only a few variants contribute to the phenotypic variability. Our approach combines the merits of both models. It allows a few outlier variants to have a large impact on the phenotype, while smaller effects, that cannot be traced back to individual markers, and environmental factors are modeled separately.

We demonstrate the practical use of LMM-Lasso in genome-wide association studies in *Arabidopsis thaliana* (Atwell et al., 2010) and linkage mapping in mouse (Valdar et al., 2006b), in which our method achieves significantly more accurate phenotype predictions than standard sparse models for 91% of the considered phenotypes. Enrichment of known candidate genes suggests that the individual associations retrieved by LMM-Lasso are likely to be genuine.

This work was done jointly with Christoph Lippert, Oliver Stegle and Karsten Borgwardt, and resulted in the publication

- Barbara Rakitsch, Christoph Lippert, Oliver Stegle, Karsten Borgwardt
  *A Lasso Multi-Marker Mixed Model for Association Mapping with Population Structure Correction*,
  **Bioinformatics 29 (2)**, 206-214.

  Barbara Rakitsch, Christoph Lippert, Oliver Stegle and Karsten Borgwardt conceived the method and designed the study. Barbara Rakitsch, Christoph Lippert and Oliver Stegle designed the experiments and analyzed the data. Barbara Rakitsch performed the experiments and wrote the source code. Bar-

bara Rakitsch, Christoph Lippert, Oliver Stegle and Karsten Borgwardt wrote the paper.

### 1.1.2 Incorporating structural information

Multi-trait models are being widely used to couple regressors that share the same underlying signal. By leveraging the data over multiple traits, a substantial power gain can be achieved in many cases (Obozinski et al., 2008; Kim and Xing, 2009). In addition, many existing methods aim to exploit prior knowledge about interactions of genetic loci in biological networks to reduce the search space of possible marker combinations (Li and Li, 2008; Azencott et al., 2013). In Chapter 4, we compare different Lasso methods that incorporate both, input and output structure.

Our experiments on simulated data and on an eQTL study in yeast (Smith and Kruglyak, 2008) suggest that structural information has to be used with care, as methods that do not incorporate structural information work better than methods that make incorrect assumptions about the data. We conduct two experiments on the yeast eQTL dataset, demonstrating that a loose coupling of the weight vectors across related phenotypes works best if the *true* genetic relatedness between the different phenotypes is not known.

This part of the thesis is based on unpublished work done in collaboration with Recep Colak and Karsten Borgwardt. The study was conceived and designed by all three authors jointly. Barbara Rakitsch developed the mathematical speed-ups. Recep Colak and Barbara Rakitsch performed the experiments and wrote the source code. Recep Colak, Barbara Rakitsch and Karsten Borgwardt wrote the manuscript.

### 1.1.3 Scalable multi-trait models

In Chapter 5, we propose a multi-trait model approach that circumvents the drawbacks of the approaches from the previous chapter by learning the coupling of the weight vectors and allowing for correlated residuals to account for hidden confounding. Models of this type have been used before (Henderson, 1984; Zhang, 2007; Korte et al., 2012), but we are the first ones to show that efficient inference of the model parameters and predictions are possible for that class of models.

We compare the predictive power of our approach against existing methods on synthetic data, gene expression levels of yeast (Smith and Kruglyak, 2008), and developmental phenotypes of *Arabidopsis thaliana* (Atwell et al., 2010). Our method outperforms its competitors on all three datasets.

This project was developed together with Christoph Lippert, Karsten Borgwardt and Oliver Stegle, and published in:

- Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt*, Oliver Stegle*
  *It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals*,
  **Neural Information Processing Systems (NIPS) 2013**, Lake Tahoe, USA.

  Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt and Oliver Stegle conceived the method. Barbara Rakitsch derived the mathematical tricks for the algorithm, performed the experiments and wrote the source code. All authors contributed in writing the manuscript.

We conclude in Chapter 6 by summarizing the individual contributions and providing an outlook to future research.

The experiments and the theoretical results from Chapter 3 and Chapter 5 are based on the publications mentioned above. We unified the notation in all chapters for better readability.

# Chapter 2

# Regression

So far, we have described the key challenges in genomics: Small sample sizes are confronted with a large number of genetic variants, complex genetic architectures and confounding factors, which may cause spurious associations between markers and phenotype. To advance the power of genome-wide association studies, it is irremissible to model these factors as accurate as possible.

Before discussing the contributions of this thesis to tackle these challenges, we here summarize the necessary background: Section 2.1 gives a brief introduction to linear models, and Section 2.2 extends these concepts to the non-linear case. It also defines the notation and terminology used in the remainder of this thesis.

## 2.1 Linear regression

In linear regression, we try to find a linear mapping between the continuous target variable $y$ and the features $\{x_1, \ldots, x_M\}$

$$y = \sum_{m=1}^{M} w_m x_m, \tag{2.1}$$

where $M$ denotes the number of features. The mapping is defined by the parametric weight vector $\boldsymbol{w} = \begin{pmatrix} w_1 & \ldots & w_M \end{pmatrix}^\top \in \mathbb{R}^M$. The weights $\boldsymbol{w}$ are unknown and must be inferred from the data. In statistical genetics, the features $\boldsymbol{x} = \begin{pmatrix} x_1 & \ldots & x_M \end{pmatrix}^\top \in \mathbb{R}^M$ typically correspond to genetic markers and the target variable $y$ denotes the phenotype of interest.

The model in (2.1) can be expressed in compact matrix notation, by stacking the $N$ observations into the vector $\boldsymbol{y} = \begin{pmatrix} y_1, \ldots & y_N \end{pmatrix}^\top$ and the features to the matrix

$\boldsymbol{X} = \left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\right)^\top \in \mathbb{R}^{N \times M}$. This allows us to rewrite the linear model as:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}. \tag{2.2}$$

In practice, the relationship between the features and the outcome is most often non-deterministic. Reasons for this can be manifold and include amongst others measurement noise, unmeasured causal processes, or non-linear relationships. We can account for this by adding a noise term in our model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}, \tag{2.3}$$

with $\mathbb{E}\left[\boldsymbol{\epsilon}\right] = \boldsymbol{0}$ and $\mathrm{Cov}\left[\boldsymbol{\epsilon}\right] = \sigma_\epsilon^2 \boldsymbol{I}$. Given a dataset $(\boldsymbol{X}, \boldsymbol{y})$, the goal of linear regression is to find a weight vector $\boldsymbol{w}$ that fits the data best. Depending on the assumptions we put on the weights and on the noise, we arrive at different estimators: In this chapter, we give a brief overview over the most important estimators and highlight some of their properties.

For a more detailed description, we refer the interested reader to (Bishop, 2006; Fahrmeir et al., 2009; Hastie et al., 2009). Bishop (2006) gives a probabilistic perspective, Hastie et al. (2009) represent the frequentist position, and Fahrmeir et al. (2009) provide the statistical framework of linear regression.

### 2.1.1 Least squares

The most common approach for fitting the weights $\boldsymbol{w}$ is to minimize the sum of the squared training error

$$\hat{\boldsymbol{w}}^{LS} = \arg\min_{\boldsymbol{w}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right) \tag{2.4}$$

Since the objective function is convex in $\boldsymbol{w}$, its global minimum can be found by setting the gradient to zero. The gradient of the sum of squares error is:

$$\nabla_{\boldsymbol{w}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right) = -2\boldsymbol{X}^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right) \tag{2.5}$$

Provided that the matrix $\boldsymbol{X}\boldsymbol{X}^\top$ is not singular, we can then obtain the estimator $\hat{\boldsymbol{w}}^{LS}$ by setting the gradient to zero and solving subsequently for $\hat{\boldsymbol{w}}^{LS}$:

$$\begin{aligned} & \boldsymbol{0} && = \boldsymbol{X}^\top \left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}_{LS}\right) \\ \Leftrightarrow \quad & \boldsymbol{X}^\top \boldsymbol{X}\hat{\boldsymbol{w}}^{LS} && = \boldsymbol{X}^\top \boldsymbol{y} && (2.6) \\ \Leftrightarrow \quad & \hat{\boldsymbol{w}}^{LS} && = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}. && (2.7) \end{aligned}$$

We can either computer $\hat{\boldsymbol{w}}^{LS}$ by solving the linear system (2.6) or by computing the inverse directly (2.7). Unless the matrix $\boldsymbol{X}\boldsymbol{X}^\top$ has a specific structure that can be exploited, either approach scales in $O(M^3)$ time.

Minimizing the least squares objective can also be motivated from the probabilistic perspective, assuming that the noise is Gaussian distributed. It is then easy to show that the maximum likelihood estimate $\hat{\boldsymbol{w}}^{ML}$ coincides with the least-squares estimator

$$\begin{aligned}\hat{\boldsymbol{w}}^{ML} &= \arg\max_{\boldsymbol{w}} \log \mathcal{N}\left(\boldsymbol{y} \left| \boldsymbol{X}\boldsymbol{w}, \sigma_e^2 \boldsymbol{I}\right.\right) \\ &= \arg\max_{\boldsymbol{w}} -\frac{1}{2\sigma_e^2} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right) \\ &= \arg\min_{\boldsymbol{w}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right) \\ &= \hat{\boldsymbol{w}}^{LS}. \end{aligned} \tag{2.8}$$

The least squares estimator has a number of appealing statistical properties, which we will study in the following. First, the least squares estimator is unbiased. That means that its expected value equals the true value

$$\mathbb{E}\left[\hat{\boldsymbol{w}}^{LS}\right] = \mathbb{E}\left[\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{y}\right] = \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\mathbb{E}\left[\boldsymbol{y}\right] = \left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{w} = \boldsymbol{w}. \tag{2.9}$$

Let $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ be the eigenvalue decomposition of $\boldsymbol{X}^\top\boldsymbol{X}$. The least squares estimator is the least certain in giving the directions which are described by the eigenvectors with small eigenvalues:

$$\begin{aligned}\mathrm{Cov}\left[\hat{\boldsymbol{w}}^{LS}\right] &= \mathbb{E}\left[\left(\hat{\boldsymbol{w}}^{LS} - \boldsymbol{w}\right)\left(\hat{\boldsymbol{w}}^{LS} - \boldsymbol{w}\right)^\top\right] \\ &= \mathbb{E}\left[\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)^\top\left(\hat{\boldsymbol{w}}^{LS} - \boldsymbol{w}\right)^\top\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\right] \\ &= \sigma_e^2\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1} \\ &= \sigma_e^2\left(\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1} \\ &= \sigma_e^2\boldsymbol{U}\boldsymbol{S}^{-1}\boldsymbol{U}^\top. \end{aligned} \tag{2.10}$$

**Geometric interpretation** The predicted outcome of the training data $\hat{\boldsymbol{y}}^{LS} = \boldsymbol{X}\hat{\boldsymbol{w}}^{LS}$ is a linear combination of the features $\boldsymbol{X}$ and lies thus in the subspace $\mathrm{span}\left(\boldsymbol{X}\right)$, spanned by the features $\boldsymbol{X}$. If we minimize the sum of squared distances between the observed and the predicted outcomes, we seek the predictor that

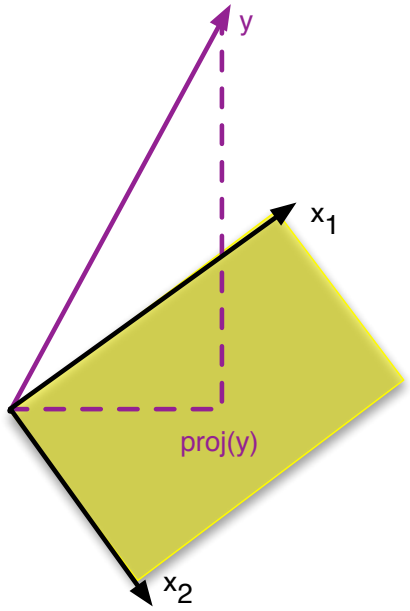



Figure 2.1: **A geometric interpretation of least squares.** Let the number of data points be $N = 3$ and the number of features be $M = 2$. The two feature vectors $(x_{11}, x_{21}, x_{31}), (x_{12}, x_{22}, x_{32})$ span a two-dimensional subspace in $\mathbb{R}^3$. The orthogonal projection of $\boldsymbol{y} \in \mathbb{R}^3$ onto the subspace is the least squares estimator of $\boldsymbol{y}$. Based on Hastie et al. (2009).

is closest to to true outcome and within the subspace span $(\boldsymbol{X})$. From a geometric perspective, that point is given by the orthogonal projection of $\boldsymbol{y}$ onto span $(\boldsymbol{X})$ (see Figure 2.1). This point is mathematically equivalent to the least squares estimator. If $\boldsymbol{y}$ lies inside the spanned subspace, the predicted outcome recovers the true outcome.

**Best linear unbiased estimator** In statistics, the least squares estimator is known to be the best linear unbiased estimator, as proven in the Gauss-Markov theorem (Fahrmeir et al., 2009). Assuming that the noise is independent and identically distributed, one can show that under all unbiased estimates, the least squares estimator has the smallest variance

$$\text{Var}\left[\hat{w}_m^{LS}\right] \leq \text{Var}\left[\hat{w}_m\right], \quad m = 1, \ldots, M. \tag{2.11}$$

Notably, that theorem holds not only for Gaussian noise, but for general noise distributions. In general, there exist estimators with lower variance, however this reduction of variance comes at the cost of biased estimates. Optimizing the trade-off between bias and variance is an important challenge in statistics and machine learning (Geman et al., 1992; Domingos, 2000). In the following, we present a selection of biased estimators which outperforms the least squares predictor in terms of interpretability

and predictive power. Intuitively, the reduction in variance is achieved by introducing regularization to shrink many of the coefficients towards zero, at the price of zero-biased solutions.

### 2.1.2 Ridge regression

One of the most widely used approaches to reduce the variance of the estimator is to penalize the $\ell_p$-norm of the regularizer:

$$\|\boldsymbol{w}\|_p = \sqrt[p]{\sum_{m=1}^{M} |w_m|^p}. \tag{2.12}$$

Depending on which $\ell_p$-norm is used, different estimators are preferred: Comparing the $\ell_1$ and $\ell_2$-norm, we can observe that the $\ell_2$-norm penalizes large entries stronger, leading to an estimator which tends to have only few large entries. Contrary, the $\ell_1$-norm punishes vectors with many small values more, resulting in an estimator which contains many very small or zero entries (Boyd and Vandenberghe, 2004). In ridge regression (Hoerl and Kennard, 1970), the squared $\ell_2$-norm is used:

$$\hat{\boldsymbol{w}}^{ridge} = \arg\min_{\boldsymbol{w}} \underbrace{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})}_{\text{error term}} + \lambda \underbrace{\|\boldsymbol{w}\|_2^2}_{\text{regularizer}}. \tag{2.13}$$

The parameter $\lambda$ determines the trade-off between fitting the data and regularization. The setting $\lambda = 0$ correspond to an unregularized model which is equivalent to the least squares regressor. However, if the dataset contains much more features than samples ($N \ll M$), the learned estimator tends to overfit the data, i.e. the model fit does not only capture the true signal but also the noise. This reduces the generalization performance to new data points (see also Figure 2.2). With increasing $\lambda$, less noise is explained by the estimator which in turn leads to better generalization behavior and a smaller variance of the estimator. However, an increase of $\lambda$ also shrinks the coefficients further towards zero leading to a larger bias. Figure 2.2 summarizes the relationship between overfitting, bias and variance on a synthetic dataset. In practice, the best value for $\lambda$ is not known and is often learnt by maximizing the out of sample predictive performance via cross-validation.

Solutions to the regularized least squares problem can be derived analogously to the standard least squares problem: By setting the gradient to zero, we can solve for the global minimum of the objective. The gradient with respect to the weight vector is given by

$$\nabla_{\boldsymbol{w}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \lambda \boldsymbol{w}^\top \boldsymbol{w} = -2\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + 2\lambda \boldsymbol{w} \tag{2.14}$$

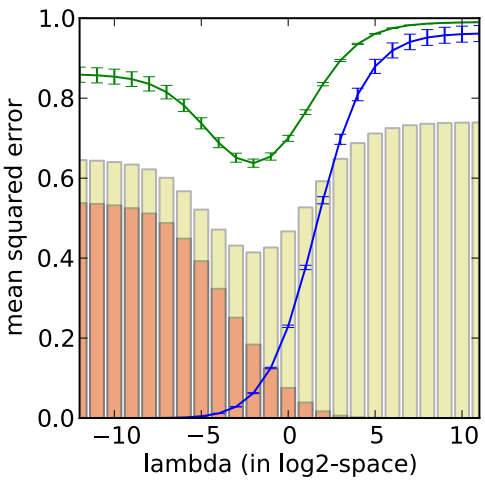



Figure 2.2: **Bias-Variance decomposition.** The training error (blue line) and test error (green line) are shown as a function of the regularization parameter $\lambda$. Standard errors are computed over 30 repetitions. The test error (green line) decomposes into the squared bias (yellow), the variance (red) and noise.

For each repetition, we draw $N = 200$ random points as training set. The target is determined by the function $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$ with signal-to-noise ratio of 0.8 and $M = 300$. The weight vector and the test set (200 samples) are fixed over all repetitions.

After setting it to zero, we can solve for $\hat{\boldsymbol{w}}^{ridge}$

$$\hat{\boldsymbol{w}}^{ridge} = \left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} \tag{2.15}$$

$$= \boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y}. \tag{2.16}$$

If the number of samples $N$ is larger than the number of features $M$, we can either solve for the estimator in feature-space (2.15) or by using one of the Searle identities (Petersen and Pedersen, 2012) in sample-space (2.16), leading to a runtime of $O\left(\min(M^3, N^3)\right)$.

Important insights into the regularization mechanism can be gained from employing the singular value decomposition of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ and looking at the predicted outcomes (as discussed in Hastie et al. (2009) and Murphy (2012))

$$\begin{aligned}
\hat{\boldsymbol{y}}^{ridge} &= \boldsymbol{X}\left[\boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y}\right] \\
&= \boldsymbol{U}\boldsymbol{S}^{\mathbf{2}}\boldsymbol{U}^\top \left(\boldsymbol{U}\boldsymbol{S}^2\boldsymbol{U}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y} \\
&= \boldsymbol{U}\boldsymbol{S}^2 \left(\boldsymbol{S}^2 + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{U}^\top \boldsymbol{y} \\
&= \boldsymbol{U}\hat{\boldsymbol{S}}\boldsymbol{U}^\top \boldsymbol{y},
\end{aligned} \tag{2.17}$$

where $\hat{\boldsymbol{S}}$ is a diagonal matrix with $\hat{S}_{jj} = S_{jj}^2 / \left(S_{jj}^2 + \lambda\right)$. The diagonal entries of $\hat{\boldsymbol{S}}$ lie between 0 if $S_{jj}^2 \ll \lambda$ and 1 if $S_{jj}^2 \gg \lambda$. The smaller therefore the $j$th eigenvalue is, the less the direction of the $j$th eigenvector is taken into account. In the last

section 2.1.1, we showed that the least squares estimator is the least certain for the directions corresponding to small eigenvalues. The ridge regression estimator shrinks these directions resulting in a more stable solution.

### 2.1.3 Bayesian linear regression

It turns out that the ridge regression objective is equivalent to a MAP solution of linear regression with a Gaussian prior on the regression weights:

$$\boldsymbol{w} \sim \mathcal{N}\left(\boldsymbol{w} \left| \mathbf{0}, \sigma_g^2 \boldsymbol{I}\right.\right). \tag{2.18}$$

The hyperparameter $\sigma_g^2$ controls thereby the width of the Gaussian. We can obtain the posterior distribution of the weight vector by applying Bayes theorem

$$\begin{aligned} p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \sigma_g^2, \sigma_e^2) &\propto p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \sigma_e^2) p(\boldsymbol{w}|\sigma_g^2) \\ &= \mathcal{N}\left(\boldsymbol{y} \left| \boldsymbol{X}\boldsymbol{w}, \sigma_e^2 \boldsymbol{I}\right.\right) \mathcal{N}\left(\boldsymbol{w} \left| \mathbf{0}, \sigma_g^2 \boldsymbol{I}\right.\right) \\ &= \mathcal{N}\left(\boldsymbol{w} \left| \boldsymbol{\Sigma}^{-1}\boldsymbol{X}^\top \boldsymbol{y}, \sigma_e^2 \boldsymbol{\Sigma}^{-1}\right.\right), \end{aligned} \tag{2.19}$$

where $\boldsymbol{\Sigma}$ is defined as $\boldsymbol{X}^\top \boldsymbol{X} + \frac{\sigma_e^2}{\sigma_g^2}\boldsymbol{I}$. Since the Gaussian distribution is symmetric, the mean estimator coincides with the maximum a posteriori estimate, which in turn is equivalent to the ridge regression estimator

$$\begin{aligned} &\arg\max_{\boldsymbol{w}} \log \mathcal{N}\left(\boldsymbol{y} \left| \boldsymbol{X}\boldsymbol{w}, \sigma_e^2 \boldsymbol{I}\right.\right) + \log \mathcal{N}\left(\boldsymbol{w} \left| \mathbf{0}, \sigma_g^2 \boldsymbol{I}\right.\right) \\ &= \arg\max_{\boldsymbol{w}} -\frac{1}{2\sigma_e^2}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right) - \frac{1}{2\sigma_g^2}\boldsymbol{w}^\top \boldsymbol{w} \\ &= \arg\min_{\boldsymbol{w}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right) + \frac{\sigma_e^2}{\sigma_g^2}\boldsymbol{w}^\top \boldsymbol{w}. \end{aligned} \tag{2.20}$$

The tradeoff parameter $\lambda$ can thereby be reinterpreted as the ratio between the noise level $\sigma_e^2$ and the prior strength $\sigma_g^2$. If the uncertainty in $\boldsymbol{w}$ is large, it makes sense to not work with a point estimate, like the maximum a posterior estimate, but to marginalize over all possible weights.

The uncertainty of the weight vector $\boldsymbol{w}$ can then be retrained when predicting the outcome for some new features $\boldsymbol{X}^\star$:

$$\begin{aligned} p(\boldsymbol{y}^\star|\boldsymbol{X}^\star, \boldsymbol{X}, \boldsymbol{y}, \sigma_g^2, \sigma_e^2) &= \int p(\boldsymbol{y}^\star|\boldsymbol{X}^\star, \boldsymbol{w}, \sigma_e^2) p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, \sigma_g^2, \sigma_e^2) \boldsymbol{dw} \\ &= \int \mathcal{N}\left(\boldsymbol{y}^\star \left| \boldsymbol{X}^\star \boldsymbol{w}, \sigma_e^2 \boldsymbol{I}\right.\right) \mathcal{N}\left(\boldsymbol{w} \left| \boldsymbol{\Sigma}^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \sigma_e^2 \boldsymbol{\Sigma}^{-1}\right.\right) \\ &= \mathcal{N}\left(\boldsymbol{y}^\star \left| \boldsymbol{X}^\star \boldsymbol{\Sigma}^{-1} \boldsymbol{X} \boldsymbol{y}, \sigma_e^2 \left(\boldsymbol{I} + \boldsymbol{X}^\star \boldsymbol{\Sigma}^{-1} \boldsymbol{X}^{\star\top}\right)\right.\right) \\ &= \mathcal{N}\left(\boldsymbol{y}^\star \left| \boldsymbol{X}^\star \boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top + \frac{\sigma_e^2}{\sigma_g^2}\boldsymbol{I}\right)^{-1} \boldsymbol{y} \right.\right. \\ &\qquad \left.\left. , \sigma_e^2 \boldsymbol{I} + \sigma_g^2 \boldsymbol{X}^\star \boldsymbol{X}^{\star\top} \right.\right. \\ &\qquad \left.\left. -\sigma_g^2 \boldsymbol{X}^\star \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X}^\top + \frac{\sigma_e^2}{\sigma_g^2}\boldsymbol{I})^{-1} \boldsymbol{X}\boldsymbol{X}^{\star\top} \right.\right), \end{aligned} \tag{2.21}$$

where the last equality can be derived by applying the Woodbury identity (A.28). We note that the mean predictions of Bayesian linear regression is consistent with the predictions obtained by ridge regression (2.15). The hyperparameters $\sigma_g^2, \sigma_e^2$ are most often not known beforehand and can be inferred from the data via cross-validation or by optimizing the evidence

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}, \sigma_e^2, \sigma_g^2) &= \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \sigma_e^2) p(\boldsymbol{w}|\sigma_g^2) \boldsymbol{dw} \\ &= \int \mathcal{N}\left(\boldsymbol{y} \left| \boldsymbol{X}\boldsymbol{w}, \sigma_e^2 \boldsymbol{I}\right.\right) \mathcal{N}\left(\boldsymbol{w} \left| \boldsymbol{0}, \sigma_g^2 \boldsymbol{I}\right.\right) \boldsymbol{dw} \\ &= \mathcal{N}\left(\boldsymbol{y} \left| \boldsymbol{0}, \sigma_g^2 \boldsymbol{X}\boldsymbol{X}^\top + \sigma_e^2 \boldsymbol{I}\right.\right). \end{aligned} \tag{2.22}$$

When maximizing the marginal likelihood, we can either work with the gradient or do an exhaustive search over all $(\sigma_g^2, \sigma_e^2)$ combinations. Since the likelihood function is non-convex, gradient-based optimization can be stuck in a local optimum. On the contrary, scanning all $(\sigma_g^2, \sigma_e^2)$ combinations ensures that we find the global optimum. If done naively, evaluating the likelihood for the complete $(\sigma_g^2, \sigma_e^2)$ grid is slow since we have to invert the covariance matrix for each likelihood evaluation. However, by using the reparameterization $(\sigma_g^2, \lambda = \frac{\sigma_e^2}{\sigma_g^2})$ and employing the eigenvalue decomposition of

$\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$, computing the likelihood can be done efficiently

$$\begin{aligned}
\log p(\boldsymbol{y}|\boldsymbol{X}, \sigma_e^2, \sigma_g^2) &\propto -N \log 2\pi - \log \left|\sigma_g^2 \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}\right)\right| - \boldsymbol{y}^\top \left[\sigma_g^2 \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}\right)\right]^{-1} \boldsymbol{y} \\
&\propto -N \log 2\pi\sigma_g^2 - \log \left|\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top + \lambda \boldsymbol{I}\right| - \frac{1}{\sigma_g^2} \boldsymbol{y}^\top \left(\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y} \\
&\propto -N \log 2\pi\sigma_g^2 - \log \left|\boldsymbol{U}\left(\boldsymbol{S} + \lambda \boldsymbol{I}\right)\boldsymbol{U}^\top\right| - \frac{1}{\sigma_g^2} \boldsymbol{y}^\top \left[\boldsymbol{U}\left(\boldsymbol{S} + \lambda \boldsymbol{I}\right)\boldsymbol{U}^\top\right]^{-1} \boldsymbol{y} \\
&\propto -N \log 2\pi\sigma_g^2 - \log \left|\boldsymbol{S} + \lambda \boldsymbol{I}\right| - \frac{1}{\sigma_g^2} \left(\boldsymbol{U}^\top \boldsymbol{y}\right)^\top \left(\boldsymbol{S} + \lambda \boldsymbol{I}\right)^{-1} \left(\boldsymbol{U}^\top \boldsymbol{y}\right) \\
&\propto -N \log 2\pi\sigma_g^2 - \sum_{i=1}^{m} \log \left(S_{ii} + \lambda\right) - \frac{1}{\sigma_g^2} \sum_{i=1}^{m} \frac{1}{S_{ii} + \lambda} \left(\boldsymbol{U}^\top \boldsymbol{y}\right)_i^2
\end{aligned} \tag{2.23}$$

after having once computed the eigenvalue decomposition (Lippert et al., 2011). A full Bayesian treatment, in which we marginalize over the hyperparameters, is possible as well (Murray and Adams, 2010). However, it comes at the cost of a substantial increase in computation time, since the resulting integral is no longer analytically tractable.

### 2.1.4 Linear mixed models

A linear mixed model is a linear model containining a fixed effect $\boldsymbol{w}$ that has an unobserved, but fixed value and a random effect $\boldsymbol{u}$ which is an unobserved random variable drawn from a normal distribution $\mathcal{N}\left(\boldsymbol{u}\,|\boldsymbol{0}, \boldsymbol{R}\right)$. The feature matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times Q}$ describes the features belonging to the random effect $\boldsymbol{u} \in \mathbb{R}^Q$:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon}. \tag{2.24}$$

From the Bayesian perspective, we can interpret the linear mixed model as a linear model in which the weights come from two different prior distributions: $\boldsymbol{w}$ has an improper uniform prior and $\boldsymbol{u}$ a Gaussian prior (Robinson, 1991).

Linear mixed models are often used in genome-wide association studies as they can account for population stratification (Yu et al., 2006; Kang et al., 2008; Zhang et al., 2010; Kang et al., 2010; Lippert et al., 2011). In a nutshell, the significance of a single SNP is determined by comparing the fit between the model having the marker included as a fixed effect and the the model having the marker excluded. All other observed covariates, such as age or gender, are included as fixed effects as well. Population stratification cannot be directly observed and is treated as a

random effect. Its covariance matrix can be estimated from the genetic markers, and describes the genetic similarity between the samples.

If we are interested in the estimates of the fixed effects $\hat{\boldsymbol{w}}$ and of the random effects $\hat{\boldsymbol{u}}$, we can maximize over the posterior of our model:

$$\begin{aligned}&\arg\max_{\boldsymbol{w},\boldsymbol{u}} \log \mathcal{N}\left(\boldsymbol{y} \left| \boldsymbol{X}\boldsymbol{w} + \boldsymbol{Z}\boldsymbol{u}, \sigma_e^2 \boldsymbol{I}\right.\right) + \log \mathcal{N}\left(\boldsymbol{u} \left| \boldsymbol{0}, \boldsymbol{R}\right.\right) \\ &= \arg\min_{\boldsymbol{w},\boldsymbol{u}} \frac{1}{\sigma_e^2} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - \boldsymbol{Z}\boldsymbol{u}\right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - \boldsymbol{Z}\boldsymbol{u}\right) + \boldsymbol{u}^\top \boldsymbol{R}^{-1} \boldsymbol{u} \end{aligned} \tag{2.25}$$

Evaluating the gradient and setting it to zero, as done in Section 2.1.2, we arrive at the so-called mixed model equations (Henderson, 1950; Henderson et al., 1959)

$$\begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{X} & \boldsymbol{X}^\top \boldsymbol{Z} \\ \boldsymbol{Z}^\top \boldsymbol{X} & \boldsymbol{Z}^\top \boldsymbol{Z} + \sigma_e^2 \boldsymbol{R}^{-1} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{w}} \\ \hat{\boldsymbol{u}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{Z}^\top \boldsymbol{y} \end{pmatrix} \tag{2.26}$$

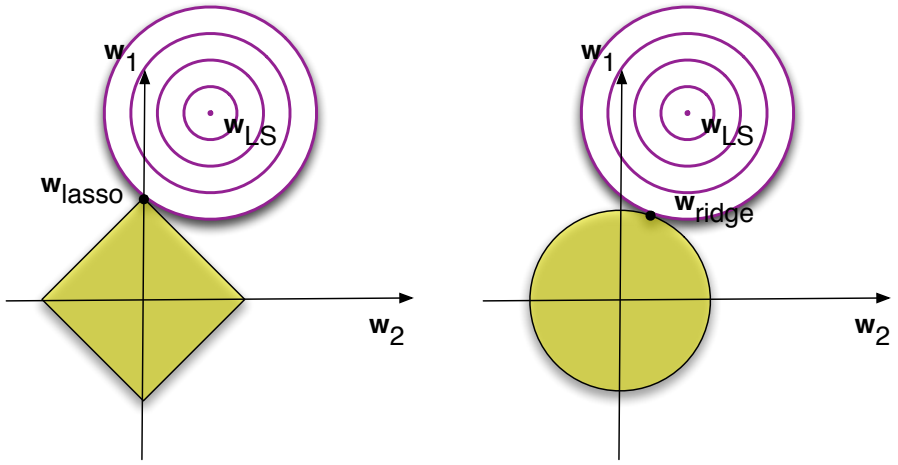
The solution of the linear system is also known as the BLUP estimate, whereby BLUP stands for best linear unbiased predictions (Goldberger, 1962; Robinson, 1991). By fixing one of the weight vectors in (2.25), we see the close connections between linear mixed models and the approaches discussed before: when keeping $\boldsymbol{w}$ fixed, the estimate of $\boldsymbol{u}$ is mathematically equivalent to the MAP estimate of Bayesian linear regression on the residual outcome $\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}$ and under the non-isotropic Gaussian prior $\mathcal{N}\left(\boldsymbol{0}, \boldsymbol{R}\right)$. When fixing $\boldsymbol{u}$, $\boldsymbol{w}$ is equivalent to the least squares estimator on the residuals $\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{u}$. When we are only interested in the fixed effects, we can also marginalize over the random effects

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{R}, \sigma_e^2) = \mathcal{N}\left(\boldsymbol{y} \left| \boldsymbol{X}\boldsymbol{w}, \boldsymbol{Z}\boldsymbol{R}\boldsymbol{Z}^\top + \sigma_e^2 \boldsymbol{I}\right.\right), \tag{2.27}$$

similarly to what we have done in Bayesian linear regression (2.23). Hyperparameters can be optimized analogously to what we have discussed there.

### 2.1.5 Lasso methods

In statistical genetics, we measure up to millions of SNPs and want to identify a small subset of them that play an important role in understanding the biological mechanisms underlying the phenotype. There is a substantial body of literature concerning feature selection in general (Buehlmann and van de Geer, 2011; Miller, 2002; O'Hara and Sillanpaa, 2009) and tailored to genome-wide association studies, in particular (Hoggart et al., 2008; He and Lin, 2011; Carbonetto and Stephens, 2012).




(a) Lasso (b) Ridge Regression

Figure 2.3: **Contours of the error and regularization function.** We show the contour plots of the error function for Lasso (left) and Ridge regression (right) in pink. Points along one contour line have the same function value. We restrict our attention to the weight vectors for which the regularization function is smaller than a certain threshold $\alpha$ (yellow-shaded area). The optimal solution is found when the contours first hit the constraint region. For the Lasso, the solution is sparse (it lies on the axis), while for the Ridge it is not. Adopted from Tibshirani (1994).

Finding a subset of relevant features is in general an NP-hard problem, and although exact algorithms exist, they do not scale up to problems of our size (Hastie et al., 2009). We concentrate here on the so called Lasso methods (Tibshirani, 1994), which employ an $\ell_1$-norm as regularization term:

$$\hat{\boldsymbol{w}}_{\text{lasso}} = \arg\min_{\boldsymbol{w}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1. \tag{2.28}$$

The name Lasso is a short for "least absolute selection and shrinkage operator". The first term "least absolute selection" means that the Lasso selects a subset of variables (for a graphical explanation see 2.3). The second term "shrinkage operator" means that the coefficients of the selected variables are shrunk towards zero. For understanding the Lasso, it is helpful to look at its subderivative as done in Murphy

(2012)

$$
\begin{aligned}
&\frac{\partial}{\partial w_k}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)^\top\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)+\lambda\sum_{m=1}^{M}|w_m| \\
=&\frac{\partial}{\partial w_k}\sum_{n=1}^{N}\left[\underbrace{y_n-\sum_{m\neq k}X_{nm}w_m}_{\tilde{y}_n}-X_{nk}w_k\right]^2+\lambda\,|w_k| \\
=&\sum_{n=1}^{N}\left[-2\tilde{y}_nX_{nk}+2\left(X_{nk}\right)^2w_k\right]+\frac{\partial}{\partial w_k}\lambda\,|w_k| \\
=&\underbrace{-2\sum_{n=1}^{N}\tilde{y}_nX_{nk}}_{c_k}+\underbrace{2\sum_{n=1}^{N}\left(X_{nk}\right)^2}_{a_k}\cdot w_k+\lambda\begin{cases}\{-1\} & \text{if } w_k<0\\ [-1,+1] & \text{if } w_k=0\\ \{+1\} & \text{if } w_k>0\end{cases}
\end{aligned}
\tag{2.29}
$$

We then examine if zero is contained in the subdifferential:

$$
0\in -c_k+a_k\hat{w}_k+\begin{cases}\{-\lambda\} & \text{if } \hat{w}_k<0\\ [-\lambda,+\lambda] & \text{if } \hat{w}_k=0\\ \{+\lambda\} & \text{if } \hat{w}_k>0\end{cases}
\tag{2.30}
$$

If $c_k<-\lambda$, then $-c_k-\lambda>0$. Since $a_k>0$ that implicates that $\hat{w}_k<0$ and the optimum is found at $\hat{w}_k=\frac{c_k+\lambda}{a_k}$. The case $c_k>\lambda$ is symmetric and leads to $\hat{w}_k=\frac{c_k-\lambda}{a_k}>0$. If $-\lambda<c_k<\lambda$, then the subdifferental is zero at $\hat{w}_k=0$. As $\frac{c_k}{a_k}$ is the least squares fit, we can observe that for $|c_k|>\lambda$, the coefficient is shrunk by the factor $\frac{\lambda}{a_k}$, while for$|c_k|<\lambda$, the feature is not selected.

In contrast to the other estimators we have seen so far, there is neither a closed form solution for the Lasso estimator, nor can we use a gradient-based solver since the regularization term is not differentiable at zero. Nevertheless, the development of efficient Lasso solvers has been an active area of research over the last years, and different algorithms, including coordinate descent (Fu, 1998), stochastic gradient (Shalev-Shwartz and Tewari, 2009) or interior point methods (Kim et al., 2007) have been established that scale up to reasonable dataset sizes.

Lasso approaches have also been considered under the name Basis Pursuit in signal processing (Chen et al., 1998). They can also be derived from a Bayesian perspective when using a Laplacian prior and solving for the maximum a posteriori

estimate (Park and Casella, 2008). However, a fully Bayesian treatment has to be treated with care: In contrast to the MAP solution, the posterior mean, as well as the samples drawn from the posterior, are non-sparse.

## 2.2 Gaussian processes

When we discussed Bayesian linear regression in Section 2.1.3, we observed that predictions (2.21) could be either evaluated in feature or in sample space. In feature space, we needed to compute the weight vector $\boldsymbol{w}$ directly, whereas in sample space, we solely worked with the scalar product between the inputs $\boldsymbol{X}\boldsymbol{X}^\top$. More generally, it is often advantageous to not work with the raw features directly, but to map them into a higher non-linear space $\boldsymbol{\Phi}(\boldsymbol{X})$. Then, as long as we can efficiently compute the kernel $\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\Phi}(\boldsymbol{X})^\top$, we do not have to explicitly map the raw features into the new space. In fact, we do not even have to know the feature space as long as we can prove that it exists. This is commonly known as the kernel trick (Scholkopf and Smola, 2001), which is one of the main ideas behind Gaussian processes.

An excellent overview of Gaussian processes is given for instance by Barber (2012) or MacKay (1998) and an in-depth description by Rasmussen and Williams (2005).

### 2.2.1 Prior on functions

In Rasmussen and Williams (2005), a Gaussian process is a distribution over functions $f$, defined by the mean function $m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})]$ and the covariance function $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}[f(\boldsymbol{x})])(f(\boldsymbol{x}') - \mathbb{E}[f(\boldsymbol{x}')])]$

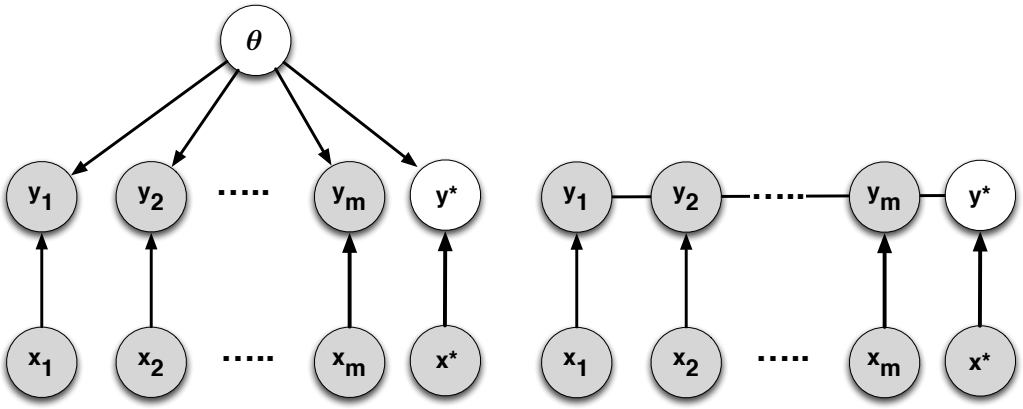
$$f \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')). \tag{2.31}$$

Intuitively, a Gaussian process can be regarded as the generalization of a Gaussian distribution to infinite many samples. For any finite dataset $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, we can marginalize out all the infinitely many unobserved samples: the corresponding function values $(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))$ follow then a joint Gaussian distribution.

The definition of the covariance function implicates that, for any input data $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, the covariance matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, defined by

$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2.32}$$

must be positive semidefinite, that is $\boldsymbol{x}\boldsymbol{K}\boldsymbol{x}^\top \geq 0$ for all $\boldsymbol{x}$.



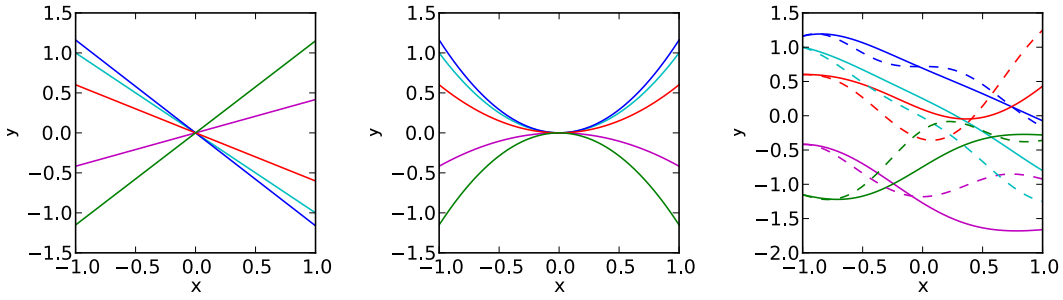

(a) Parametric Model (b) Nonparametric Model

Figure 2.4: **Difference between parametric and nonparametric models.** Graphical presentation of a parametric model (left) and of a nonparametric model (right). Given the parameters, predictions are independent of the training data in parametric methods. In nonparametric methods, the dependencies cannot be resolved. Adopted from Barber (2012).

**Nonparametric models** Gaussian processes are a prominent member of the class of nonparametric methods (Ghahramani, 2013). One way to distinguish parametric models from non-parametric models can be obtained by looking at the number of parameters: parametric methods have a limited number of parameters, while nonparametric methods can have infinitely many. Another way of distinguishing can be obtained by looking at their predicting behavior. For parametric models, the predictive distribution of a new data point $\boldsymbol{x}^\star$ is independent of the training data $(\boldsymbol{X}, \boldsymbol{y})$, once the parameters $\boldsymbol{\theta}$ are known

$$p(\boldsymbol{y}^\star|\boldsymbol{x}^\star, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}) = p(\boldsymbol{y}^\star|\boldsymbol{x}^\star, \boldsymbol{\theta}). \tag{2.33}$$

In contrast, non-parametric models have no parametric representation, and we need the complete training data to make further predictions (see also Figure 2.4). While this comes at the cost of a higher memory requirement, it also implies more flexibility: The complexity of the model is not bound by the size of the parametric vector, but can grow with the dataset size (Ghahramani, 2013).




(a) linear (b) polynomial (c) squared exponential

Figure 2.5: **Samples drawn from a Gaussian process with different covariance functions.** From left to right: We used a linear ($\sigma^2 = 1$), polynomial ($\sigma^2 = 1, c = 0, d = 2$) and squared exponential covariance ($\sigma^2 = 1, l^2 = 1$) function. To demonstrate the wiggling effect of $l^2$, we also show the squared exponential covariance with $l^2 = 0.5$ (dashed lines). The mean function was set to zero in all three experiments.

### 2.2.2 Covariance functions

The covariance of the outcomes $\boldsymbol{y}$ and $\boldsymbol{y'}$ is defined by its features and the choice of the kernel:

$$\text{cov}(\boldsymbol{y}, \boldsymbol{y}') = k(\boldsymbol{x}, \boldsymbol{x}'). \tag{2.34}$$

The kernel measures the similarity between the features $\boldsymbol{x}$ and $\boldsymbol{x}'$: The larger the similarity is, the more the two points covary. In the following, we will first give a few examples of commonly used kernel functions and then show how to create new kernel functions out of old ones.

The linear kernel is defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \, \boldsymbol{x}\boldsymbol{x}'^{\top}, \tag{2.35}$$

with $\sigma^2$ as a scaling factor. In fact, Gaussian processes with a linear kernel and a zero mean function $\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{0}$ are equivalent to Bayesian linear regression (see Section 2.1.3). As we showed there using linear algebra, predictions (2.21) and the evidence (2.22) can be expressed via the kernel $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^{\top}$ only. There is also a close connection between linear mixed models (see Section 2.1.4) and Gaussian processes:

After marginalizing out the random effects, a linear mixed model can be seen as a Gaussian process with the linear mean function $\boldsymbol{m}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{w}$ and the covariance matrix $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{R}\boldsymbol{Z}^\top$.

The polynomial kernel is defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \left( \boldsymbol{x}\boldsymbol{x}'^\top + c \right)^d, \tag{2.36}$$

where the scaling factor $\sigma^2$, the constant $c$ and the degree $d$ of the polynomial are hyperparameters. If $c = 0$ and $d = 2$, the feature space consists of all pairs of features $\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{x} \otimes \boldsymbol{x} \in \mathbb{R}^{M^2}$. By using the kernel trick, we can reduce the runtime for computing the covariance matrix from $O(M^2N^2)$ to $O(MN^2)$:

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') &= (\boldsymbol{x} \otimes \boldsymbol{x}) \left( \boldsymbol{x}' \otimes \boldsymbol{x}' \right)^\top \\ &= \left( \boldsymbol{x}\boldsymbol{x}'^\top \right) \otimes \left( \boldsymbol{x}\boldsymbol{x}'^\top \right) \\ &= \left( \boldsymbol{x}\boldsymbol{x}'^\top \right)^2 \end{aligned}$$

The squared exponential kernel is given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left( -\frac{(\boldsymbol{x} - \boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')}{2l^2} \right) \tag{2.37}$$

where $\sigma^2$ is a scaling factor and $l^2$ the lengthscale of the kernel. It can serve as an example of a stationary covariance function since the covariance function can be expressed as a function of the distance of the two inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$. Samples from the different covariance functions are shown in Figure 2.5.

A straight-forward way to create new covariance functions is to combine already existing ones (Scholkopf and Smola, 2001). Let $k_1, k_2$ be two valid covariance functions. Then the sum of the two

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}') \tag{2.38}$$

is again a valid covariance function. The same holds for the pointwise product

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') \cdot k_2(\boldsymbol{x}, \boldsymbol{x}'), \tag{2.39}$$

the Kronecker product

$$k\left( \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{pmatrix}, \begin{pmatrix} \boldsymbol{x}' \\ \boldsymbol{z}' \end{pmatrix} \right) = k_1(\boldsymbol{x}, \boldsymbol{x}') \cdot k_2(\boldsymbol{z}, \boldsymbol{z}'), \tag{2.40}$$

and the sum of two covariance functions

$$k\left(\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{pmatrix},\begin{pmatrix}\boldsymbol{x}'\\ \boldsymbol{z}'\end{pmatrix}\right)=k_1(\boldsymbol{x},\boldsymbol{x}')+k_2(\boldsymbol{z},\boldsymbol{z}'). \tag{2.41}$$

The Kronecker product plays thereby a prominent role within the different combinators as it can be either used to speed up computations (Wilson et al., 2013) or extend the Gaussian process framework to multitask learning (Bonilla et al., 2008).

It is important to decide carefully which covariance function to use. If the function to be learnt is not captured by the class of functions the covariance function can present, the performance of the Gaussian process, as of any other kernel-based method, will be poor. Fortunately, the design of a new expressive covariance functions is still an active area of research (Wilson et al., 2013), as is the extension to non-vectorial inputs (Sonnenburg et al., 2007; Feragen et al., 2013).

### 2.2.3 Predictions

So far, we have discussed different priors for Gaussian processes. In the following, we will investigate how observations change our beliefs over the function $f$. For that we assume a given training set $\boldsymbol{X}\in\mathbb{R}^{N\times M}, \boldsymbol{f}\in\mathbb{R}^N$ and a test data set for which we want to make predictions $\boldsymbol{X}^\star\in\mathbb{R}^{N^\star\times M}$. We then first note the joint probability over the training and testing points

$$\begin{pmatrix}\boldsymbol{f}\\ \boldsymbol{f}^\star\end{pmatrix}\sim\mathcal{N}\left(\begin{pmatrix}\boldsymbol{m}(\boldsymbol{X})\\ \boldsymbol{m}(\boldsymbol{X}^\star)\end{pmatrix},\begin{pmatrix}\boldsymbol{K} & \boldsymbol{K}^{\star\top}\\ \boldsymbol{K}^\star & \boldsymbol{K}^{\star\star}\end{pmatrix}\right), \tag{2.42}$$

where $\boldsymbol{K}^\star$ denotes the covariance matrix between the test and the training instances, and $\boldsymbol{K}^{\star\star}$ the covariance matrix between the test instances.

By conditioning on the training data, we yield a posterior distribution over the function values of the test dataset

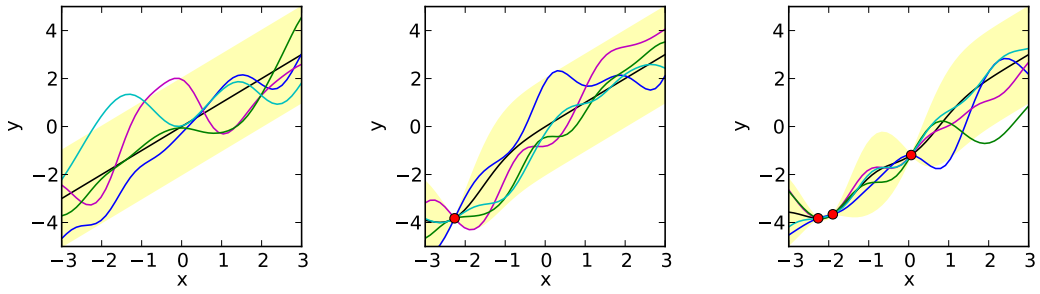
$$p(\boldsymbol{f}^\star|\boldsymbol{X}^\star,\boldsymbol{X},\boldsymbol{f})=\mathcal{N}\left(\boldsymbol{f}^\star\,|\boldsymbol{m}^\star,\boldsymbol{V}^\star\right), \tag{2.43}$$

where

$$\boldsymbol{m}^\star=\boldsymbol{m}(\boldsymbol{X}^\star)+\boldsymbol{K}^\star\boldsymbol{K}^{-1}\left(\boldsymbol{f}-\boldsymbol{m}(\boldsymbol{X})\right) \tag{2.44}$$

$$\boldsymbol{V}^\star=\boldsymbol{K}^{\star\star}-\boldsymbol{K}^\star\boldsymbol{K}^{-1}\boldsymbol{K}^{\star\top}. \tag{2.45}$$

By discussing the two extreme cases, we are able to gain important insights:




(a) No observation (b) 1 observation (c) 3 observations

Figure 2.6: **Drawing functions from the posterior.** We used a Gaussian process with mean function $m(x) = x$ and squared exponential covariance function ($\sigma^2 = 1$, and $l^2 = 1$). In the first plot from the left, we draw samples from the prior. In the other two plots, we draw samples from the posterior after having made one observation (middle) and three observations (right plot). Observations are marked as red dots. The black line depicts the mean predictions. The yellow area contains all predictions within two standard derivations from the mean. Adopted from Rasmussen and Williams (2005).

1. If the training data points are equal to the test points, i.e. $\boldsymbol{X}^\star = \boldsymbol{X}$, the mean predictions collapse to the observed function values: all functions drawn from the posterior pass through the observed data points (see also Figure 2.6).

2. If the test points are not similar to the training points, $\boldsymbol{K}^\star$ approaches zero, which results in predictions that are close to the ones obtained by the prior. That also makes intuitively sense: We cannot learn anything about the function values of $\boldsymbol{X}^\star$, as long as we have not yet observed any similar inputs.

In reality, we rarely observe the function values directly. Instead, they are usually perturbed by noise:

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon} \tag{2.46}$$

Assuming that the noise is Gaussian, $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_e^2 \boldsymbol{I})$, we can marginalize over the noise, leading to a covariance matrix of the form $\boldsymbol{K} + \sigma_e^2 \boldsymbol{I}$. Similar to the noiseless case, we can then perform predictions via

$$\begin{aligned} p(\boldsymbol{f}^\star | \boldsymbol{X}^\star, \boldsymbol{X}, \boldsymbol{y}) = &\mathcal{N}\left(\boldsymbol{f}^\star \,\middle|\, \boldsymbol{m}(\boldsymbol{X}^\star) + \boldsymbol{K}^\star \left[\boldsymbol{K} + \sigma_e^2 \boldsymbol{I}\right]^{-1} (\boldsymbol{y} - \boldsymbol{m}(\boldsymbol{X}))\,; \right. \\ &\left. \boldsymbol{K}^{\star\star} - \boldsymbol{K}^\star \left[\boldsymbol{K} + \sigma_e^2 \boldsymbol{I}\right]^{-1} \boldsymbol{K}^{\star\top} \right). \end{aligned} \tag{2.47}$$

Making predictions, with or without noise, scales cubically in the number of samples $O(N^3)$ since we have to invert the covariance matrix between the training points. In practice, this can be prohibitive if the number of data points is large ($> 10.000$). However, the development of scalable approximations is still an active area of research, and exciting progress has been made over the last years (Titsias, 2009; Hensman et al., 2013).

### 2.2.4 Learning the hyperparameters

In Section 2.2.2, we got to know a number of different covariance functions. They all share a dependence on some hyperparameters $\boldsymbol{\theta}$. For instance, the squared exponential kernel has two hyperparameters, $\boldsymbol{\theta} = \{\sigma^2, l^2\}$: the scaling factor $\sigma^2$ and the lengthscale $l^2$. In practice, we often do not know what the true values of the hyperparameters are, and we have to infer them from the data. Two popular approaches for doing so are cross-validation and maximizing the evidence of the data. However, both have advantages and disadvantages: Cross-validation can be slow, because it entails a grid search over all possible parameter combinations. Computing the evidence is faster, but often trapped to a local optimum, since the likelihood function is

not convex. However, we also discussed earlier that the success of Gaussian processes is dependent on an expressive covariance function. Unfortunately, the expressiveness of the covariance functions often comes at the price of a large number of hyperparameters (for instance when we start combining different covariance functions), making cross-validation infeasible.

Maximizing the evidence function is often carried out by using a gradient-based optimization technique, such as Quasi-Newton algorithms (Nocedal and Wright, 2000). In a nutshell, all we have to do is to provide the solver with the evidence function

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) &= \log \mathcal{N}\left(\boldsymbol{y}\,|\boldsymbol{m}(\boldsymbol{X}),\boldsymbol{K}\right)\\ &= -\frac{N}{2}\log\left(2\pi\right) - \frac{1}{2}\log|\boldsymbol{K}| - \frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{m}(\boldsymbol{X})\right)^\top \boldsymbol{K}^{-1}\left(\boldsymbol{y}-\boldsymbol{m}(\boldsymbol{X})\right),\end{aligned} \tag{2.48}$$

the gradient with respect to the covariance parameters $\theta_j$

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = &-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{K}^{-1}\frac{\partial}{\partial\theta_j}\boldsymbol{K}\right)\\ &+\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{m}(\boldsymbol{X})\right)^\top \boldsymbol{K}^{-1}\left(\frac{\partial}{\partial\theta_j}\boldsymbol{K}\right)\boldsymbol{K}^{-1}\left(\boldsymbol{y}-\boldsymbol{m}(\boldsymbol{X})\right)\end{aligned} \tag{2.49}$$

and with respect to the mean parameters $\theta_j$

$$\frac{\partial}{\partial\theta_j}\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = -\left(\boldsymbol{y}-\boldsymbol{m}(\boldsymbol{X})\right)^\top \boldsymbol{K}^{-1}\frac{\partial}{\partial\theta_j}\boldsymbol{m}(\boldsymbol{X}). \tag{2.50}$$

The optimizer then finds a local maximum by iteratively approximating the evidence function with a local quadratic model followed by a step in its steepest direction. In contrast to Newton approaches, quasi-Newton approaches build the quadratic approximation without making use of the Hessian, often leading to a superior runtime. For evaluating the function and its gradients, we have to invert the covariance matrix $\boldsymbol{K}$, leading to a runtime of $O(N^3)$ per iteration. By using multiple restarts with different initializations, we can avoid being caught in a bad local maximum.

## 2.3 Summary

In this chapter, we studied linear models and Gaussian processes. It is important to understand the connections between the two. First, Gaussian processes can be derived from Bayesian linear regression by mapping the data into a higher dimensional

feature space. More importantly, we do not need to know that mapping explicitly, as long as we can prove that it exists and can compute its kernel efficiently. Second, linear mixed models are also closely connected to Gaussian processes if the random effect is treated as a Gaussian process prior (Liu et al., 2007). The work presented here builds on recent advances made in both fields.

In genomics, linear mixed models are often the model of choice when it comes to association testing: The random effect allows to correct for confounding, either induced by shared genetic or environmental factors, reducing the number of False Positives. For computing the random effect covariance matrix, a linear kernel on all markers is commonly used. This can be interpreted in two ways: We can either think of the covariance matrix as a genetic similarity matrix that measures the relatedness between individuals by using the SNP markers, or, as a Bayesian linear additive model, in which each marker contributes to the phenotype (Goddard et al., 2009). The second approach reveals one of the main limitations of linear mixed models: It assumes that all SNPs are associated with the phenotype, and it does not allow for outlier SNPs, which have a larger effect size (Zaitlen and Kraft, 2012; Lippert et al., 2013). In the following chapter, we will present an algorithm that relaxes these assumptions by including markers with a large effect size as fixed effects in the model. By designing a better background model, we can increase the power to detect weak associations.

# Chapter 3

# Confounder correction for Lasso methods

One of the key challenges in association testing is, as we elucidated before, to design multivariate methods that can correct for population stratification. Linear mixed models are often used to correct for population stratification, but do predominantly consider individual markers in isolation. In contrast, sparse methods increase the power to detect multifactorial associations, but cannot deal with confounding.

The goal of this chapter is to develop an algorithm that combines the merits of linear mixed models and sparse approaches while allowing efficient computation. Our approach tackles the problem in a three-step procedure: in a first step, it estimates how much phenotypic variance can be explained by population structure. In a second step, it transforms the markers and the phenotype such that the correlation due to the population structure is removed. Finally, a sparse solver is used on the transformed data to identify a set of markers that jointly contribute to the phenotype. The additional runtime for confounder correction is a one-time cubic operation in the number of samples $O(N^3)$, which is negligible compared to the runtime of the sparse solver.

We define our new approach in Section 3.1 and give a detailed description of the inference scheme in Section 3.2. In Section 3.3, our experiments show that the rigorous combination of sparse and mixed modeling approaches yields greater power to detect true causal effects in a large range of settings. In genome-wide association studies in *Arabidopsis thaliana* and linkage mapping in mice, our method achieves significantly more accurate phenotype predictions than its competitors and retrieves associations that are enriched for known candidate genes.

## 3.1 Feature selection in the presence of confounding

Our approach builds on linear mixed models (see Section 2.1.4), explaining the phenotype variability by a sum of individual genetic effects and random confounding variables. In brief, the phenotype of $N$ samples $\boldsymbol{y} = (y_1, \ldots, y_N)$ is expressed as a linear function of the markers $\boldsymbol{X} \in \mathbb{R}^{N \times M}$

$$\boldsymbol{y} = \underbrace{\boldsymbol{X}\boldsymbol{w}}_{\text{genetic factors}} + \underbrace{\boldsymbol{u}}_{\text{confounding}} + \underbrace{\boldsymbol{\epsilon}}_{\text{noise}}. \tag{3.1}$$

Here, $\boldsymbol{\epsilon} \in \mathbb{R}^N$ denotes observation noise and $\boldsymbol{u} \in \mathbb{R}^N$ are confounding influences. Confounding influences in genetic mapping are typically not directly observed, however their Gaussian covariance $\boldsymbol{K}$ can in many cases be estimated from the observed data. To account for confounding by population structure, $\boldsymbol{K}$ can be reliably estimated from genetic markers, for example using the realized relationship matrix which captures the overall genetic similarity between all pairs of samples (Hayes et al., 2009). Similarly, in genetic analyses of gene expression, $\boldsymbol{K}$ can be fit to capture and correct for the confounding effect of gene expression heterogeneity (Listgarten et al., 2010; Fusi et al., 2012). Marginalizing over the random effect $\boldsymbol{u}$ results in a Gaussian marginal likelihood model (Kang et al., 2008) whose covariance matrix accounts for confounding variation and observation noise.

The resulting mixed model is typically considered in the context of single candidate SNPs, i.e. restricting the sum in Eq. (3.1) to a particular SNP while ignoring all others (see Section 2.1.4). While computationally efficient and easy to interpret, this independent analysis can be compromised by complex genetic architectures with some genetic factors masking others (Platt et al., 2010b). Some improvements can be achieved by step-wise regression or forward selection, which has recently been extended to the mixed model framework (Yang et al., 2012a; Segura et al., 2012). However, these approaches are often caught in suboptimal modes as they are order dependent (Segura et al., 2012). As an alternative, we propose an efficient approach to carry out joint inference over all markers as implied by Eq. (3.1). Our approach assesses all SNPs at the same time while accounting for their interdependencies and without making any assumptions on their ordering. To allow for applications to genome-wide SNP data, we regularize the fixed effects by an $\ell_1$-norm, assigning zero effect size to the majority of SNPs as done in the classical Lasso (see Section 2.1.5). We call this approach LMM-Lasso as it combines the advantages of established linear mixed models (LMM) with sparse Lasso regression.

There is a vast amount of literature using a $\ell_1$-regularized approach for genome-wide association studies (Wu et al., 2009; Lee and Xing, 2012; Kim and Xing, 2009). In Foster et al. (2007), a sparse random effect model is proposed, in which the markers are modeled as random effects drawn from a Laplacian distribution. In Hoggart et al. (2008) and Li et al. (2011), the authors suggest to add principal components to the model to correct for population structure. While these approaches can be effective in some settings, principal components cannot account for family structure or cryptic relatedness (Price et al., 2010). Importantly, none of these approaches considers including random effects to control for confounding. A notable exception is the general $\ell_1$-mixed model framework by Schelldorfer et al. (2011) and Schelldorfer and Bühlmann (2011), who consider a random effect component but do not provide a scalable algorithm that is applicable to genome-wide settings. More recently, Zhou et al. (2013) introduced a fully Bayesian approach to tackle the same problem by using a mixture of two Gaussians as prior. This is conceptually close to the work presented here, as it is equivalent to a linear mixed model with a spike-and-slab prior on the fixed effects and employing a linear kernel as random effect covariance matrix.

**Probabilistic model** Let $\boldsymbol{X}$ denote the $N \times M$ matrix of $M$ SNPs for $N$ individuals, $\boldsymbol{x}_j$ is then the $N \times 1$ vector representing SNP $j$. We model the phenotype for $N$ individuals, $\boldsymbol{y} = (y_1, \ldots, y_N)$ as the sum of genetic effects $w_j$ of SNPs $\boldsymbol{x}_j$ and confounding influences $\boldsymbol{u}$ (see Eq. (3.1)). The genetic effects are treated as fixed effects, whereas the confounding influences are modeled as random effects. The genetic effect terms are summed over genome-wide polymorphisms, where the great majority of SNPs has zero effect size, i.e. $w_j = 0$, which is achieved by a Laplace shrinkage prior on all weights. The random variable $\boldsymbol{u}$ is not observed directly. Instead, we assume that the distribution of $\boldsymbol{u}$ is Gaussian with covariance $\boldsymbol{K}$, $\boldsymbol{u} \sim \mathcal{N}(0, \sigma_g^2 \boldsymbol{K})$.

Assuming Gaussian noise, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_e^2 \boldsymbol{I})$, and marginalizing over the random variable $\boldsymbol{u}$, we can write down the conditional posterior distribution over the weight vector $\boldsymbol{w}$:

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{K}, \sigma_g^2, \sigma_e^2, \lambda) \propto \underbrace{\mathcal{N}\left(\boldsymbol{y} \,|\, \boldsymbol{X}\boldsymbol{w}, \sigma_g^2 \boldsymbol{K} + \sigma_e^2 \boldsymbol{I}\right)}_{\text{marginal likelihood}} \underbrace{\prod_{m=1}^{M} e^{-\frac{\lambda}{2}|w_m|}}_{\text{prior}}. \tag{3.2}$$

Here, $\lambda$ denotes the sparsity hyperparameter of the Laplace prior, $\sigma_e^2$ is the residual noise variance and $\sigma_g^2$ denotes the variance of the random effect component.

## 3.2 Parameter inference

Learning the hyperparameters $\boldsymbol{\Theta} = \{\lambda, \sigma_g^2, \sigma_e^2\}$ and the weights $\boldsymbol{w}$ jointly is a hard non-convex optimization problem. Here, we propose a combination of fitting some of these parameters on the null model with the individual SNP effects excluded and reduction to a standard Lasso regression problem.

**Null-model fitting** To obtain a practical and scalable algorithm, we first optimize $\sigma_g^2, \sigma_e^2$ by maximum likelihood under the null model ($\boldsymbol{w} = \boldsymbol{0}$), ignoring the effect of individual SNPs. The analogous procedure is widely used in single-SNP mixed models, and has been shown to yield near-identical results to an exact approach (Kang et al., 2010). To speed up the computations needed, we optimize the ratio of the random effect and the noise variance, $\delta = \sigma_e^2/\sigma_g^2$, which can be optimized efficiently by using computational tricks similar to (2.23):

$$p(\boldsymbol{y}|, \boldsymbol{K}, \sigma_g^2, \delta, \lambda) \propto \mathcal{N}\left(\boldsymbol{y} \,|\boldsymbol{0}, \sigma_g^2(\boldsymbol{K} + \delta \boldsymbol{I})\right). \tag{3.3}$$

Briefly, we compute the eigendecomposition of the covariance $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ which can be used to rotate the data such that the covariance matrix of the normal distribution is isotropic. We carry out one-dimensional numerical optimization of the marginal likelihood (Eq. (3.3)) with respect to $\delta$, whereas $\sigma_g^2$ can be optimized in closed form in every evaluation.

**Whitening the data** Having fixed $\delta$, we use the eigendecomposition of $\boldsymbol{K}$ again to rotate our data such that the covariance matrix becomes isotropic:

$$p(\boldsymbol{w}|\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{X}}, \boldsymbol{K}, \sigma_g^2, \lambda) \propto \mathcal{N}\left(\tilde{\boldsymbol{y}} \,\middle|\, \tilde{\boldsymbol{X}}\boldsymbol{w}, \sigma_g^2 \boldsymbol{I}\right) \prod_{m=1}^{M} e^{-\frac{\lambda}{2}|w_m|} \tag{3.4}$$

Here, $\tilde{\boldsymbol{X}}$ denote the rotated and rescaled genotypes and $\tilde{\boldsymbol{y}}$ the respectively phenotypes:

$$\begin{aligned} \tilde{\boldsymbol{X}} &= (\boldsymbol{S} + \delta \boldsymbol{I})^{-\frac{1}{2}} \mathbf{U}^\top \mathbf{X}, \\ \tilde{\boldsymbol{y}} &= (\boldsymbol{S} + \delta \boldsymbol{I})^{-\frac{1}{2}} \mathbf{U}^\top \mathbf{y}. \end{aligned} \tag{3.5}$$

In Figure 3.1, we show graphically how the correlation between the SNPs and the phenotypes are resolved when projecting both.

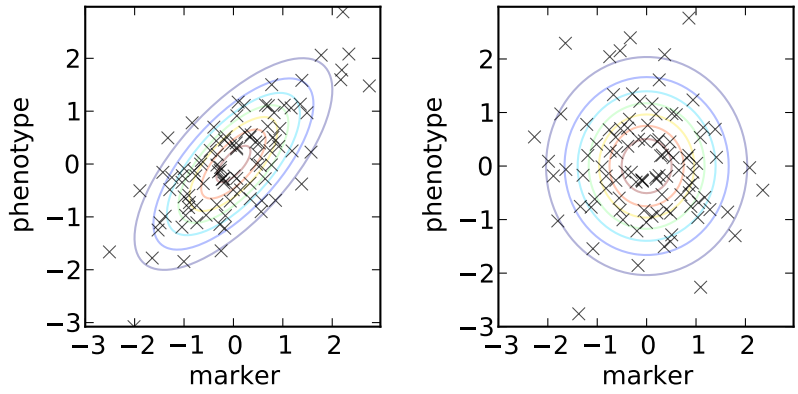



Figure 3.1: **Whitening the data.** The covariance matrix $\boldsymbol{K} + \delta \boldsymbol{I}$ is used to decorrelate the markers from the phenotype by projecting them along the principal components and rescaling them to unit variance.

**Solving the Lasso** Using this transformation, the task of determining the most probable weights in Eq. (3.4) is now equivalent to the Lasso regression model, since maximizing the posterior with respect to $\boldsymbol{w}$ is equivalent to minimizing the negative log of Eq. (3.4):

$$\min_{\boldsymbol{w}} \frac{1}{\sigma_g^2} \|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{X}}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_1. \tag{3.6}$$

An appropriate setting of $\lambda$ can be found by cross-validation to maximize the overall predictive performance or stability selection (Meinshausen and Bühlmann, 2010).

The computational efficiency of the three-stage procedure proposed here depends on the approximation to fit $\delta$ on the null model, allowing for the reduction of the problem to standard Lasso regression. For univariate single-SNP mixed models, efficient optimization of $\delta$ for each SNP can be done by recently proposed computational tricks (Lippert et al., 2011; Zhou and Stephens, 2012). Unfortunately, these techniques cannot be directly applied in the multivariate setting. In principle it is possible to extend the cross-validation to optimize over pairs $(\delta, \lambda)$. However, this remains impracticable for most datasets due to the additional computational cost implied and hence we consider optimizing $\delta$ on the null model in the experiments (Kang et al., 2010).

### 3.2.1 Phenotype prediction

Given a trained LMM-Lasso model on a set of genotypes and phenotypes, we can predict the unobserved phenotype of test individuals. The predictive distribution can be derived by conditioning the joint distribution over all individuals on the training individuals, resulting in a Gaussian predictive distribution (2.43) with mean

$$\boldsymbol{m}^\star = \underbrace{\boldsymbol{X}^\star \boldsymbol{w}}_{\text{Lasso prediction}} + \underbrace{\boldsymbol{K}^\star(\boldsymbol{K}+\delta\boldsymbol{I})^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})}_{\text{Random effect prediction}} \tag{3.7}$$

and covariance

$$\boldsymbol{\Sigma}^\star = \sigma_g^2(\boldsymbol{K}^{\star\star}+\delta\boldsymbol{I}) - \sigma_g^2\boldsymbol{K}^\star(\boldsymbol{K}+\delta\boldsymbol{I})^{-1}\boldsymbol{K}^{\star\top}, \tag{3.8}$$

where $\boldsymbol{K}^\star$ is the covariance matrix between the test and the training samples and $\boldsymbol{K}^{\star\star}$ the covariance between the test samples. The mean prediction is thereby a sum of contributions from the Lasso component and the Gaussian process prediction on the Lasso residuals.

### 3.2.2 Choice of the random effect covariance to account for population structure

Depending on the application, the random effect covariance $\boldsymbol{K}$ can be chosen in a variety of ways. Here, we discuss specific options to account for population structure.

**Choice of genetic similarity matrix** For the identity by descent matrix (IBD), an entry is defined as the predicted proportion of the genome that is identical by descent given the pedigree information. In contrast, the identity by state matrix (IBS) simply counts the number of loci on which the samples agree, whereas the realized relationship matrix (RRM) is calculated as the linear kernel between the SNPs (Hayes et al., 2009). In subsequent experiments, we have used the realized relationship matrix. An example for the RRM-matrix derived from the *Arabidopsis thaliana* dataset is given in Figure 3.2

**Realized relationship matrix and relationship to Bayesian linear regression** From a Bayesian perspective, employing the realized relationship matrix as the covariance matrix is equivalent to integrating over all SNPs in a linear additive model with an independent Gaussian prior over the weights $\mathcal{N}\left(\mathbf{0}, \frac{\sigma_g^2}{M}\boldsymbol{I}\right)$ (Goddard et al., 2009). The choice of a Gaussian prior reflects the belief that many markers

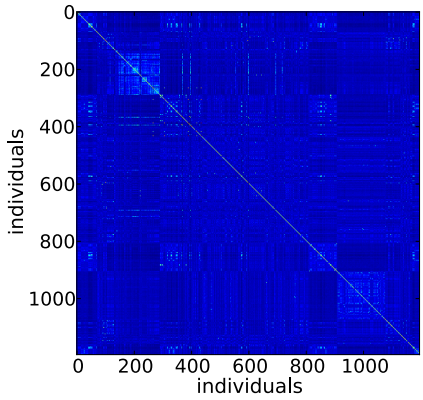



Figure 3.2: **Realized relationship matrix from the 1196 plants of *Arabidopsis thaliana* available from (Horton et al., 2012).** The relatedness between the individuals is complex and strong as the matrix is deeply structured.

have a small effect on the phenotype. However, it does not allow for a few markers to have a large effect on the phenotype as it is often the case (Zaitlen and Kraft, 2012). For instance in rice, large amounts of the phenotypic variance of agronomic traits can be explained by single markers (Huang et al., 2010).

Thus, choosing this particular covariance matrix $\mathbf{K}$ can be regarded as modeling genetic effects that are confounded due to population structure or to small additive infinitesimal effects, whereas single SNPs that have a sufficiently large effect size are directly included as fixed effects into the Lasso model.

### 3.2.3 Relationship to stepwise regression

The difference between stepwise regression and the Lasso can be seen easiest by going over Stagewise Linear Regression. In stepwise regression, we start with the SNP having the largest effect size. We then iteratively add SNPs that can explain most of the phenotype conditioned on the markers that have already been selected. In Stagewise Linear Regression instead, one moves only a small step in the direction of the most correlated SNP and then re-estimates the most correlated SNP on the residual phenotype which is far less greedy. In Efron et al. (2004), it is shown that there is a close relation between Stagewise Linear Regression and Lasso resulting in almost identical solutions.

### 3.2.4 Scalability and runtime

The appeal of the LMM-Lasso is a runtime performance comparable to the standard LASSO. The difference is a one-time off cubic cost for the decomposition of the random effect matrix $\boldsymbol{K}$ to rotate the genotype and phenotype data (3.5).

To demonstrate the applicability to genome-wide datasets, we have empirically measured the runtime for computing the complete path of sparsity regularizers on the synthetic dataset, consisting of 1,196 plants and 213,624 SNPs. On a single core of a Mac Pro (3GHz, 12 MB L2-Cache, 16GB Memory), the Lasso required 145 minutes CPU time and the LMM-Lasso 146 minutes of CPU time.

If needed, the runtime of LMM-Lasso could be improved in several ways. First, the runtime of the $\ell_1$-solver is heavily dependent on the optimization method used, see also Section 2.1.5 for a discussion of state-of-the art methods. Second, if the number of samples is large ($N > 10^5$), the runtime is dominated by the decomposition of $\boldsymbol{K}$ and rotating the data for the optimization of $\delta$. As shown in Lippert et al. (2011), reducing the covariance $\boldsymbol{K}$ to a low-rank representation calculated from a small subset of $M_{active}$ SNPs, yields very similar results while reducing the runtime from $O(N^2M)$ to $O(NM_{active}^2)$.

## 3.3 Experiments

**Preprocessing** We standardized the SNP data, which has the effect that the prior on the effect size is dependent on the minor allele frequency (MAF): SNPs with a low MAF require a smaller weight to have the same effect on the phenotype, and hence will be more likely driven to zero at the MAP-solution. On the phenotypes, we performed a Box-Cox transformation (Sakia, 1992) and subsequently standardized the data.

**Model selection** Variation of the model complexity of Lasso methods can either be done by choosing the number of active SNPs or equivalently by varying the hyperparameter $\lambda$ explicitly. For the benefit of direct interpretability, we chose to vary the number of active SNPs. For a fixed number of selected SNPs, we find the corresponding hyperparameter $\lambda$ by a combination of bracketing and bisection as done in Wu et al. (2009). To select which of these Lasso-model is most suitable, we consider alternative strategies, depending on the objective.

1. **Phenotype prediction** To predict phenotypes, we use 10-fold cross-validation. We split the data randomly into 10 folds. Each fold is once picked as test dataset, with all other folds being used for training the model. The model is selected to maximize the explained variance on the test set. In this comparison, we consider models with different numbers of SNPs, varying from $\{0, 1, 2, \ldots, 10, 20, 30, \ldots, 100, 150, 200, 250\}$ with the additional constraint that the number of active SNPs shall not exceed the number of samples.

2. **Variable selection** To assess the significance of individual features, we consider stability selection (Meinshausen and Bühlmann, 2010). Here, we fix the number of active SNPs to 20 and draw randomly 90% of the data 100 times. To accommodate the limited sample size, we did not use 50% of the samples for each draw as proposed in the original article. We selected all SNPs that were found in $> 50\%$ of all restarts. We used the smallest threshold possible to also detect SNPs that have a small effect size. In consequence, we allowed to select of a modest number of false-positive results. Significance estimates can be deduced from the selection frequency of individual SNPs (Meinshausen et al., 2009).

   To obtain a complete ranking of features, as used to evaluate models in the simulation study, we use the LASSO regularization path and rank features by the order of inclusion into the model.

### 3.3.1 Semi-empirical setting with known ground truth

We assessed the ability of LMM-Lasso to recover true genotype to phenotype associations in a semi-empirical simulated dataset based on the extended *A. thaliana* dataset (Horton et al., 2012) consisting of 1196 plants. To ensure realistic characteristics of population structure, we simulated confounding such that it borrows key characteristics from *Arabidopsis thaliana*, which is a strongly structured population. We considered real phenotype data to obtain realistic background signal that is subject to population structure. In addition to this empirical background, we added simulated associations with different effect sizes and a range of complexities of the genetic models.

For simulating population-driven effects, we used the real phenotype leaf number at flowering time (LN, 16°C, 16 hrs daylight) which is available for 176 plants. Univariate analyses as done in Atwell et al. (2010) have shown that the phenotype has an excess of associations when population structure is not accounted for. On the other hand, after correction the p-values are approximately uniformly distributed. First, to determine the fraction of genetic and residual variance, we fit a random effects model to LN, which we subsequently used to predict the population structure for the remaining 1,020 plants. We then simulated the phenotypes as follows:

$$\boldsymbol{y} = \sigma^2_{\text{sig}}\boldsymbol{y}_{\text{sig}} + (1-\sigma^2_{\text{sig}})[\sigma^2_{\text{pop}}\boldsymbol{y}_{\text{pop}} + (1-\sigma^2_{\text{pop}})\boldsymbol{\epsilon}],$$

where $\boldsymbol{y}_{\text{sig}} = \boldsymbol{X}^{(k)}\mathbf{w}$, $\boldsymbol{X}^{(k)}$ is the SNP data for the $k$ causal SNPs, $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \frac{1}{k}\boldsymbol{I})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The first two causal SNPs are drawn such that they are in close

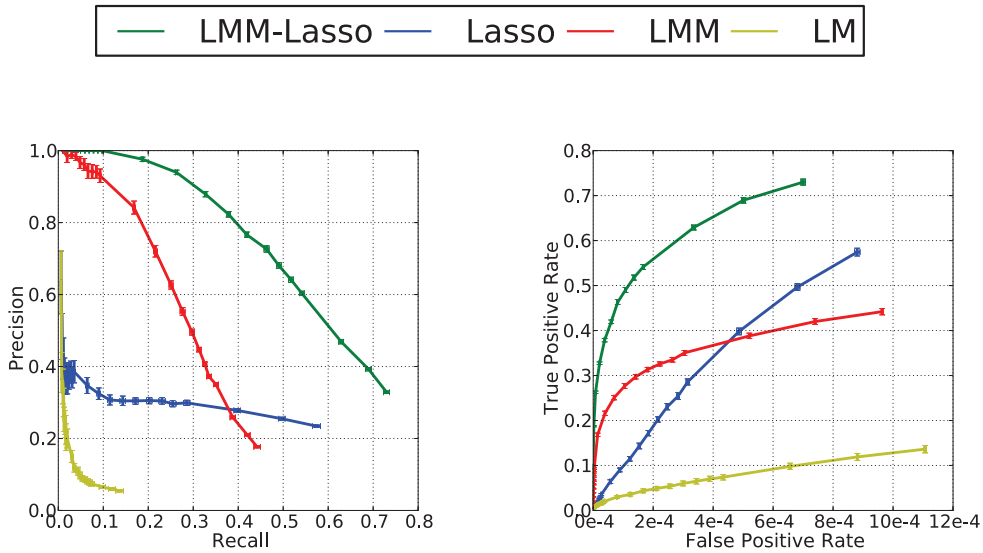
linkage (distance between 1kb and 10kb), the remaining causal SNPs are randomly drawn from the complete genome.

The initial settings used for the simulation experiments were $\sigma^2_{\text{sig}} = 0.7$, $\sigma^2_{\text{pop}} = 0.5$ and $k = 100$. To determine the influence of the population strength, we considered $\sigma^2_{\text{sig}} = 0.5$, $k = 20$ and varied $\sigma^2_{\text{pop}} \in \{0.0, 0.3, 0.5, 0.7, 0.9, 1.0\}$. To experimentally assess the impact of the overall noise, we fixed $k = 100, \sigma^2_{\text{pop}} = 0.5$, and let $\sigma^2_{\text{sig}}$ vary in $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. Finally, we considered different numbers of causal SNPs $k \in \{10, 20, 500, 100, 300, 1000\}$ and fixed $\sigma^2_{\text{sig}} = 0.7, \sigma^2_{\text{pop}} = 0.5$. For the linkage experiments, we used the $\sigma^2_{\text{sig}} = 0.7$, $\sigma^2_{\text{pop}} = 0.5$ and $k = 10$. We simulated 30 phenotypes for all settings.

To compare our method to existing techniques, we considered the standard Lasso, which models all SNPs jointly but without correcting for population structure, as well as an univariate Linear Mixed Model, which effectively controls for confounding, but considers each SNP in isolation. As a baseline, we also considered a standard univariate Linear Model (LM), which neither accounts for confounding nor considers joint effects due to complex genetic architectures. Both, the standard Lasso and LMM-Lasso were fit in identical ways (see Section 3.3). For the linear mixed model and the LMM-Lasso, we used the RRM as covariance matrix and fit $\delta$ on the null model. For univariate models, the ranking of individual SNPs was done according to their $p$-values, for multivariate models we considered the order of inclusion into the model.

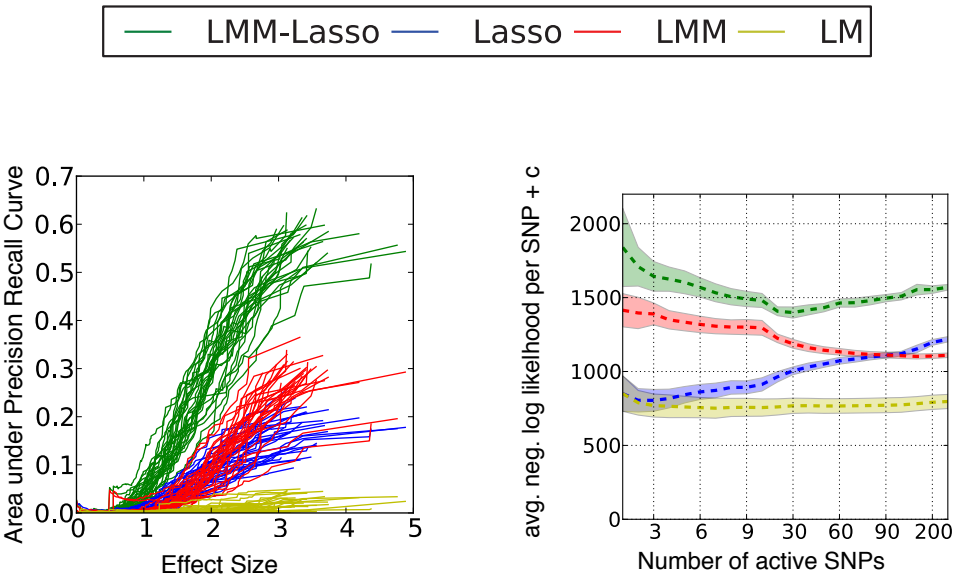
Since in many cases the causal loci might not be genotyped and stochastic effects might cause larger correlations between strongly correlated SNPs and the phenotype, we consider a SNP, called positive by the model, as a True Positive if it is in close proximity to the known causal SNP (+/- 10kb). On the other hand, if a SNP, called positive by the model, is not close to a causal SNP, it is considered as a False Positive. A fair comparison between the univariate and multivariate methods is difficult as the univariate methods select blocks of linked markers, whereas the multivariate methods select only representative markers per block. To account for this principle difference, we employed a post-processing procedure to sparsify the solutions of all methods in a comparable manner. For this purpose, we iteratively selected the most associated marker genome-wide. To ensure that the next marker is not in LD to this SNP, we ignored neighboring markers (+/- 10kb) and proceeded with selecting the next SNP. This process was repeated until no marker above the threshold was left.

**LMM-Lasso ranks causal SNPs higher than alternative methods** First, we compared the alternative methods in terms of their accuracy in recovering SNPs with a true simulated association (Figure 3.3a). Methods that account for popula-




(a) Precision/Recall (b) ROC

Figure 3.3: **Evaluation of alternative methods on semi-empirical GWAS datasets, mimicking population structure as found in *Arabidopsis thaliana*.** **(a)** Precision-Recall Curve for recovering simulated causal SNPs using alternative methods. Shown is precision (TP/(TP+FP)) as a function of the recall (TP/(TP+FN)). **(b)** Alternative evaluation of each method on the identical dataset using Receiver operating characteristics (ROC). Shown is the True Positive Rate (TPR) as a function of the False Positive Rate (FPR).




(a) Precision/Recall

(b) ROC

Figure 3.4: **Characteristics of alternative methods on semi-empirical GWAS dataset. (a)** Area under the precision recall curve as a function of the total effect size of all causal SNPs. **(b)** Average negative log-likelihood of each selected SNPs under the multivariate normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{K})$ as a function of the number of SNPs that are active in the model. The smaller the log likelihood is, the more the SNPs are correlated with the population structure. For the LMM-Lasso and the Lasso active SNPs have been selected by following the regularization path. For linear mixed model (LMM) and linear model (LM), the set of active SNPs have been obtained in ascending order of the p-value obtained. In the beginning, Lasso and the linear model choose SNPs that heavily reflect the population structure, while the mixed model approaches do not. In both figures, the number of causal SNPs was 100.

tion structure (LMM-Lasso, LMM) are more accurate than their counterparts, with LMM-Lasso performing best. While the linear mixed model performs well at recovering strong associations, the independent statistical testing falls short in detecting weaker associations which are likely masked by stronger effects (Figure 3.4a). Comparing methods that account for population structure and naive methods, we observe that accounting for this confounding effect avoids the selection of SNPs that merely reflect relatedness without a causal effect (Figure 3.4b). An alternative evaluation, which considers the receiver operating characteristic curve, given in Figure 3.3b, yields identical conclusions.

Next, we explored the impact of variable simulation settings. As common in the literature, we used the area under the precision-recall curve as a summary performance measure to compare different algorithms. Precision and recall both depend on the decision threshold, above which a marker is predicted to be activated. By varying this threshold, one obtains a precision-recall curve.

Figure 3.5a shows the area under the precision recall curve as a function of an increasing ratio of population structure and independent environmental noise. When the confounding population structure is weak, both the Lasso and the LMM-Lasso perform similar. As expected, the benefits of population structure correction in LMM-Lasso are most pronounced in the regime of strong confounding. We also examined the ability of each method to recover genetic effects for increasing complexities of the genetic model, varying the number of true causal SNPs while keeping the overall genetic heritability fixed (Figure 3.5b). LMM-Lasso performs better than alternative methods for the whole range of considered settings with the difference in accuracy being the largest for genetic architectures of medium complexity. These results show that, in the regime of a larger number of true weak associations, it is advantageous to include a genetic covariance $\boldsymbol{K}$ that accounts for some of the weak effects (Yang et al., 2010). The identical effect is observed when varying the ratio between true genetic signal versus confounding and noise (Figure 3.5c). Again, the performance of the LMM-Lasso is superior to all other methods and the strengths are particularly visible for medium signal to noise ratios.

**Multivariate models better differentiate multiple causal loci from correlation due to linkage** Previously, step-wise regression models that include genetic variants in the order of effect sizes have been considered to differentiate between true genetic heterozygosity and local correlation due to linkage (Yang et al., 2012a). Here, we show that LMM-Lasso can be successfully applied for the same task, however with the additional benefit that a step-wise order of including genetic markers as co-factors is not needed (Figure 3.6). The comparison includes true genetic het-

erogeneity where two loci within linkage disequilibrium (LD) jointly regulate the phenotype (left) as well as a single genetic effect that is broadened by LD (right). The LMM-Lasso model is able to differentiate between the two types of genetic architectures reliably, whereas the Linear Mixed Model suffers from correlation due to linkage.

### 3.3.2 LMM-Lasso explains the genetic architecture of complex traits in model systems

Having shown the accuracy of LMM-Lasso in recovering causal SNPs in simulations, we now demonstrate that the LMM-Lasso better models the genotype-to-phenotype map in *Arabidopsis thaliana* and mouse (Valdar et al., 2006a). In this experiment, we focus on 20 flowering time phenotypes for *Arabidopsis thaliana*, which are well characterized, and 273 mouse phenotypes which are relevant to human health.

**Data** We obtained genotype and phenotype data for up to 199 accessions of *Arabidopsis thaliana* from Atwell et al. (2010). Each genotype comprises 216,130 single nucleotide polymorphisms per accession. We study the group of phenotypes related to the flowering time of the plants. We excluded phenotypes that were measured for less than 150 accessions to avoid possible small sample size effects, resulting in a total of 20 flowering phenotypes that were considered. The relatedness between individuals ranges in a wide spectrum leading to a complex population structure (Platt et al., 2010a).

We also obtained genotype and phenotype data for 1,940 mice from a multiparent inbred population (Valdar et al., 2006a). Each individual genotype comprises of 12,226 single nucleotide polymorphisms. All mice were derived from eight inbred strains and were crossed to produce a heterogenous stock. The phenotypes span a large variety of different measurements ranging from biochemical to behavioral traits. Here, we focused on 273 phenotypes which have numeric or binary values.

**LMM-Lasso more accurately predicts phenotype from genotype and uncovers sparser genetic models** First, we considered phenotype prediction to investigate the capability of alternative methods to explain the joint effect of groups of SNPs on phenotypes. To measure the predictive power, we assessed which fraction of the total phenotypic variation can be explained by the markers using different methods (Ober et al., 2012). Explained variance is defined as the fraction of the total variance of the phenotype that can be explained by the model and in our experiments equals one minus the mean squared error as we preprocesed the data to have

zero-mean and unit-variance. We avoided prediction on the training data, as for all methods this leads to anti-conservative estimates of variance explained due to overfitting.

Figure 3.7a and 3.7b show the explained variance of the two methods on the independent test data set for each phenotype in the two datasets. For both model organisms, LMM-Lasso explained at least as much variation as the Lasso. In a fraction of 85.00% of the *Arabidopsis thaliana* and 91.58% of the mouse phenotypes, LMM-Lasso was more accurate in predicting the phenotype and thus explained a greater fraction of the phenotype variability from genetic factors than the Lasso. In contrast, Lasso achieved better performance in only 15.00% of the *Arabidopsis thaliana* and 8.42% of the mouse phenotypes. Beyond an assessment of the genetic component of phenotypes, LMM-Lasso dissects the phenotypic variability into the contributions of individual SNPs and of population structure. Figure 3.7c and 3.7d show the number of SNPs selected in the respective genetic models for prediction. With the exception of two phenotypes, LMM-Lasso selected substantially fewer SNPs than the Lasso, suggesting that the Lasso includes additional SNPs into the model to capture the effect of population structure through an additional set of individual SNPs. This observation is in line with the insights derived from the simulation setting where the majority of excess SNPs selected by Lasso are indeed driven by population effects (Figure 3.4b). Although the genetic models fit by LMM-Lasso are substantially sparser, they nevertheless suggest complex genetic control by multiple loci. In 90.00% of *Arabidopsis thaliana* and in 66.06% of the mouse phenotypes, LMM-Lasso selected more than one SNP, in 40.00/45.49% of the cases the number of SNPs in the model was greater than 10.

**LMM-Lasso allows for dissecting individual SNP effects from global genetic effects driven by population structure** Next, we investigated the ability of LMM-Lasso to differentiate between individual genetic effects and effects caused by population structure. Figure 3.8 shows the explained variances for the phenotype flowering time (measured at 10° C) for *Arabidopsis thaliana*. Again, these estimates were obtained using a cross validation approach. It is known (Zhao et al., 2007) that flowering is strikingly associated with population structure, which explains why the LMM-Lasso already captured a substantial fraction (45.17%) of the phenotypic variance, when using realized relationships alone (no active SNPs). Due to the small sample size, cross-validation can underestimate the true explained variance (Hastie et al., 2009). Nevertheless, cross-validation is fair for comparison and conservative as it avoids possible overfitting.

For increasing number of SNPs included in the model, the explained variance

| Phenotype | LMM-Lasso | Lasso |
|---|---|---|
| LD | **5/54** | 4/69 |
| LDV | **5/63** | 3/69 |
| SD | **3/55** | 2/61 |
| SDV | **5/54** | 2/60 |
| FT10 | 1/48 | **4/67** |
| FT16 | 3/51 | **4/68** |
| FT22 | **2/54** | 1/64 |
| 2W | **3/53** | 2/65 |
| 8W | 2/51 | **4/59** |
| FLC | **5/52** | 3/53 |
| FRI | 3/43 | 3/46 |
| 8WGHFT | **4/59** | 2/66 |
| 8WGHLN | 1/48 | **4/58** |
| 0WGHFT | **4/58** | 3/63 |
| FTField | **4/61** | 3/69 |
| FTDiameterField | 1/49 | 1/51 |
| FTGH | 1/49 | **2/61** |
| LN10 | **3/50** | 2/67 |
| LN16 | 2/58 | **3/64** |
| LN22 | **4/54** | 2/65 |

Table 3.1: **Associations close to known candidate genes.** We report true positives/positives (TP/P) for LMM-Lasso and Lasso for all phenotypes related to flowering time in *Arabidopsis thaliana*. P are all activated SNPs and TP are all activated SNPs that are close to candidate genes.

of LMM-Lasso gradually shifted from the kernel to individual SNP effects. In this example, the best performance (48.87%) was reached with 30 SNPs in the model where the relative contribution of the random effect model was 33.10% and of the individual SNPs is 15.77%. In comparison, Lasso explained at most 46.53% of the total variance, when 125 SNPs were included in the model.

**Associations found by LMM-Lasso are enriched for SNPs in proximity to known candidate genes** Finally, we considered the associations retrieved by alternative methods in terms of their enrichment near candidate genes with known implications for flowering in *Arabidopsis thaliana*. Lippert et al. (2011) showed that

| Phenotype | Chrom. | Position | GeneID | LM | LMM |
|---|---|---|---|---|---|
| LD | 4 | (466307,466800) | AT4G01060 | (2.55,6.40) | (3.37,4.20) |
| 2W | 4 | (454542,460246) | AT4G01060 | (8.29,1.89) | (6.03,4.26) |
| FLC | 4 | (205170,210657) | AT4G00450 | (6.88,5.40) | (5.01,4.78) |
| FRI | 4 | (268809,268990) | AT4G00650 | (20.91,15.13) | (17.45,13.65) |
| FRI | 4 | (268990,276143) | AT4G00650 | (15.13,17.36) | (13.65,14.37) |

Table 3.2: **Candidate genes containing multiple associations.** List of all candidate genes that have two activated SNPs in close proximity for all phenotype related to flowering time of *Arabidopsis thaliana*. The last two columns show the $-\log_{10}$ transformed $p$-values for the linear and the linear mixed model.

it can be advantageous to remove the SNP of interest from the population structure covariance. Thus, we applied LMM-Lasso on a per-chromosome basis estimating the effect of population structure from all remaining chromosomes. To obtain a comparable cutoff of significance, we employed stability selection for both the LMM-Lasso and Lasso (see Section 3.3).

Table 3.1 shows that the LMM-Lasso found a greater number of SNPs linked to candidate genes for twelve phenotypes, whereas Lasso retrieved a greater number for only six phenotypes. In the remaining two phenotypes, both methods performed identically We also investigated to what extent the solution is affected by different selection thresholds (see Figure 3.9). Reassuringly, the LMM-Lasso outperformed the standard Lasso over a large range of different values.

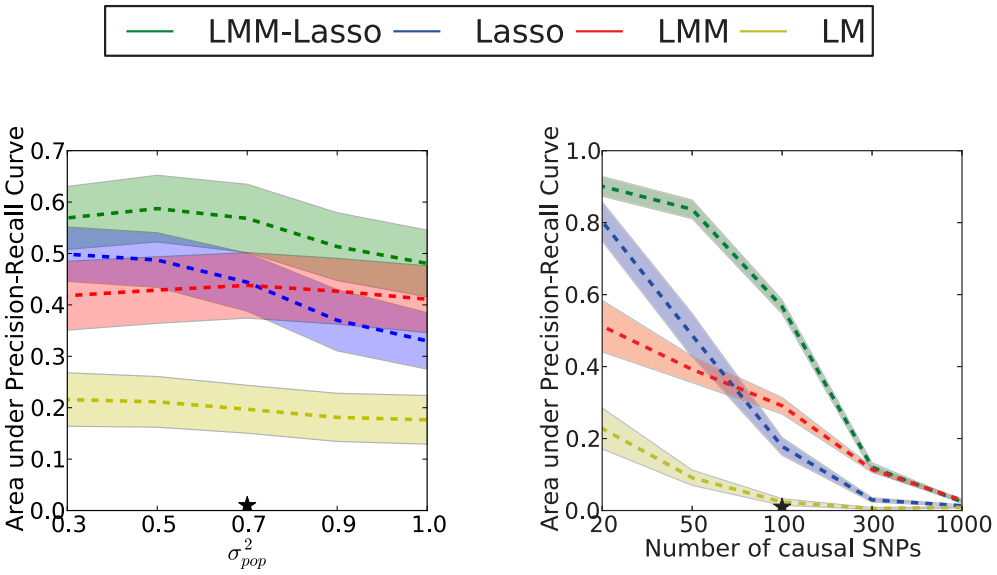
We also considered to what extent the findings provide evidence for allelic heterogeneity or the existence of an imperfectly tagged causal locus. Overall, 14.75% of the SNPs linked to candidate genes and selected by the LMM-Lasso appear as adjacent pairs (Table 3.2), i.e. having a distance less than 10kb to each other, while 5.56% of the SNPs selected by the Lasso do. From all activated SNPs, 8.18% selected by LMM-Lasso and 18.96% selected by the Lasso have at least a second active SNP in close proximity.

## 3.4 Summary

In this chapter, we have presented a Lasso multi-marker mixed model for detecting genetic associations in the presence of confounding influences such as population structure. The approach combines the attractive properties of mixed models that allow for elegant correction for confounding effects and those of multi-marker models that consider the joint effects of sets of genetic markers rather than one single locus.

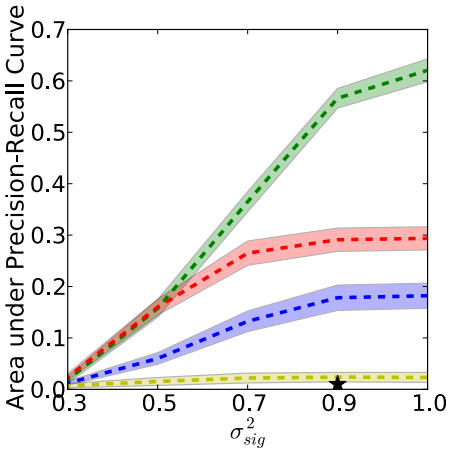
In our experiments, we could show that the LMM-Lasso does not only improve the prediction accuracy, but also allows for dissecting the explained variance into broad-scale genetic effects and individual genetic effects.

Arguably, the LMM-Lasso works best if population structure is present and few markers have a strong effect on the phenotype. If only population structure is present, the solution found by the LMM-Lasso resembles the ridge regression estimate. In contrast, if no population structure is present, it is similar to the Lasso solution. The power of the presented method lies in its ability to adapt to the genetic architecture at hand as it reduces the assumptions made on the effect sizes.



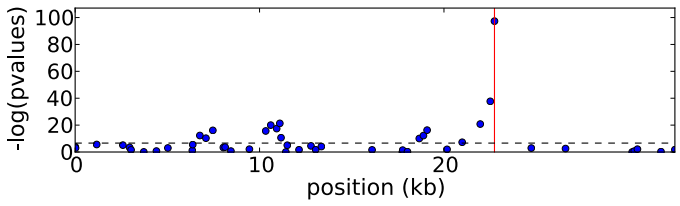

(a) Population structure strength

(b) Trait complexity: Varying signal strength

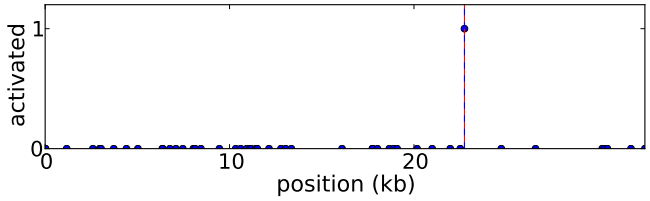



(c) Trait complexity: Varying number of causal SNPs

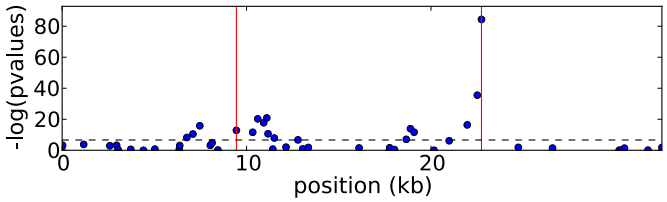
Figure 3.5: **Evaluation of alternative methods on the semi-empirical GWAS dataset for different simulation settings.** Area under precision recall curve for finding the true simulated associations. Alternative simulation parameters have been varied in a chosen range. **(a)** Evaluation for different relative strength of population structure $\sigma^2_{\text{pop}}$. **(b)** Evaluation for true simulated genetic models with increasing complexity (more causal SNPs). **(c)** Evaluation for variable signal to noise ratio $\sigma^2_{\text{sig}}$.



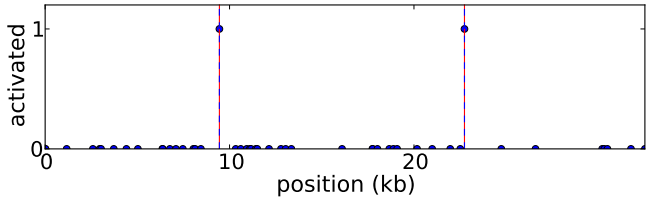

(a) Linear Mixed Model, one causal variant




(b) LMM-Lasso, one causal variant

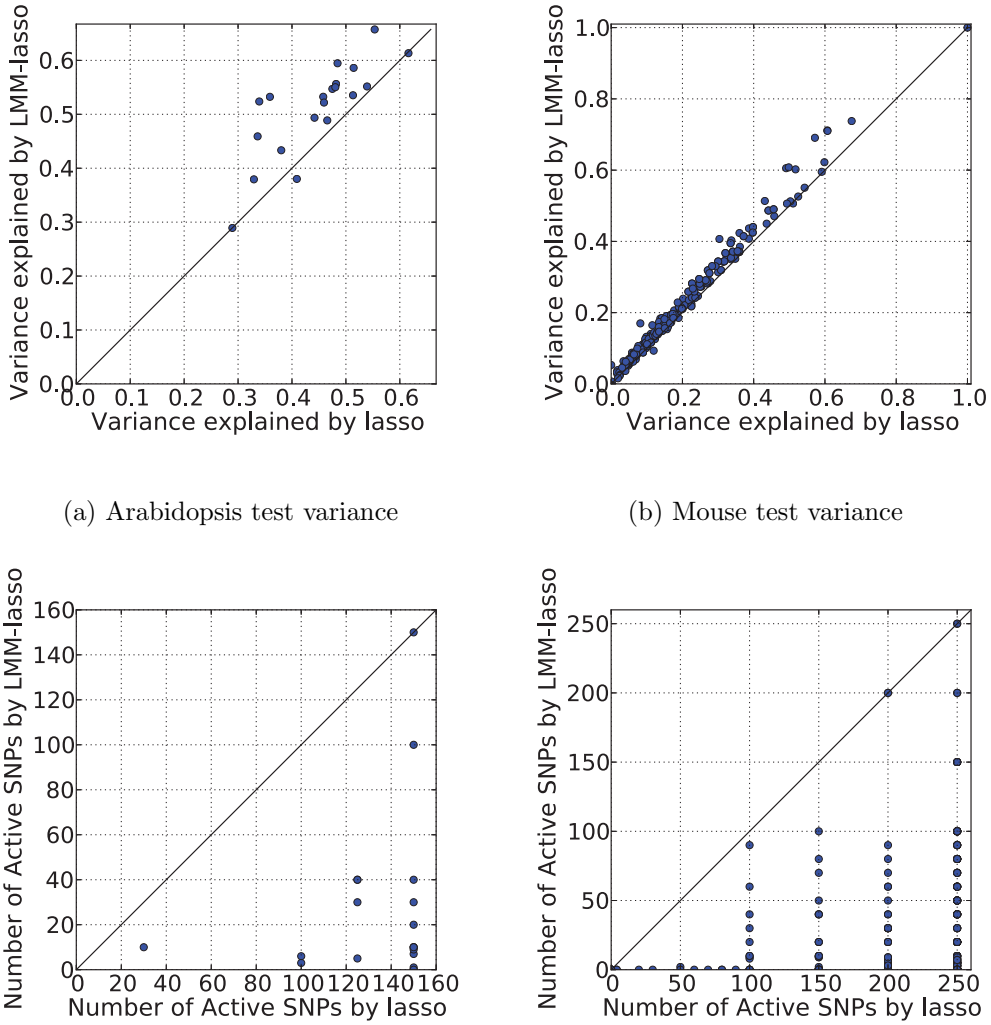



(c) Linear Mixed Model, two causal variants




(d) LMM-Lasso, two causal variants

Figure 3.6: **Differentiation between multiple causal loci from spurious correlation due to linkage on simulated data.** The upper two plots show a single SNP with a strong effect in an LD block. The lower two plots show the same LD block, but with an additional SNP effect with weaker effect size in the opposite direction. While both methods detect the SNP with large effect size, the second one is only uniquely recovered by the LMM-Lasso. The red lines indicate the causal SNPs, the blue dots the assigned score.




(a) Arabidopsis test variance

(b) Mouse test variance

(c) Arabidopsis number of SNPs

(d) Mouse number of SNPs

Figure 3.7: **Predictive power and sparsity of the fitted genetic models for Lasso and LMM-Lasso applied to quantitative traits in model systems.** Considered were flowering phenotypes in *Arabidopsis thaliana* and bio-chemical and physiological phenotypes with relevance for human health profiled in mouse. Comparative evaluations include the fraction of the phenotypic variance predicted and the complexity of the fitted genetic model (number of active SNPs). **(a)** Explained variance in *Arabidopsis*. **(b)** Explained variance in mouse. **(c)** Complexity of fitted models in *Arabidopsis*. **(d)** Complexity of fitted models in mouse.

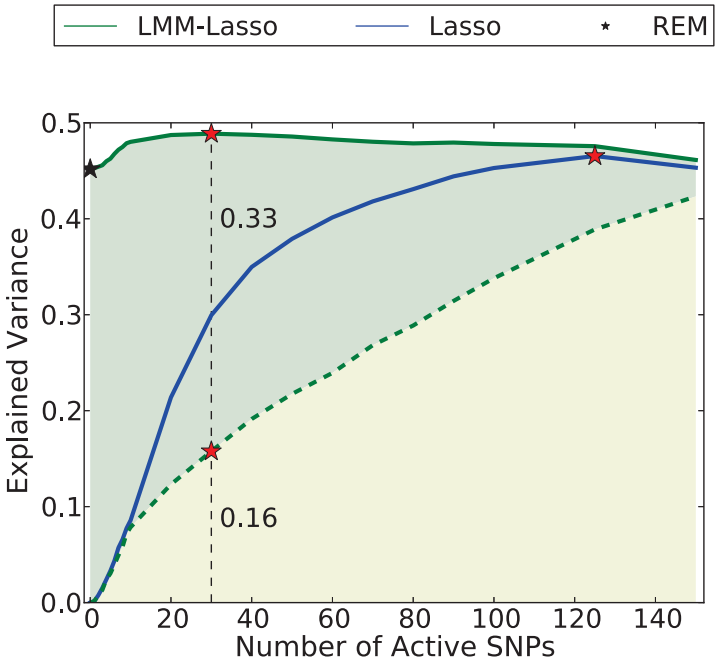



Figure 3.8: **Variance dissection into individual SNP effects and global genetic background driven by population structure.** Shown is the explained variance on an independent test set as a function of the number of active SNPs for the flowering phenotype (10° C) in *Arabidopsis thaliana*. In blue, the predictive test set variance of the Lasso as a function of the number of SNPs in the model. In green, the total predictive variance of LMM-Lasso for different sparsity levels. The shaded area indicates the fraction of variance LMM-Lasso explains by means of individual SNP effects (yellow) and population structure (green). LMM-Lasso without additional SNPs in the model corresponds to a genetic random effect model (black star).

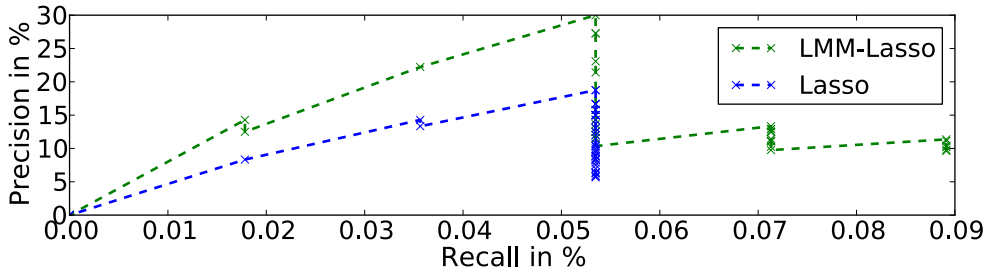



Figure 3.9: **Evaluation of the Lasso methods for FLC gene expression in *Arabidopsis thaliana*.** Precision-Recall Curve for recovering SNPs in proximity to known candidate genes using alternative methods. Shown is precision (TP/(TP+FP)) as a a function of the recall (TP/(TP+FN)). Each point in the plot corresponds to a specific selection threshold.

# Chapter 4

# Incorporating structural information into the Lasso

In the last chapter, we presented an algorithm that increased power by adding a random effect to the Lasso model that can account for population structure. In this chapter, we take a different route and increase power by either leveraging samples over correlated phenotypes or by coupling the effect sizes of markers that are known to be related. Both results in a reduction of the model space and can be combined with an additional random effect if confounding is present.

The idea of coupling regressors of correlated traits is well established and has been used before to induce information sharing across related traits (Knott and Haley, 2000; Jiang and Zeng, 1995; Caruana, 1997; Argyriou et al., 2007). In statistical genetics, the applications of multi-trait models are wide-spread and can roughly be divided into two lines: first, sparse linear models that reward solutions in which the same markers are selected (Kim and Xing, 2009; Lee et al., 2010; Wang et al., 2012), and, second, mixed-model approaches that allow for shared random effects (Henderson, 1984; Price et al., 2011; Korte et al., 2012). In this chapter, we concentrate on different $\ell_1$-regularized models that belong to the group of sparse linear models. The Lasso objective is a flexible one which can be extended by additional regularization terms with ease. For instance, if a set of correlated traits has been recorded, then a multi-task Lasso model is employed to reward solutions in which the same markers are selected over related phenotypes (Kim and Xing, 2009). If gene pathways or other network informations between the markers are available, solutions that select markers concordant with this prior knowledge are preferred (Lee et al., 2009).

In Section 4.1, we give an overview of different Lasso models utilizing input and output prior structure. We also derive a new Lasso model, the AAALasso, that fills

an important gap in the taxonomy of methods using input and output structure. In Section 4.2, we present an efficient inference scheme for estimating its parameters. In our experiments (Section 4.3), we critically assess the advantages and pitfalls of Lasso models that incorporate input and output structure.

## 4.1 Methods overview

In this section, we give an overview of different Lasso models that take relations between markers and phenotypes into account. In its most general form, the structured Lasso problem is to minimize

$$\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_{Fro}^2 + \lambda\|\boldsymbol{W}\|_1 + \lambda_{in}\Omega_{in}(\boldsymbol{W}) + \lambda_{out}\Omega_{out}(\boldsymbol{W}). \tag{4.1}$$

The phenotype matrix $\boldsymbol{Y}$ is a $N \times T$ dimensional matrix, where $N$ is the number of samples and $T$ is the number of traits. $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ is the SNP matrix, where $M$ is the number of SNPs, and $\boldsymbol{W} \in \mathbb{R}^{M\times T}$ is the coefficient matrix to be estimated. With a slight abuse of notation, we write $\boldsymbol{Y}_{n,:}$ to access the $n$th row of the phenotype matrix, and $\boldsymbol{Y}_{:,t}$ for the $t$th column of the phenotype matrix. The first term rewards a solution with accurate prediction of the phenotype (minimal loss), while the second term commonly rewards sparsity in the solution, that is, solutions that set the weights of only few SNPs to non-zero are preferred. $\lambda$ is a trade-off parameter between loss minimization and sparseness. The larger $\lambda$ is, the more markers are set to 0. $\Omega_{in}(\boldsymbol{W})$ and $\Omega_{out}(\boldsymbol{W})$ are regularizers that reward solutions in which related input variables (SNPs) or related output variables (phenotypes) receive similar weights $\boldsymbol{W}$. $\lambda_{in}$ and $\lambda_{out}$ are their respective regularization parameters, which control to which degree our solution should respect input or output structure. If both, $\lambda_{in}$ and $\lambda_{out}$ are zero, the problem reduces to the standard Lasso problem which we discussed in Section 2.1.5. A family of recent studies have investigated different types of regularizers on input and/or output structure, the most prominent ones of which we discuss below.

### 4.1.1 Exploiting input structure

The Group Lasso (Yuan et al., 2006) extends the Lasso in a way that it defines groups of input features, and rewards similar weights for features from the same group by an $\ell_2$-norm regularization. Such groups could be SNPs from genes that are members of the same biochemical pathway. In Group Lasso, $\Omega_{in}(\boldsymbol{W}_{:,t})$ is defined for phenotype

$t$ as

$$\Omega_{in}(\boldsymbol{W}_{:,t}) = \sum_{\boldsymbol{g} \in \mathcal{G}_{in}} \|\boldsymbol{W}_{\boldsymbol{g},t}\|_2, \tag{4.2}$$

where $\mathcal{G}_{in}$ is the set of predefined groups of SNPs. A popular instance of the Group Lasso is the Elastic Net algorithm (Zou and Hastie, 2005), where all SNPs belong to the same group. Within a group, the grouped $\ell_2$-norm allows for effects in opposite directions, but unlike the following approaches, it cannot consider the relatedness between two individual SNPs within a group. Two other models that incorporate input structure into the Lasso problem are the non-convex Fused Graph Lasso (ncFGS) approach by Yang et al. (2012b) and the network-constrained Regularized Lasso by Li and Li (2008). The Fused Graph Lasso model (Yang et al., 2012b) couples input variables via

$$\Omega_{in}(\boldsymbol{W}_{:,t}) = \sum_{i \sim j \in G_{in}} \| |W_{i,t}| - |W_{j,t}| \|_1 \tag{4.3}$$

That is, it assumes that we are given a graph $G_{in}$ between SNPs, e.g. a protein interaction network, and minimizes the difference in the absolute weights of SNPs that are interacting in this network. Due to the use of absolute weights, interacting SNPs can have identical or opposite directions of effect. The network constrained regularized Lasso (Li and Li, 2008) defines $\Omega_{in}$ as:

$$\Omega_{in}(\boldsymbol{W}_{:,t}) = \sum_{i \sim j \in G_{in}} \alpha_{i,j} \left( \frac{W_{i,t}}{\sqrt{d_i}} - \frac{W_{j,t}}{\sqrt{d_j}} \right)^2 \tag{4.4}$$

Here, we again assume that a network between SNPs is given, with edge weights $\alpha_{i,j}$ that represent the strength of the interaction. This regularizer rewards solutions in which SNPs with strong interactions in the graph receive similar weights. These weights are rescaled by the square root of the degree of each SNP $d_i$ in the SNP graph, to allow SNPs that are having more connections, for instance regulatory elements, to have larger coefficients.

In contrast to the other two approaches discussed before, this model assumes that coupled markers have an effect in the same direction (synergistic effect) and does not allow for effects in opposite directions (antagonistic effect). However, synergistic and antagonistic effects of SNPs are both plausible scenarios, as demonstrated in yeast: In a recent study, Jelier et al. (2011) were able to predict yeast growth phenotypes based on genotypes using a simple additive model across affected genes and observed genetic loci with growth-enhancing and inhibiting effects.

### 4.1.2 Exploiting output structure

The models using output structure strive to exploit the fact that in the presence of a collection of several phenotypes, we may know which of these phenotypes are related to each other and search for a possibility to use this knowledge for more accurate variable selection. The first common way of coupling phenotypes is the standard Multi-Task Lasso (MTLasso) from Obozinski et al. (2008). It uses a $\ell_2$-norm to reward solutions in which a SNP receives similar weights across all phenotypes

$$\Omega_{out}(\boldsymbol{W}) = \sum_{m=1}^{M} \|\boldsymbol{W}_{m,:}\|_2 \tag{4.5}$$

If we have more specific information about which phenotypes are correlated, then we can reward solutions in which the weight vector for correlated phenotypes are similar. This model is referred to as Graph-Guided Fused Lasso (FGLasso) (Kim et al., 2009).

$$\Omega_{out}(\boldsymbol{W}) = \sum_{k \sim l \in G_{out}} \|\boldsymbol{W}_{:,k} - \text{sign}(r_{k,l})\boldsymbol{W}_{:,l}\|_1 \tag{4.6}$$

Here, $G_{out}$ is the graph of correlation scores between phenotypes, and $r_{k,l}$ is the correlation between phenotype $k$ and $l$.

### 4.1.3 Exploiting input and output structure

Structured Input Output Lasso (SIOL) by Lee and Xing (2012) and Two-Graph Guided Multitask Lasso (MTLasso2G) by Chen et al. (2012) are two recent Lasso models which take both input and output structure into account. In Lee and Xing (2012), the authors extend the idea of group Lasso in having both groups $\mathcal{G}_{in}$ of SNPs in the input level and groups $\mathcal{G}_{out}$ of phenotypes in the output level. That is, the regularization terms are defined as:

$$\lambda_{in}\Omega_{in}(\boldsymbol{W}) + \lambda_{out}\Omega_{out}(\boldsymbol{W}) = \lambda_{in} \sum_{t=1}^{T} \sum_{\boldsymbol{g} \in \mathcal{G}_{in}} \|\boldsymbol{W}_{\boldsymbol{g},t}\|_2 + \lambda_{out} \sum_{m=1}^{M} \sum_{\boldsymbol{h} \in \mathcal{G}_{out}} \|\boldsymbol{W}_{m,\boldsymbol{h}}\|_2 \tag{4.7}$$

Chen et al. (2012), on the other hand, couples the weights of correlated SNPs on the input level and the weights of correlated phenotypes on the output level:

$$\lambda_{in}\Omega_{in}(\boldsymbol{W}) + \lambda_{out}\Omega_{out}(\boldsymbol{W}) = + \lambda_{in} \sum_{i \sim j \in G_{in}} \alpha_{i,j} \|\boldsymbol{W}_{i,:} - \text{sign}(s_{i,j})\boldsymbol{W}_{j,:}\|_1 + \lambda_{out} \sum_{k \sim l \in G_{out}} \beta_{k,l} \|\boldsymbol{W}_{:,k} - \text{sign}(r_{k,l})\boldsymbol{W}_{:,l}\|_1, \quad (4.8)$$

where $s_{i,j}$ is the Pearson's correlation coefficient between SNP $i$ and $j$, $\alpha_{i,j}$ is the edge weight between SNP $i$ and $j$, $\beta_{k,l}$ is the edge weight between trait $k$ and $l$. For simplicity, the authors (Chen et al., 2012) set the edge weights to the absolute value of the correlation coefficient.

While we agree with the authors that strongly correlated phenotypes are more likely to be caused by the same subset of SNPs, we argue that linkage disequilibrium between two SNPs does not imply functional coherence and propose to use a biological network as input information instead. We propose a novel model called AAALasso, a Lasso model that exploits both structured input and structured output information. For linking the SNPs, we use a regularization term similar to (Yang et al., 2012b), for linking the phenotypes, we use a regularization term similar to (Kim et al., 2009). Although both regularizers have been used before, the combination of both is novel. Our proposed method can handle antagonistic effects between related markers, while at the same time favoring pleiotropic markers that have a shared effect across correlated traits:

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_{Fro}^2 + \lambda\|\boldsymbol{W}\|_1 + \lambda_{in} \sum_{i \sim j \in G_{in}} \| |\boldsymbol{W}_{i,:}| - |\boldsymbol{W}_{j,:}| \|_1 + \lambda_{out} \sum_{k \sim l \in G_{out}} \|\boldsymbol{W}_{:,k} - \text{sign}(r_{k,l})\boldsymbol{W}_{:,l}\|_1 \quad (4.9)$$

Figure 4.1 demonstrates how our proposed model exploits input and output structure for coupling the coefficients to be estimated. In this toy example, the traits $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are correlated to each other and hence their respective coefficients, *i.e.* $W_{i,1}$ and $W_{i,2}$ for SNP $i \in \{1 \dots 8\}$, will be coupled via shrinking the difference between their magnitudes. However, the coefficients of $\boldsymbol{y}_3$ will be optimized independently of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Similarly, the coefficients of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, *i.e.* $\boldsymbol{X}_{:,1}$ and $\boldsymbol{X}_{:,2}$, are coupled (across all outputs) as the blue edge between them implies they have similar effects on the phenotypic traits. In this case, their coefficients will both be pushed to zero as they have no effect on any of the output traits. On the other hand, $W_{4,1}$ and $W_{5,1}$ (resp. $W_{4,2}$ and $W_{5,2}$) will both be non-zero and close to each other in terms of magnitude due to the dashed lines between $\boldsymbol{x}_4, \boldsymbol{x}_5$ and $\boldsymbol{y}_1, \boldsymbol{y}_2$ .

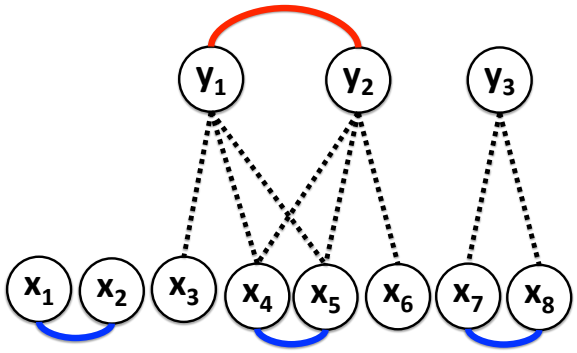



Figure 4.1: **Demonstration of the coupling of input and output.** Red solid edges represent correlations between the phenotypes, blue solid lines represent relational dependencies between the SNPs. A dashed line represents an association between an SNP and a phenotype.

## 4.2 Parameter inference

Solving the optimization problem (4.9) is hindered by the input penalty term that makes the objective non-convex. While in principle, we can still use a similar procedure as proposed in Yang et al. (2012b), the complexity of the algorithm changes from $O(TM^3)$ when solving for each trait individually to $O(T^3M^3)$ when solving jointly which is clearly prohibitive. By exploiting structured sparsity using Kronecker Products, we are able to improve the complexity of the algorithm significantly. In the following, we will first give a very brief outline of the general algorithm and then introduce the new speed-ups. For a more detailed description of the general algorithm, we refer the interested reader to the original paper from Yang et al. (2012b), for an introduction to Kronecker products we refer to the Appendix A.3.

The objective can be recast as an instance of DC (difference of convex functions) programming (Tao and An, 1997) by rewriting the objective as the difference between the two convex functions $f_1(\boldsymbol{W})$, $f_2(\boldsymbol{W})$, where:

$$\begin{aligned} f_1(\boldsymbol{W}) =& \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_{Fro}^2 + \lambda_{out} \sum_{k\sim l \in G_{out}} \|\boldsymbol{W}_{:,k} - \text{sign}(r_{k,l})\boldsymbol{W}_{:,l}\|_1 \\ &+ \lambda_{in} \sum_{i\sim j \in G_{in}} \|\boldsymbol{W}_{i,:} - \boldsymbol{W}_{j,:}\|_1 + \|\boldsymbol{W}_{i,:} + \boldsymbol{W}_{j,:}\|_1 + \lambda\|\boldsymbol{W}\|_1 \\ f_2(\boldsymbol{W}) =& \lambda_{in} \sum_{i\sim j \in G_{in}} \|\boldsymbol{W}_{i,:}\|_1 + \|\boldsymbol{W}_{j,:}\|_1, \end{aligned} \tag{4.10}$$

where we exploit the identity

$$\||\boldsymbol{W}_{i,:}| - |\boldsymbol{W}_{j,:}|\| = \|\boldsymbol{W}_{i,:} - \boldsymbol{W}_{j,:}\|_1 + \|\boldsymbol{W}_{i,:} + \boldsymbol{W}_{j,:}\|_1 - \|\boldsymbol{W}_{i,:}\|_1 - \|\boldsymbol{W}_{j,:}\|_1. \quad (4.11)$$

The problem is solved iteratively by substituting the concave part $-f_2(\boldsymbol{W})$ by its affine minorization until convergence. The affine minimization of the $m$th iteration is defined as

$$f_2^m(\boldsymbol{W}) = f_2(\boldsymbol{W}^m) + \ \mathrm{vec}(\boldsymbol{W} - \boldsymbol{W}^m)^T \mathrm{vec}(\partial f_2(\boldsymbol{W}^m)), \quad (4.12)$$

where $\boldsymbol{W}^m$ is the solution of the previous iteration. In the following, we abbreviate the derivative $\partial f_2$, evaluated at $\boldsymbol{W}^m$, with $\boldsymbol{C}^m = \partial f_2(\boldsymbol{W}^m)$. The resulting subproblem is convex and is solved by alternating direction method of multipliers (ADMM) (Boyd et al., 2011). We first introduce auxiliary variables $\boldsymbol{Z}, \boldsymbol{Z}_{in}, \boldsymbol{Z}_{out}$ to decouple the regularization terms from the loss term in the objective, while coupling them again in the constraints:

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{Z},\boldsymbol{Z}_{in},\boldsymbol{Z}_{out}} \quad & \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_{Fro}^2 + \lambda\|\boldsymbol{Z}\|_1 + \lambda_{in}\|\boldsymbol{Z}_{in}\|_1 + \lambda_{out}\|\boldsymbol{Z}_{out}\|_1 + f_2^m(\boldsymbol{W}) \\ \text{s.t.} \quad & \boldsymbol{W} - \boldsymbol{Z} = \boldsymbol{0},\ \boldsymbol{T}_{in}\boldsymbol{W} - \boldsymbol{Z}_{in} = \boldsymbol{0},\ \boldsymbol{W}\boldsymbol{T}_{out}^T - \boldsymbol{Z}_{out} = \boldsymbol{0}, \end{aligned} \quad (4.13)$$

where $\boldsymbol{T}_{in}, \boldsymbol{T}_{out}$ embed the network constraints: for each edge in the input network, we add two rows in $\boldsymbol{T}_{in}$, one with $+1$ on positions $i$ and $j$, and one with $+1$ on position $i$ and $-1$ on position $j$. For each edge in the output network, we add one row in $\boldsymbol{T}_{out}$, with $+1$ on position $k$ and $\mathrm{sign}(-r_{k,l})$ on position $l$. The remaining entries are set to 0.

ADMM uses the augmented Lagrangian to reformulate the problem as follows

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_{Fro}^2 + \lambda\|\boldsymbol{Z}\|_1 + \lambda_{in}\|\boldsymbol{Z}_{in}\|_1 + \lambda_{out}\|\boldsymbol{Z}_{out}\|_1 \\ & + \mathrm{vec}(\boldsymbol{\mu})^T \mathrm{vec}(\boldsymbol{W} - \boldsymbol{Z}) + \mathrm{vec}(\boldsymbol{\mu}_{in})^T \mathrm{vec}(\boldsymbol{T}_{in}\boldsymbol{W} - \boldsymbol{Z}_{in}) \\ & + \mathrm{vec}(\boldsymbol{\mu}_{out})^T \mathrm{vec}(\boldsymbol{W}\boldsymbol{T}_{out}^T - \boldsymbol{Z}_{out}) + \frac{\rho}{2}\|\boldsymbol{W} - \boldsymbol{Z}\|_{Fro}^2 \\ & + \frac{\rho}{2}\|\boldsymbol{T}_{in}\boldsymbol{W} - \boldsymbol{Z}_{in}\|_{Fro}^2 + \frac{\rho}{2}\|\boldsymbol{W}\boldsymbol{T}_{out}^T - \boldsymbol{Z}_{out}\|_{Fro}^2 + f_2^m(\boldsymbol{W}), \end{aligned} \quad (4.14)$$

where $\rho > 0$ are the step sizes in the dual update, $\boldsymbol{\mu}, \boldsymbol{\mu}_{in}, \boldsymbol{\mu}_{out}$ are the Lagrangian multipliers. We then iteratively solve for $\boldsymbol{\Theta} = \{\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{Z}_{in}, \boldsymbol{Z}_{out}, \boldsymbol{\mu}, \boldsymbol{\mu}_{in}, \boldsymbol{\mu}_{out}\}$ until convergence. Other than $\boldsymbol{W}$, the updates are easy to compute and are as follows:

$$
\begin{aligned}
\boldsymbol{Z}^{k+1} &= S_{\lambda/\rho}\left(\boldsymbol{W}^{k+1} + \frac{1}{\rho}\boldsymbol{\mu}^{k}\right), && (4.15)\\
\boldsymbol{Z}_{in}^{k+1} &= S_{\lambda/\rho}\left(\boldsymbol{T}_{in}\boldsymbol{W}^{k+1} + \frac{1}{\rho}\boldsymbol{\mu}_{in}^{k}\right), && (4.16)\\
\boldsymbol{Z}_{out}^{k+1} &= S_{\lambda/\rho}\left(\boldsymbol{W}^{k+1}\boldsymbol{T}_{out}^{T} + \frac{1}{\rho}\boldsymbol{\mu}_{out}^{k}\right), && (4.17)\\
\boldsymbol{\mu}^{k+1} &= \boldsymbol{\mu}^{k} + \rho\left(\boldsymbol{W}^{k+1} - \boldsymbol{Z}^{k+1}\right) && (4.18)\\
\boldsymbol{\mu}_{in}^{k+1} &= \boldsymbol{\mu}_{in}^{k} + \rho\left(\boldsymbol{T}_{in}\boldsymbol{W}^{k+1} - \boldsymbol{Z}_{in}^{k+1}\right) && (4.19)\\
\boldsymbol{\mu}_{out}^{k+1} &= \boldsymbol{\mu}_{out}^{k} + \rho\left(\boldsymbol{W}^{k+1}\boldsymbol{T}_{out}^{T} - \boldsymbol{Z}_{out}^{k+1}\right) && (4.20)\\
& && (4.21)
\end{aligned}
$$

where $S_{\lambda/\rho}$ is the soft-thresholding operator defined as follows (Rockafellar, 1970):

$$S_{\lambda}(\boldsymbol{W}) = \text{sign}(\boldsymbol{W}) - \max\left(|\boldsymbol{W}| - \lambda, 0\right) \tag{4.22}$$

Solving for the coefficients $\boldsymbol{W}^{k+1}$ in iteration $k$ requires to solve the linear system $\boldsymbol{b}^{k} = \boldsymbol{K}\text{vec}(\boldsymbol{W}^{k+1})$, where

$$
\begin{aligned}
\boldsymbol{K} &= (\boldsymbol{I}\otimes\boldsymbol{X})^{T}(\boldsymbol{I}\otimes\boldsymbol{X}) + \rho\left(\boldsymbol{I}\otimes\boldsymbol{T}_{in}\right)^{T}\left(\boldsymbol{I}\otimes\boldsymbol{T}_{in}\right) + \rho\left(\boldsymbol{T}_{out}\otimes\boldsymbol{I}\right)^{T}\left(\boldsymbol{T}_{out}\otimes\boldsymbol{I}\right) + \rho\boldsymbol{I}\\
\boldsymbol{b}^{k} &= \text{vec}(\boldsymbol{X}^{T}\boldsymbol{Y}) + \text{vec}\left(\boldsymbol{C}^{m}\right) + \text{vec}\left(\rho\boldsymbol{Z}^{k} - \boldsymbol{\mu}^{k}\right) + \text{vec}\left(\boldsymbol{T}_{in}^{T}(\rho\boldsymbol{Z}_{in}^{k} - \boldsymbol{\mu}_{in}^{k}\right)\\
&\quad + \text{vec}\left((\rho\boldsymbol{Z}_{out}^{k} - \boldsymbol{\mu}_{out}^{k})\boldsymbol{T}_{out}\right), && (4.23)
\end{aligned}
$$

While the authors in Yang et al. (2012b) propose to compute one Cholesky factorization in the beginning to speed-up the subsequent iterations, computing the factorization is prohibitive in our setting as its runtime complexity is $O(T^3M^3)$. Instead, we propose to solve the linear system by using conjugate gradients (Shewchuk, 1994). Conjugate gradient (CG) is an iterative method for solving linear systems efficiently. Specially, if the matrix-vector product can be computed within acceptable time and memory resources, as we do by exploiting the Kronecker structure, the speed-up can be significant:

$$
\begin{aligned}
\boldsymbol{K}\text{vec}(\boldsymbol{W}) = &\ \text{vec}(\boldsymbol{X}^{T}\boldsymbol{X}\boldsymbol{W}) + \text{vec}(\boldsymbol{T}_{in}^{T}\boldsymbol{T}_{in}\boldsymbol{W}) + \text{vec}(\boldsymbol{W}\boldsymbol{T}_{out}\boldsymbol{T}_{out}^{T}) + \rho\text{vec}(\boldsymbol{W})\\
&. && (4.24)
\end{aligned}
$$

Using the Kronecker speed-ups reduces the runtime for computing the matrix-vector product from $O(T^2M^2)$ to $O(T^2M + M^2T)$ and reduces the memory complexity from

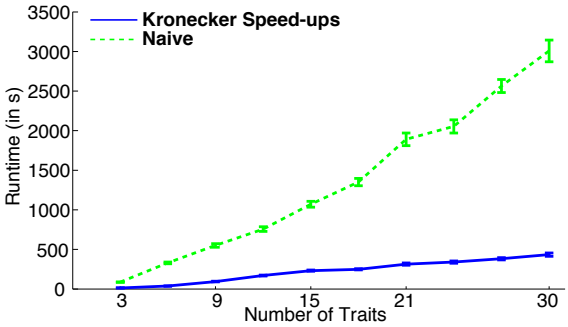



Figure 4.2: **Runtime comparison for varying number of traits.** The naive method, based on a Cholesky factorization, is shown in green (dashed), the proposed method, exploiting the Kronecker structure, is shown in blue (solid). Shown is the averaged runtime (in seconds) and its standard error as a function of the number of traits.

$O(M^2T^2)$ to $O(M^2 + T^2)$. If the number of SNPs is large, computing the $M \times M$ matrices is still prohibitive. For $\boldsymbol{X}^T\boldsymbol{X}$ we can exploit its low-rank structure, since the number of samples is usually small compared to the number of SNPs, decreasing the runtime again to $O(MNT)$ and the memory requirements to $O(NM+MT+NT)$ by never explicitly computing the outer product $\mathbf{X^T X}$, but $\boldsymbol{X}^T(\boldsymbol{XW})$. While we cannot assume that the matrices $\boldsymbol{T}_{in}, \boldsymbol{T}_{out}$ have low-rank, we can exploit the fact that they are very sparse (having only two non-zero entries per row).

We measured the runtime of our algorithm on a MacPro (8 cores, 2GHz, 12MB L2-Cache, 16GB Memory) for a varying number of traits $\{3, 6, \ldots, 30\}$, while keeping the number of SNPs and samples fixed ($P = 500, N = 100$). Subsequently, we compared our method to a naive algorithm that computes a one-time Cholesky factorization of $\boldsymbol{K}$ for solving the ridge regression subproblems. The corresponding run times are shown in Figure 4.2, confirming the significantly improved scalability of our method.

## 4.3 Experiments

In this section, we compare different Lasso approaches in extensive simulation studies and in an eQTL study that aims to determine variants affecting gene expression under nutrient limiting response in yeast (Smith and Kruglyak, 2008). For comparison, we selected a comprehensive, publicly available set of six algorithms and our newly defined method AAALasso. As a baseline, we considered the standard Lasso (Tibshirani, 1994). The family of methods using input structure only is represented by the non-convex Fused Graph Lasso approach (ncFGS) (Yang et al., 2012b). The Multi-Task Lasso (MTLasso) (Obozinski et al., 2008) and the Graph-Guided Fused Lasso (FGLasso) (Kim et al., 2009) are representatives of the methods using output structure only. The two very recent models MTLasso2G (Chen et al., 2012),

SIOL (Lee and Xing, 2012) and the AAALasso use input as well as output structure. FGLasso and ncFGS are both special instances of our proposed method. For the standard Lasso and MTLasso, we used the freely available MATLAB toolbox of Tomioka et al. (2011).

### 4.3.1 Simulations

The evaluation of the different methods on real data is complicated by the fact that reliable ground truth information is missing. To systematically evaluate the aforementioned algorithms, we conducted experiments on synthetic datasets with known ground truth first.

**Data** We generated datasets with a $N = 100$ samples, $M = 500$ SNPs and $T = 3$ traits each. The SNPs are bi-allelic, each of the two alleles is binary and the minor allele frequency ranges between 0.05 and 0.45. For each trait, 3% of all SNPs are selected as causal, having a non-zero weight, while the remaining SNP weights are set to 0. We implemented a simple scheme to simulate shared SNP effects, which in turn gives rise to pleiotropy and hence to correlated outputs. We chose the SNPs and their coefficients for the first two traits $T_1, T_2$ independently. The third trait $T_3$ is a combination of the former two having $\alpha\%$ of its causal SNPs inherited from the first trait, the remaining $(100 - \alpha)\%$ causal SNPs are derived from the second trait. The overlap parameter $\alpha$ varies in $\{0, 10, 20, 30, 40, 50\}$. By varying $\alpha$, we were able to generate arbitrary correlation dependencies between the traits. For example, if $\alpha = 0$, $T_3$ has the same causal SNPs as $T_1$ and is independent of $T_2$. On the other end of the spectrum, when $\alpha = 50$, $T_3$ gets half of its causal SNPs from $T_1$, and half of $T_2$, being equally correlated with the two traits. The intermediate $\alpha$ values result in various other dependency structures and thereby provide a comprehensive set of cases that one might experience in real experimental data.

The coefficients of $T_1$ and $T_2$ are randomly drawn from one of the four Gaussian distributions $N(\mu = \pm 1, \sigma = 0.05)$, $N(\mu = \pm 0.7, \sigma = 0.05)$. This in turn means several loci are affecting the trait in an additive manner with similar magnitude. By allowing positive and negative effect sizes of similar magnitudes, we simulate antagonistic additive effects. Next, the coefficients of $T_3$ are adopted from the source traits and a small amount of noise is added.

Subsequently, we added some noise to the simulated phenotypes, such that the explained variances lies between 0.85 and 0.95 when knowing the *true* coefficients.

Finally, we simulated the input network by adding 50 edges between causal SNPs, 500 edges between non-causal SNPs and 10 conflicting edges between causal and non-

| | Simulation | | | Glucose | | | Glucose vs. Ethanol | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | $\lambda$ | $\lambda_{in}$ | $\lambda_{out}$ | $\lambda$ | $\lambda_{in}$ | $\lambda_{out}$ | | | |
| STDLasso | 1.0000 | - | - | 4.0000 | - | - | 4.0000 | - | - |
| MTLasso | 4.0000 | - | - | 16.0000 | - | - | 16.0000 | - | - |
| FGLasso | 1.0000 | - | 1.0000 | 4.000 | - | 1.0000 | 4.0000 | - | 1.0000 |
| ncFGS | 1.0000 | 0.0625 | - | 4.000 | 0.0156 | - | 4.0000 | 0.0625 | - |
| MTLasso2G | 0.2500 | 0.0625 | 0.0039 | 0.2500 | 0.0039 | 0.0039 | 0.0625 | 0.0039 | 0.0039 |
| eSIOL | 0.0312 | 0.0156 | 0.0312 | 0.0078 | 0.0078 | 0.0312 | 0.0117 | 0.0117 | 0.0312 |
| AAALasso | 1.0000 | 0.2500 | 1.0000 | 4.0000 | 0.0039 | 0.2500 | 4.0000 | 0.0039 | 1.0000 |

Table 4.1: **Chosen regularization parameters.** For each method, we show the median of the regularization parameters determined by cross-validation. All regularization parameters lie within the chosen intervals or on the left-boundary of the interval. For all methods except SIOL we used the set $\{4^{-4}, 4^{-3}, 4^{-2}, 4^{-1}4^{0}, 4^{1}, 4^{2}, 4^{3}\}$. For SIOL, we changed the range to $\{2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}\}$ since the data is internally normalized.

causal SNPs, which amounts to noise input network. Using this scheme, we generated 300 datasets, 50 for each value of $\alpha$. To ensure that our input network is topologically close to real gene/protein networks, the input network is generated as a scale-free network using the iGraph (Csardi and Nepusz, 2006) package of R.

**Experimental Setting** We used 5-fold cross validation with a grid search scheme to choose regularization parameters that minimize mean squared error (MSE) for all algorithms. For each method, we optimized the applicable regularization parameters $\lambda$, $\lambda_{in}$, $\lambda_{out}$. All regularization parameters can take one of the values in $\{4^{-4}, 4^{-3}, 4^{-2}, 4^{-1}, 4^{0}, 4^{1}, 4^{2}, 4^{3}\}$. Since SIOL (Lee and Xing, 2012) uses an internal normalization step in its training procedure, we changed the range to $\{2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}\}$ and learned a least squares estimator on the non-zero coefficients on top. Since we do not account for epistasis, we set the corresponding parameter of SIOL to 0. In Table 4.1, the median of the chosen regularization parameters is shown for each method assuring that the chosen intervals are sensitive.

We defined an output edge, if the absolute correlation between the two traits is larger than 0.4 (needed for MTLasso2G and AAALasso). In addition, SIOL requires group definitions on the input and output level. For this we used the connected components of the trait correlation network as the output groups and the edges of the input network for the input groupings as suggested by the authors (Seunghak Lee, personal communication, 2013).

We measured the performance of algorithms in terms of (a) detecting SNPs with

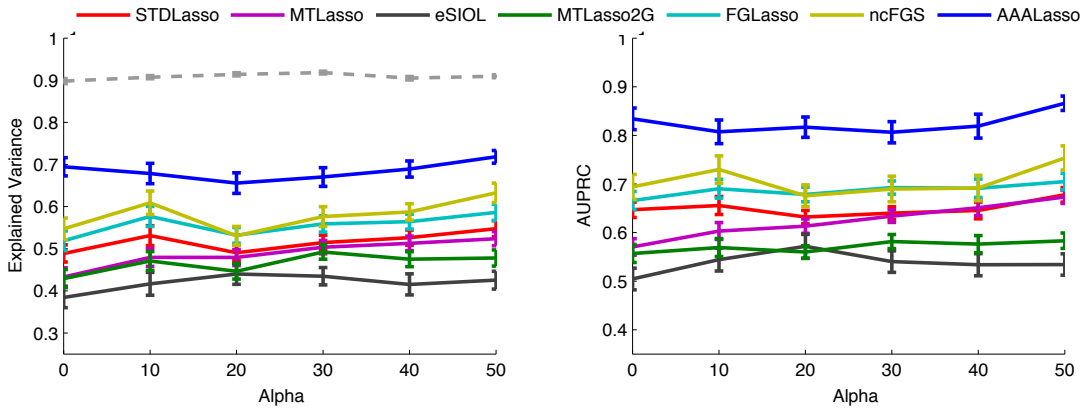



Figure 4.3: **Performance of all algorithms on simulated data.** Left: Comparing the methods in terms of predictive power. Shown is the Explained Variance (EV) as a function of $\alpha$. The dashed gray line represents the upper bound, *i.e.* EV obtained when using true co-efficient matrix. Right: Area under Precision recall curve (AUPRC) for recovering the simulated associations as a function of the overlap parameter $\alpha$. The error bars represent the standard error.

a true simulated effect and (b) in terms of predictive power. To measure the power to recover the true coefficient matrix $\boldsymbol{\beta}$, we calculated the Area Under Precision Recall Curve (AUPRC) for the non-zero coefficients. The precision-recall curve shows the precision (TP/(TP+FP)) as a function of the recall (TP/(TP+FN)), where TP are the True Positives, FP the False Positives and FN the False Negatives. For measuring the predictive power, we used the explained variance which is defined as the squared Pearson Correlation coefficient between the true and the predicted phenotype, averaged over the traits.

**Results** We display the results for the different methods for varying $\alpha$ in Figure 4.3. In general, we depict that methods that make correct use of either/both structural information perform better than those that do not. Methods that do not incorporate structural information in turn perform better than methods that make incorrect use of structural information.

While our simulation setting assumes that prior information is given via networks, SIOL is designed for grouping information. We tried various different network-group transformations (data not shown here), and continued with the one working best. The input/output structure in our scenario is complex allowing for overlapping groups and groups of all different sizes. As discussed in Kim and Xing (2010), this can cause to imbalance between the different SNPs if not accounted for and leads

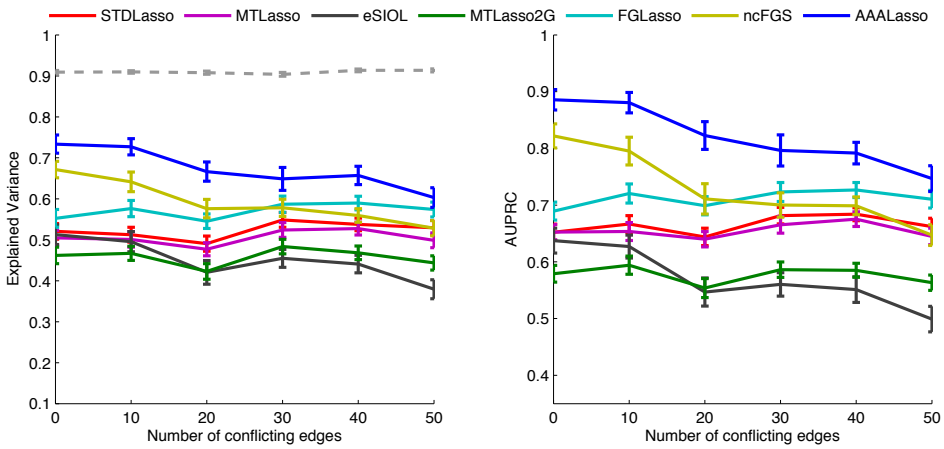



Figure 4.4: **Power comparison for varying input noise.** We fixed the overlap parameter $\alpha = 0.3$ and varied the number of conflicting edges between 0 and 50. For each setting, we generated 50 datasets. The performance of all methods that are exploiting input structure is dropping as the number of conflicting edges is increasing, while the performance of the other methods stays constant.

to a loss of power. In contrast, the other structural methods shrink the difference between the weight coefficients and not their magnitude.

FGLasso and M2Lasso2G have the same output penalty term, while M2Lasso2G has an additional input penalty term. Since the input penalty does not allow for antagonistic effects, but rather minimizes the difference between the coefficients of two connected SNPs, it performs worse than FGLasso which can flexibly set the coefficients of all SNPs. ncFGS, which has an input penalty term that supports antagonistic effects, also reaches high levels of precision/recall and predictive power. FGLasso can account for two strongly correlated traits (small $\alpha$) as well as for three mildly correlated traits (large $\alpha$). Our proposed method, AAALasso, consistently outperforms all competitors as it combines the advantages of ncFGS and FGLasso.

We also performed simulations to measure robustness of the algorithms under noisy input priors (see Figure 4.4). Our results suggest that the AAALasso, and its submodels, can cope with a modest amount of input noise.

### 4.3.2 eQTL study in yeast

In our next experiment, we assessed the performance of the methods AAALasso, STDLasso, MTLasso, MTLasso2G and SIOL on the yeast eQTL dataset from Smith

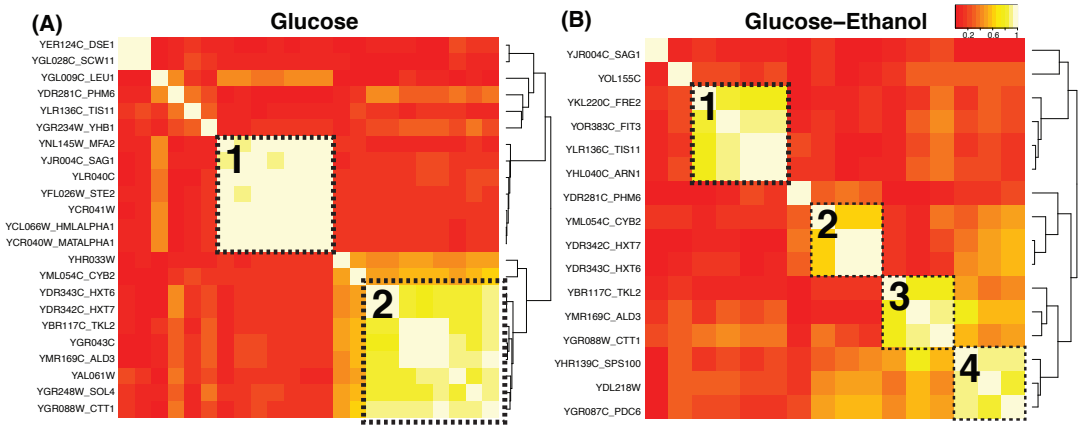



Figure 4.5: **Hierarchical clustering of the gene expression levels under glucose treatment (left) and of the difference between gene expression levels under ethanol and glucose treatments (right).** Under the glucose condition, genes can be divided into two major clusters of co-expressed genes, whereas on the right, the co-expression pattern is more complex containing two major co-expression clusters, one of which contains three smaller clusters that are weakly co-expressed with the other small clusters. For the EG experiment, we focused on cluster 2, for the DCS experiment on cluster 2 and 4.

and Kruglyak (2008).

**Data** The dataset contains gene expression levels of 5637 genes and 1260 unique SNPs (out of 2956 SNPs) for 109 yeast strains. The strains are obtained by crosses of the parental strains BY and RM. The main goal of the original eQTL study of Smith and Kruglyak (2008) was to uncover associations between genetic loci and gene expression under different environmental conditions (glucose or ethanol as the sole carbon source).

We conducted two experiments: in the first one, we studied gene expression levels related to metabolism under the glucose condition, in the second one we studied gene expression levels related to metabolism that are differentially expressed under the glucose and ethanol condition. From this point on, we refer to the first experiment as **EG**, which stands for excessive glucose, and to the second one as **DCS**, which stands for differential carbon source.

We first selected differentially expressed genes by filtering out genes with standard deviation less than 1.5. Then, we clustered the remaining differentially expressed genes using a simple hierarchical clustering scheme with complete linkage. As Fig-

ure 4.5 (A) shows, we obtained two relatively large clusters of co-expressed genes in the EG experiment: Cluster-2 is related to carbon and energy metabolism, whereas Cluster-1 contains genes involved in mating. Since the main findings of Smith and Kruglyak (2008) are related to growth under changing carbon resources, we concentrated on the metabolism cluster, containing eight genes (listed in Table 4.2), in our analysis. Since the pairwise correlation coefficients in the cluster are strong ($> 0.8$), we used a fully connected graph as output network.

In the DCS experiment, we found four small clusters of co-expressed genes. Cluster-2,3 and 4 contain genes related to carbon metabolism, stress response and respiration regulation, while the Cluster-1 contains genes related to iron transport. We again focused on carbon related clusters but this time we only picked Cluster-2 and Cluster-4 to obtain a more complicated output structure. We coupled all traits that have an absolute correlation threshold of 0.5 or larger, resulting in a more complex output structure compared to the EG experiment.

For the input network, we linked each SNP with its nearest gene within a 2Kb neighborhood and constructed an edge if the corresponding genes are interacting. We selected the top 100K interactions from each of the yeast function networks String (Franceschini et al., 2013) and GeneMania (Warde-Farley et al., 2010) resulting in 2221 interactions for the SNPs under observation.

**Experiments** We learned the best tuning parameters with a 5-fold cross validation scheme for each algorithm, as described in the previous section. We randomly split the data into 80% for training and 20% for testing and repeated this procedure 100 times. Similar to Meinshausen and Bühlmann (2010), we counted all SNPs as Positives that were turned on, i.e. non-zero coefficients, in at least 80 runs.

**Known associations** IRA2 is a GTPase-activating protein that negatively regulates the RAS-cAMP pathway, a key component of the cellular response to glucose (Broach, 1991). Smith and Kruglyak (2008) reported a strong association between the gene *IRA2* and the expression of energy metabolism and growth related transcripts. The *IRA2* region is highly polymorphic containing dozens of SNPs and and is recovered by all methods in both experiments. In addition, Smith and Kruglyak (2008) report a weakly significant association with ethanol induced differential genes and *CIN5*, a transcription factor that mediates pleiotropic drug resistance and salt tolerance and has a function in oxidative stress. This gene contains several SNPs including two non-synonymous ones and is found by all methods in the glucose and ethanol comparison experiment as well. This finding suggests that all methods successfully recover the two main effectors.

| Exp. | ORF | Gene | Function |
|---|---|---|---|
| EG | YDR343C | HXT6 | High-affinity glucose transporter; nearly identical to Hxt7p; expressed at high basal levels relative to other HXTs, repression of expression by high glucose |
| | YDR342C | HXT7 | High-affinity glucose transporter, member of the major facilitator superfamily; expression repressed by high glucose levels |
| | YBR117C | TKL2 | Transketolase, together with transaldolase phosphate creates a reversible link between the pentose phosphate pathway and glycolysis |
| | YGR043C | | Transaldolase of unknown function involved in diauxic shift |
| | YMR169C | ALD3 | Cytoplasmic aldehyde dehydrogenase, involved in beta-alanine synthesis; expression is induced by stress and repressed by glucose |
| | YAL061W | | Cytoplasmic aldehyde dehydrogenase, whose expression is induced by stress and repressed by glucose |
| | YGR248W | SOL4 | 6-phosphogluconolactonase; protein abundance increases in response to DNA replication stress |
| | YGR088W | CTT1 | Cytosolic catalase T, which is involved in hydrogen peroxide detoxification and oxidative stress response |
| DGS | YML054C | CYB2 | Cytochrome b2 (L-lactate cytochrome-c oxidoreductase), required for lactate utilization; expression is repressed by glucose and anaerobic conditions |
| | YDR343C | HXT6 | High-affinity glucose transporter; nearly identical to Hxt7p; expressed at high basal levels relative to other HXTs, repression of expression by high glucose |
| | YDL218W | | Putative protein of unknown function; YDL218W transcription is regulated by Azf1p and induced by starvation and aerobic conditions |
| | YDR342C | HXT7 | High-affinity glucose transporter, member of the major facilitator superfamily; expression repressed by high glucose levels |
| | YHR139C | SPS100 | Protein required for spore wall maturation; expressed during sporulation |
| | YGR087C | PDC6 | Minor isoform of pyruvate decarboxylase, decarboxylates pyruvate to acetaldehyde, involved in amino acid catabolism; transcription is glucose- and ethanol-dependent |

Table 4.2: **Phenotypes used in the yeast experiments.** Members of the co-expression cluster used in the glucose (top) and glucose vs. ethanol (bottom) experiments. The classification to functions were derived from Saccharomyces Genome Database (Cherry et al., 2012).

| Experiment | Algorithm | Antagonistic: $\frac{A}{A+S}$ | Synergistic: $\frac{S}{A+S}$ |
|---|---|---|---|
| EG | AAALasso | 0.43 | 0.57 |
| | FGLasso | 0.40 | 0.60 |
| | MTLasso | 0.42 | 0.58 |
| | MTLasso2G | 0.35 | 0.65 |
| | STDLasso | 0.44 | 0.56 |
| | eSIOL | 0.43 | 0.57 |
| | ncFGS | 0.49 | 0.51 |
| DGS | AAALasso | 0.57 | 0.43 |
| | FGLasso | 0.56 | 0.44 |
| | MTLasso | 0.55 | 0.45 |
| | MTLasso2G | 0.27 | 0.73 |
| | STDLasso | 0.55 | 0.45 |
| | eSIOL | 0.56 | 0.44 |
| | ncFGS | 0.54 | 0.46 |

Table 4.3: **Frequencies of categories of antagonistic and synergistic effects among interacting pairs of active SNPs.**

**Antagonistic effects** We analyzed the prevalence of antagonistic effects among the co-selected SNPs. We again focused on the top-100 active SNPs of each method and counted the ratios between synergistic and antagonistic pairs.

For each method, we went over the input network edges whose SNPs are selected and compared the SNP weights for the given edge. We then classified the edges as synergistic (S), if both SNPs are active and the coefficients have the same direction, and antagonistic (A), if both SNPs are active and the coefficients have opposite directions. We then derived ratios antagonistic/synergistic interactions among active edges as shown in Table 4.3. All algorithms, except of MTLasso2G, find abundant antagonistic effects. Moreover, the ratio of antagonistic effects is similar between our proposed method and the standard methods (STDLasso, MTLasso) that do not use any input structure suggesting that antagonistic effects are as common as synergistic ones. MTLasso2G significantly deviates from all other methods as it assumes in the model that all interactions are synergistic (see Table 4.3). We believe this assumption is the core reason for its overall poor performance.

Next, we checked if the gene IRA2 has an antagonistic effector in the glucose condition. We report an antagonistic relation if the two genes are co-selected in at least half of the runs. The methods AAALasso, ncFGS and MTLasso2G recovered an antagonistic association between *IRA2* and *GPB2*. The gene *GPB2* is a multi-

| Algorithm | EG | DCS |
|---|---|---|
| AAALasso | $0.40 \pm 0.014$ | $0.36 \pm 0.010$ |
| FGLasso | $0.43 \pm 0.014$ | $0.34 \pm 0.010$ |
| ncFGS | $0.36 \pm 0.014$ | $0.30 \pm 0.009$ |
| eSIOL | $0.25 \pm 0.012$ | $0.22 \pm 0.008$ |
| MTLasso2G | $0.32 \pm 0.014$ | $0.31 \pm 0.010$ |
| MTLasso | $0.44 \pm 0.014$ | $0.39 \pm 0.010$ |
| STDLasso | $0.42 \pm 0.013$ | $0.34 \pm 0.009$ |

Table 4.4: **Predictive power analysis.** Explained variance ($\pm$ standard error) achieved by all algorithms in both datasets.

step regulator of the cAMP-PKA signalling pathway, which has major roles in the regulation of metabolism, stress resistance and cell cycle progression. In Phan et al. (2010), it is shown that the *GPB2* protein binds and negatively regulates *IRA2* by promoting its ubiquitin-dependent proteolysis.

**Predictive power** We also investigated the predictive power of the different approaches by measuring the explained variance (see Table 4.4). MTLasso performs best over both datasets, followed by the standard Lasso, FGLasso and the AAALasso. In the following, we go over the different approaches and relate their performance to the MTLasso and the standard Lasso, depending if the method is a single or multi-trait approach.

The MTLasso assumes that all tasks are correlated, but does neither put any restrictions on the magnitude of the coefficients nor on their directions. In contrast to eSIOL, the groups cannot be overlapping, preventing that different SNPs are differently penalized.

ncFGS extends the standard Lasso by coupling the weights of markers that are connected via the input network. In both experiments, the predictive performance deteriorates by incorporating the input structure. Here, we give two potential explanations for this: First, the input network is likely to be noisy resulting in linked markers that do not have a joint effect on the phenotype. Second, the assumption that the weights of connected markers are equal in magnitude might be violated.

While MTLasso and FGLasso are both multi-trait approaches, FGLasso attempts to model the phenotypic relatedness at a finer level. In the EG experiment, we used a fully connected graph as output network, as the pairwise correlation coefficients are strong. Both methods perform roughly equally, which shows that both regularization schemes work well in the case of strong genetic correlations. On the DCS experi-

ment, however, the MTLasso outperforms the FGLasso. This can be attributed to the following: First, the phenotypic correlations are less pronounced, making it less likely that a SNP has the same effect size on both phenotypes. Second, we coupled two traits dependent on their phenotypic correlation assuming that phenotypic correlation can be used as an proxy for the genetic correlation. However, shared hidden confounding factors can also lead to spurious phenotypic correlations (Listgarten et al., 2010; Fusi et al., 2012), inducing false edges in the output network.

Since the AAALasso is a combination of the two methods above, their discussion applies here as well: The assumption that the weight coefficients have the same magnitude might be too strong, and both, the input and the output network, might be noisy. Finally, MTLasso2G is significantly outperformed by the standard Lasso in both experiments, which is most possibly caused by its restriction to synergistic effects, as we discussed above.

## 4.4 Summary

In this chapter, we have compared different Lasso models that utilize input and output structure. In simulations, we could verify that incorporating prior knowledge improves the predictive performance of standard Lasso methods, if its assumptions are fulfilled. However, on the yeast dataset, a loose coupling of the weight vectors across correlated phenotypes worked best for both experiments. Methods that use biological knowledge to couple the coefficients across markers did not improve the predictive performance, neither did methods that couple the magnitudes of the weight coefficients across traits.

A fundamental challenge of all approaches using input structure is that they rely on incomplete and noisy prior information (von Mering et al., 2002). Unfortunately, the same also applies to approaches using output structure: We cannot infer which phenotypes are genetically correlated by looking at the phenotypic correlations only. Instead, we have to account for hidden confounding and learn the genetic correlations to reduce the number of false edges that are induced by spurious correlations. In addition, coupling the magnitudes of the weight vector assumes that the same amount of the phenotypic variance can be explained by the genetic markers. This assumption is for instance violated if the phenotypes are measured in different scales. In the following chapter, we will present a method that learns the genetic correlations between the phenotypes while accounting for hidden confounding factors, resolving the drawbacks of the methods studied in this chapter.

# Chapter 5

# Scalable multi-trait models

In the last chapter, we compared different multi-trait models that increase power by leveraging samples over related phenotypes. One of the main drawbacks of the studied methods has been their underlying assumption that the genetic relatedness between the different phenotypes is known *a-priori*.

Here, we propose a multi-task Gaussian process approach that instead learns the genetic relatedness between the phenotypes, as well as the relatedness between the residuals. Conceptually similar methods have been proposed before, but they either use a simpler noise model (Bonilla et al., 2008; Stegle et al., 2011) or lack efficient inference (Henderson, 1984): While the naive approach has a cubic runtime in the number of samples and in the number of traits $O(N^3 \cdot T^3)$ and a quadratic memory requirement $O(N^2 \cdot T^2)$, our reformulation reduces the runtime to $O(N^3 + T^3)$ and the memory requirement to $O(N^2 + T^2)$, making the analysis of large number of phenotypes and sample cohorts feasible. In statistical genetics, applications of this class of models are wide-spread and include amongst others cross-heritability estimation (Deary et al., 2012), phenotype prediction (Jia and Jannink, 2012) and finding pleiotropic effects (Korte et al., 2012).

In Section 5.1, we first show how our model can be derived from the standard linear model. In Section 5.2, we provide an efficient inference scheme for parameter estimation and out-of-sample prediction. Our experiments in Section 5.3 confirm the benefits of our model on simulated data as well as on genome-wide data from *Arabidopsis thaliana* and an eQTL study in yeast.

## 5.1 From linear models to multi-task Gaussian processes

In the following, we derive the multi-task Gaussian process regression model with correlated noise from standard linear regression (see also Section 2.1). Let $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ denote the $N \times T$ phenotype matrix for $N$ samples and $T$ traits. A column of this matrix corresponds to a particular phenotype $t$ and is denoted as $\boldsymbol{y}_t$. Its outputs are determined by a linear function of the markers $\boldsymbol{X} \in \mathbb{R}^{N \times M}$

$$\boldsymbol{y}_t = \underbrace{\boldsymbol{X}\boldsymbol{w}_t}_{\text{genetic factors}} + \underbrace{\boldsymbol{\epsilon}_t}_{\text{noise}}. \tag{5.1}$$

We can stack the weight vectors of the single traits $\boldsymbol{w}_t \in \mathbb{R}^M$ to the matrix $\boldsymbol{W} = \begin{pmatrix} \boldsymbol{w}_1 & \dots & \boldsymbol{w}_T \end{pmatrix} \in \mathbb{R}^{M \times T}$ and the noise vectors $\boldsymbol{\epsilon} \in \mathbb{R}^N$ to the matrix $\boldsymbol{E} = \begin{pmatrix} \boldsymbol{\epsilon}_1 & \dots & \boldsymbol{\epsilon}_T \end{pmatrix} \in \mathbb{R}^{N \times T}$. Multi-task sharing is then achieved by specifying a multivariate normal prior across traits, both for the regression weights $W_{mt}$ and the noise variances $E_{nt}$:

$$p(\boldsymbol{W}^\top) = \prod_{m=1}^{M} \mathcal{N}\left(\boldsymbol{w}_m \,|\mathbf{0}, \boldsymbol{C}_{TT}\right) \quad p(\boldsymbol{E}^\top) = \prod_{n=1}^{N} \mathcal{N}\left(\boldsymbol{\epsilon}_n \,|\mathbf{0}, \boldsymbol{\Sigma}_{TT}\right).$$

Marginalizing out the weights $\boldsymbol{W}$ and the residuals $\boldsymbol{E}$ results in a matrix-variate normal model with sum of Kronecker products covariance structure

$$p(\text{vec } \boldsymbol{Y} | \boldsymbol{C}, \boldsymbol{R}, \boldsymbol{\Sigma}) = \mathcal{N}\left(\text{vec}\boldsymbol{Y}_{NT} \,\middle|\, \mathbf{0}, \underbrace{\boldsymbol{C}_{TT} \otimes \boldsymbol{R}_{NN}}_{\text{genetic signal}} + \underbrace{\boldsymbol{\Sigma}_{TT} \otimes \boldsymbol{I}_{NN}}_{\text{noise}}\right), \tag{5.2}$$

where $\text{vec}\boldsymbol{Y} = \left(\boldsymbol{y}_1^\top \dots \boldsymbol{y}_T^\top\right)^\top$ denotes the vector obtained by vertical concatenation of all columns of $\boldsymbol{Y}$, and $\boldsymbol{R}_{NN} = \boldsymbol{X}\boldsymbol{X}^\top$ is the sample-sample covariance matrix that results from the marginalization over the weights $\boldsymbol{W}$ in Eqn. (5.1). An introduction to the Kronecker product is given in the Appendix A.3.

The diagonal elements of $\boldsymbol{C}$ hereby represent the strength of the genetic signal for the individual traits, the diagonal elements of $\boldsymbol{\Sigma}$ the magnitude of the noise variance for the individual traits. The off-diagonal elements of $\boldsymbol{C}$ show how strongly the traits change together depending on their genetic similarity. The interpretation of these coefficients is dependent whether the individuals of the dataset are related or not: if the samples in the dataset are unrelated, the covariance matrix $\boldsymbol{R}$ can be seen as a proxy for the tagged causal variants, and genetic variance refers to the

aggregation of the tagged causal effects that are shared over the traits (Lee et al., 2012). Contrary, if the samples are related, the genetic variance does not only absorb the causal variants that are in linkage disequilibrium with the markers, but also with untagged causal variants as the markers are correlated over the complete genome (Vattikuti et al., 2012; Visscher et al., 2010). In practice, neither the form of the genetic trait-trait covariance matrix $\boldsymbol{C}$ nor the form of noise trait-trait covariance matrix $\boldsymbol{\Sigma}$ are known *a priori* and hence both need to be inferred from the data, fitting a set of corresponding covariance parameters $\boldsymbol{\theta}_C$ and $\boldsymbol{\theta}_\Sigma$.

In the following, we will refer to a Gaussian process model with this type of sum of Kronecker products covariance structure as GP-kronsum[1]. As common to any kernel method, the linear covariance $\boldsymbol{R}$ can be replaced with any other positive semi-definite covariance function. In statistical genetics, other common choices besides the linear kernel would be the identity by descent matrix or the identity by state matrix (see Section 3.2.2). Note, that we could have equivalently derived our model by using a standard Gaussian process over vec$\boldsymbol{Y}$ assuming a product covariance function for the genetic signal and the noise. Further, we assume here that the mean function is zero - an assumption that is often made for Gaussian processes. However, extensions to non-zero mean functions are straightforward and can for instance be used for accelerated association testing for correlated traits (Korte et al., 2012; Zhou and Stephens, 2013).

In machine learning, work proposing this type of multi-trait Gaussian process models builds on Bonilla and Williams (Bonilla et al., 2008), who have emphasized that the power of Kronecker covariance models for GP models (Eqn. (5.2)) is linked to non-zero observation noise. In fact, in the limit of noise-free training observations, the coupling of traits for predictions is lost in the predictive model, reducing to ordinary Gaussian process regressors for each individual phenotype. Most multi-task Gaussian process models build on a simple independent noise model, an assumption that is mainly routed in computational convenience. For example Stegle et al. (2011) show that this assumption renders the evaluation of the model likelihood and parameter gradients tractable, avoiding the explicit evaluation of the Kronecker covariance.

In animal breeding, conceptually similar models have been studied intensively in the context of multivariate linear mixed models. Algorithms for parameters estimation include Newton-Raphson (Patterson and Thompson, 1971; Thompson, 1973), derivative-free (Meyer, 1989) and expectation maximization (EM) approaches (Foulley and van Dyk, 2000). However, none of these approaches scale to the datasets of the size we are interested in: The gradient-based methods require to solve a cubic

[1]The covariance is defined as the sum of two Kronecker products and not as the classical Kronecker sum $\boldsymbol{C} \oplus \boldsymbol{R} = \boldsymbol{C} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{R}$.

operation $O(N^3T^3)$ per iteration, while the gradient-free methods can be computed more efficiently (Ducrocq and Chapuis, 1997; Meyer, 1991). However, they can only be used for a moderate number of tasks, since the search space explodes otherwise. For a more detailed overview over the different methods, we refer to Knight (2008).

In geostatistics (Zhang, 2007), linear coregionalization models have been introduced to allow for more complicated covariance structures: the signal covariance matrix is modeled as a sum of Kronecker products and the noise covariance matrix as a single Kronecker product. Parameter inference is carried out by expectation maximization and requires to compute the inverse of the complete covariance matrix in each iteration - an operation that is cubic in the number samples times the number of traits $O(N^3T^3)$. There exists also work in machine learning that extends Gaussian process multi-task models to more complex covariance structures (Álvarez and Lawrence, 2008; Wilson et al., 2012; Archembeau et al., 2011), but this leads inevitably to more complex inference schemes involving variational approximation or MCMC sampling.

In parallel to that work, Zhou and Stephens (2013) also developed an efficient inference scheme for the class of models we are studying here. However, there are two main differences between our and their approach: first, our approaches uses a gradient-based optimizer to find the best parameters, while the method of Zhou and Stephens (2013) employs a combination of an EM-variant and the Newton-Raphson algorithm (Meyer, 2006). Second, Zhou and Stephens (2013) do not put any restrictions on $\boldsymbol{C}$ and $\boldsymbol{\Sigma}$, while we restrict them to be low-rank. This restriction is crucial to avoid overfitting if the number of phenotypes is larger than a handful: the number of free parameters grows quadratically with the number of traits $T$, while the number of new data points grows only linearly (Friedman et al., 2008; Meyer and Kirkpatrick, 2005). Lately, Dahl et al. (2013) also considered an efficient EM-algorithm for the same model, but further assumed that the precision matrices $\boldsymbol{C}^{-1}$ and $\boldsymbol{\Sigma}^{-1}$ are sparse.

**Predictive distribution** In a GP-kronsum model, predictions for new data points can be carried out by using the standard Gaussian process framework, introduced in Section 2.2:

$$p(\text{vec } \boldsymbol{Y}^*|\boldsymbol{R}^*, \boldsymbol{Y}) = \mathcal{N}\left(\text{vec}\boldsymbol{Y}^* |\text{vec } \boldsymbol{M}^*, \boldsymbol{V}^*\right), \tag{5.3}$$

where $\boldsymbol{M}^*$ denotes the mean prediction and $\boldsymbol{V}^*$ is the predictive covariance. Analytical expressions for both can then be obtained analogously to the single trait

case (2.47), yielding:

$$\begin{aligned}\text{vec } \boldsymbol{M}^* &= (\boldsymbol{C} \otimes \boldsymbol{R}^*) (\boldsymbol{C} \otimes \boldsymbol{R} + \boldsymbol{\Sigma} \otimes \boldsymbol{I})^{-1} \text{vec} \boldsymbol{Y}, \\ \boldsymbol{V}^* &= (\boldsymbol{C} \otimes \boldsymbol{R}^{**}) + (\boldsymbol{\Sigma} \otimes \boldsymbol{I}) - (\boldsymbol{C} \otimes \boldsymbol{R}^*) (\boldsymbol{C} \otimes \boldsymbol{R} + \boldsymbol{\Sigma} \otimes \boldsymbol{I})^{-1} (\boldsymbol{C} \otimes \boldsymbol{R}^*),\end{aligned}$$

where $\boldsymbol{R}^*$ is the sample-sample covariance matrix between the test and training instances, and $\boldsymbol{R}^{**}$ is the sample-sample covariance matrix between the test instances.

**Task cancellation for equal task covariance matrices** A notable form of the predictive distribution (5.3) arises for the special case $\boldsymbol{C} = \boldsymbol{\Sigma}$, that is the trait-trait covariance matrix of signal and noise are identical. Similar to previous results for noise-free observations (Bonilla et al., 2008), maximizing the marginal likelihood $p(\text{vec } \boldsymbol{Y} | \boldsymbol{C}, \boldsymbol{R}, \boldsymbol{\Sigma})$ with respect to the parameters $\boldsymbol{\theta}_{\boldsymbol{R}}$ becomes independent of $\boldsymbol{C}$ and the predictions are decoupled across the phenotypes, i.e. the benefits from joint modeling are lost:

$$\begin{aligned}\text{vec } \boldsymbol{M}^* &= (\boldsymbol{C} \otimes \boldsymbol{R}^*) (\boldsymbol{C} \otimes \boldsymbol{R} + \boldsymbol{\Sigma} \otimes \boldsymbol{I})^{-1} \text{vec} \boldsymbol{Y} \\ &= (\boldsymbol{C} \otimes \boldsymbol{R}^*) (\boldsymbol{C} \otimes \boldsymbol{R} + \boldsymbol{C} \otimes \boldsymbol{I})^{-1} \text{vec} \boldsymbol{Y} \\ &= (\boldsymbol{C} \otimes \boldsymbol{R}^*) (\boldsymbol{C} \otimes (\boldsymbol{R} + \boldsymbol{I}))^{-1} \text{vec} \boldsymbol{Y} \\ &= (\boldsymbol{C} \otimes \boldsymbol{R}^*) \left(\boldsymbol{C}^{-1} \otimes (\boldsymbol{R} + \boldsymbol{I})^{-1}\right) \text{vec} \boldsymbol{Y} \\ &= \boldsymbol{C}\boldsymbol{C}^{-1} \otimes \boldsymbol{R}^* (\boldsymbol{R} + \boldsymbol{I})^{-1} \text{vec} \boldsymbol{Y} \\ &= \text{vec} \left(\boldsymbol{R}^* (\boldsymbol{R} + \boldsymbol{I})^{-1} \boldsymbol{Y}\right)\end{aligned} \tag{5.4}$$

In this case, the predictions depend only on the sample-sample covariance, but not on the trait-trait covariance. Thus, the GP-kronsum model is most useful when the covariance structure between the residuals is independent of the covariance structure of the genetic signal.

## 5.2 Efficient inference

In general, efficient inference can be carried out for Gaussian models with a sum covariance of two arbitrary Kronecker products

$$p(\text{vec} \boldsymbol{Y} | \boldsymbol{C}, \boldsymbol{R}, \boldsymbol{\Sigma}) = \mathcal{N} \left(\text{vec} \boldsymbol{Y} \, | \boldsymbol{0}, \boldsymbol{C}_{TT} \otimes \boldsymbol{R}_{NN} + \boldsymbol{\Sigma}_{TT} \otimes \boldsymbol{\Omega}_{NN}\right). \tag{5.5}$$

The key idea is to first consider a suitable data transformation that leads to a diagonalization of all covariance matrices and second to exploit Kronecker tricks whenever

possible. Let $\boldsymbol{\Sigma} = \boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma \boldsymbol{U}_\Sigma^\top$ be the eigenvalue decomposition of $\boldsymbol{\Sigma}$, and analogously for $\boldsymbol{\Omega}$. Borrowing ideas from Kalaitzis and Lawrence (2012), we can first bring the covariance matrix in a more amenable form by factoring out the structured noise:

$$\begin{aligned}
\boldsymbol{K} &= \boldsymbol{C} \otimes \boldsymbol{R} + \boldsymbol{\Sigma} \otimes \boldsymbol{\Omega} \\
&= \boldsymbol{C} \otimes \boldsymbol{R} + \left(\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{\frac{1}{2}} \otimes \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{\frac{1}{2}}\right) \left(\boldsymbol{S}_\Sigma^{\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right) \\
&= \left(\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{\frac{1}{2}} \otimes \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{\frac{1}{2}}\right) \left(\tilde{\boldsymbol{C}} \otimes \tilde{\boldsymbol{R}} + \boldsymbol{I} \otimes \boldsymbol{I}\right) \left(\boldsymbol{S}_\Sigma^{\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right), \qquad (5.6)
\end{aligned}$$

where $\tilde{\boldsymbol{C}} = \boldsymbol{S}_\Sigma^{-\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \boldsymbol{C} \boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}}$ and $\tilde{\boldsymbol{R}} = \boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_\Omega^\top \boldsymbol{R} \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{-\frac{1}{2}}$. In the following, we use $\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{C}} \otimes \tilde{\boldsymbol{R}} + \boldsymbol{I} \otimes \boldsymbol{I}$ for this transformed covariance.

**Efficient log likelihood evaluation** The log model likelihood (Eqn. (5.5)) can be expressed in terms of the transformed covariance $\tilde{\boldsymbol{K}}$:

$$\begin{aligned}
\mathcal{L} =& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{K}| - \frac{1}{2}\mathrm{vec}\boldsymbol{Y}^\top \boldsymbol{K}^{-1}\mathrm{vec}\boldsymbol{Y} \\
=& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\left(\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{\frac{1}{2}} \otimes \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{\frac{1}{2}}\right) \tilde{\boldsymbol{K}} \left(\boldsymbol{S}_\Sigma^{\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right)\right| \\
& - \frac{1}{2}\mathrm{vec}\boldsymbol{Y}^\top \left[\left(\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{\frac{1}{2}} \otimes \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{\frac{1}{2}}\right) \tilde{\boldsymbol{K}} \left(\boldsymbol{S}_\Sigma^{\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right)\right]^{-1} \mathrm{vec}\boldsymbol{Y} \\
=& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{\frac{1}{2}} \otimes \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{\frac{1}{2}}\right| - \frac{1}{2}\ln\left|\tilde{\boldsymbol{K}}\right| - \frac{1}{2}\ln\left|\boldsymbol{S}_\Sigma^{\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right| \\
& - \frac{1}{2}\left(\left(\boldsymbol{S}_\Sigma^{-\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right)\mathrm{vec}\boldsymbol{Y}\right)^\top \tilde{\boldsymbol{K}}^{-1} \left(\left(\boldsymbol{S}_\Sigma^{-\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right)\mathrm{vec}\boldsymbol{Y}\right)^\top \\
=& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln|\tilde{\boldsymbol{K}}| - \frac{1}{2}\ln|\boldsymbol{S}_\Sigma \otimes \boldsymbol{S}_\Omega| - \frac{1}{2}\mathrm{vec}\tilde{\boldsymbol{Y}}^\top \tilde{\boldsymbol{K}}^{-1}\mathrm{vec}\tilde{\boldsymbol{Y}} \\
=& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln|\tilde{\boldsymbol{K}}| - \frac{N}{2}\ln|\boldsymbol{S}_\Sigma| - \frac{T}{2}\ln|\boldsymbol{S}_\Omega| - \frac{1}{2}\mathrm{vec}\tilde{\boldsymbol{Y}}^\top \tilde{\boldsymbol{K}}^{-1}\mathrm{vec}\tilde{\boldsymbol{Y}}, \qquad (5.7)
\end{aligned}$$

where $\mathrm{vec}\tilde{\boldsymbol{Y}} = \left(\boldsymbol{S}_\Sigma^{-\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right)\mathrm{vec}\boldsymbol{Y} = \mathrm{vec}\left(\boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_\Omega^T \boldsymbol{Y} \boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}}\right)$ are the projected phenotypes. Except for the additional term $-\frac{1}{2}\ln|\boldsymbol{S}_\Sigma \otimes \boldsymbol{S}_\Omega|$, resulting from the transformation, the log likelihood has exactly the same form as for multi-task GP regression with iid noise (Bonilla et al., 2008; Stegle et al., 2011). Using an analogous

derivation, we can now efficiently evaluate the log likelihood:

$$\begin{aligned}
\mathcal{L} =& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln|\tilde{\boldsymbol{C}}\otimes\tilde{\boldsymbol{R}} + \boldsymbol{I}\otimes\boldsymbol{I}| - \frac{N}{2}\ln|\boldsymbol{S}_\Sigma| - \frac{T}{2}\ln|\boldsymbol{S}_\Omega| \\
& - \frac{1}{2}\mathrm{vec}\tilde{\boldsymbol{Y}}^\top\left(\tilde{\boldsymbol{C}}\otimes\tilde{\boldsymbol{R}} + \boldsymbol{I}\otimes\boldsymbol{I}\right)^{-1}\mathrm{vec}\tilde{\boldsymbol{Y}} \\
=& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln\left|(\boldsymbol{U}_{\tilde{C}}\otimes\boldsymbol{U}_{\tilde{R}})\left(\boldsymbol{S}_{\tilde{C}}\otimes\boldsymbol{S}_{\tilde{R}} + \boldsymbol{I}\otimes\boldsymbol{I}\right)(\boldsymbol{U}_{\tilde{C}}\otimes\boldsymbol{U}_{\tilde{R}})^\top\right| - \frac{N}{2}\ln|\boldsymbol{S}_\Sigma| \\
& - \frac{T}{2}\ln|\boldsymbol{S}_\Omega| - \frac{1}{2}\mathrm{vec}\tilde{\boldsymbol{Y}}^\top\left[(\boldsymbol{U}_{\tilde{C}}\otimes\boldsymbol{U}_{\tilde{R}})\left(\boldsymbol{S}_{\tilde{C}}\otimes\boldsymbol{S}_{\tilde{R}} + \boldsymbol{I}\otimes\boldsymbol{I}\right)(\boldsymbol{U}_{\tilde{C}}\otimes\boldsymbol{U}_{\tilde{R}})^\top\right]^{-1}\mathrm{vec}\tilde{\boldsymbol{Y}} \\
=& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{S}_{\tilde{\boldsymbol{C}}}\otimes\boldsymbol{S}_{\tilde{\boldsymbol{R}}} + \boldsymbol{I}\otimes\boldsymbol{I}| - \frac{N}{2}\ln|\boldsymbol{S}_\Sigma| - \frac{T}{2}\ln|\boldsymbol{S}_\Omega| \\
& - \frac{1}{2}\left[(\boldsymbol{U}_{\tilde{C}}\otimes\boldsymbol{U}_{\tilde{R}})^\top\mathrm{vec}\tilde{\boldsymbol{Y}}\right]^\top\left(\boldsymbol{S}_{\tilde{C}}\otimes\boldsymbol{S}_{\tilde{R}} + \boldsymbol{I}\otimes\boldsymbol{I}\right)^{-1}\left[(\boldsymbol{U}_{\tilde{C}}\otimes\boldsymbol{U}_{\tilde{R}})^\top\mathrm{vec}\tilde{\boldsymbol{Y}}\right] \\
=& -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{S}_{\tilde{\boldsymbol{C}}}\otimes\boldsymbol{S}_{\tilde{\boldsymbol{R}}} + \boldsymbol{I}\otimes\boldsymbol{I}| - \frac{N}{2}\ln|\boldsymbol{S}_\Sigma| - \frac{T}{2}\ln|\boldsymbol{S}_\Omega| \\
& - \frac{1}{2}\mathrm{vec}\left(\boldsymbol{U}_{\tilde{\boldsymbol{R}}}^\top\tilde{\boldsymbol{Y}}\boldsymbol{U}_{\tilde{\boldsymbol{C}}}\right)^\top\left[\mathrm{diag}\left(\boldsymbol{S}_{\tilde{\boldsymbol{C}}}\otimes\boldsymbol{S}_{\tilde{\boldsymbol{R}}} + \boldsymbol{I}\otimes\boldsymbol{I}\right)^{-1}\odot\mathrm{vec}\left(\boldsymbol{U}_{\tilde{\boldsymbol{R}}}^\top\tilde{\boldsymbol{Y}}\boldsymbol{U}_{\tilde{\boldsymbol{C}}}\right)\right], \qquad (5.8)
\end{aligned}$$

where we have defined the eigenvalue decomposition of $\tilde{\boldsymbol{C}}$ as $\boldsymbol{U}_{\tilde{C}}\boldsymbol{S}_{\tilde{C}}\boldsymbol{U}_{\tilde{C}}^\top$ and similar for $\tilde{\boldsymbol{R}}$.

**Efficient gradient evaluation** The derivative of the log marginal likelihood with respect to a single covariance parameter $\theta_R \in \boldsymbol{\theta}_R$ is given by:

$$\frac{\partial}{\partial\theta_R}\mathcal{L} = -\frac{1}{2}\frac{\partial}{\partial\theta_R}\ln|\tilde{\boldsymbol{K}}| - \frac{1}{2}\mathrm{vec}\tilde{\boldsymbol{Y}}^\top\left(\frac{\partial}{\partial\theta_R}\tilde{\boldsymbol{K}}^{-1}\right)\mathrm{vec}(\tilde{\boldsymbol{Y}}) \qquad (5.9)$$

The derivative of the log determinant term can then be efficiently evaluated as follows:

$$\begin{aligned}
\frac{\partial}{\partial\theta_R}\ln|\tilde{\boldsymbol{K}}| =& \mathrm{Tr}\left(\tilde{\mathbf{K}}^{-1}\left(\tilde{\boldsymbol{C}}\otimes\frac{\partial}{\partial\theta_R}\tilde{\boldsymbol{R}}\right)\right) \\
=& \mathrm{Tr}\left((\mathbf{U}_{\tilde{\mathbf{C}}}\otimes\mathbf{U}_{\tilde{\mathbf{R}}})\left(\mathbf{S}_{\tilde{\mathbf{C}}}\otimes\mathbf{S}_{\tilde{\mathbf{R}}} + \mathbf{I}\otimes\mathbf{I}\right)(\mathbf{U}_{\tilde{\mathbf{C}}}\otimes\mathbf{U}_{\tilde{\mathbf{R}}})^\top\left(\tilde{\boldsymbol{C}}\otimes\frac{\partial}{\partial\theta_R}\tilde{\boldsymbol{R}}\right)\right) \\
=& \mathrm{Tr}\left(\left(\mathbf{S}_{\tilde{\mathbf{C}}}\otimes\mathbf{S}_{\tilde{\mathbf{R}}} + \mathbf{I}\otimes\mathbf{I}\right)(\mathbf{U}_{\tilde{\mathbf{C}}}\otimes\mathbf{U}_{\tilde{\mathbf{R}}})^\top\left(\tilde{\boldsymbol{C}}\otimes\frac{\partial}{\partial\theta_R}\tilde{\boldsymbol{R}}\right)(\mathbf{U}_{\tilde{\mathbf{C}}}\otimes\mathbf{U}_{\tilde{\mathbf{R}}})\right) \\
=& \mathrm{Tr}\left(\left(\mathbf{S}_{\tilde{\mathbf{C}}}\otimes\mathbf{S}_{\tilde{\mathbf{R}}} + \mathbf{I}\otimes\mathbf{I}\right)\left(\mathbf{U}_{\tilde{\mathbf{C}}}^\top\tilde{\mathbf{C}}\mathbf{U}_{\tilde{\mathbf{C}}}\otimes\mathbf{U}_{\tilde{\mathbf{R}}}^\top\left(\frac{\partial}{\partial\theta_{\mathbf{R}}}\tilde{\mathbf{R}}\right)\mathbf{U}_{\tilde{\mathbf{R}}}\right)\right) \\
=& \mathrm{Tr}\left(\left(\mathbf{S}_{\tilde{\mathbf{C}}}\otimes\mathbf{S}_{\tilde{\mathbf{R}}} + \mathbf{I}\otimes\mathbf{I}\right)\left(\mathbf{S}_{\tilde{\mathbf{C}}}\otimes\mathbf{U}_{\tilde{\mathbf{R}}}^\top\left(\frac{\partial}{\partial\theta_{\mathbf{R}}}\tilde{\mathbf{R}}\right)\mathbf{U}_{\tilde{\mathbf{R}}}\right)\right) \qquad (5.10)
\end{aligned}$$

Since $\mathbf{S}_{\tilde{\mathbf{C}}} \otimes \mathbf{S}_{\tilde{\mathbf{R}}} + \mathbf{I} \otimes \mathbf{I}$ is a diagonal matrix, we thereby only have to evaluate the diaogonal entries of $\mathbf{S}_{\tilde{\mathbf{C}}} \otimes \mathbf{U}_{\tilde{\mathbf{R}}}^{\top} \left( \frac{\partial}{\partial \theta_{\mathbf{R}}} \tilde{\mathbf{R}} \right) \mathbf{U}_{\tilde{\mathbf{R}}}$ instead of computing the whole Kronecker product. Next, we turn to the derivative of the squared form

$$
\begin{aligned}
\frac{\partial}{\partial \theta_R} \mathrm{vec}\tilde{\boldsymbol{Y}}^{\top} \tilde{\boldsymbol{K}}^{-1} \mathrm{vec}\tilde{\boldsymbol{Y}} &= -\mathrm{vec}\tilde{\boldsymbol{Y}}^{\top} \left[ \tilde{\boldsymbol{K}}^{-1} \left( \frac{\partial}{\partial \theta_R} \tilde{\boldsymbol{K}}^{-1} \right) \tilde{\boldsymbol{K}}^{-1} \right] \mathrm{vec}\tilde{\boldsymbol{Y}} \\
&= -\left( \tilde{\boldsymbol{K}}^{-1} \mathrm{vec}\tilde{\boldsymbol{Y}} \right)^{\top} \left( \tilde{\boldsymbol{C}} \otimes \frac{\partial}{\partial \theta_R} \tilde{\boldsymbol{R}} \right) \left( \tilde{\boldsymbol{K}}^{-1} \mathrm{vec}\tilde{\boldsymbol{Y}} \right) \\
&= -\left( (\boldsymbol{U}_{\tilde{C}} \otimes \boldsymbol{U}_{\tilde{R}}) \, \mathrm{vec}\hat{\boldsymbol{Y}} \right)^{\top} \left( \tilde{\boldsymbol{C}} \otimes \frac{\partial}{\partial \theta_R} \tilde{\boldsymbol{R}} \right) \left( (\boldsymbol{U}_{\tilde{C}} \otimes \boldsymbol{U}_{\tilde{R}}) \, \mathrm{vec}\hat{\boldsymbol{Y}} \right), \\
&= -\mathrm{vec}\hat{\boldsymbol{Y}}^{\top} \left( \boldsymbol{U}_{\tilde{C}}^{\top} \tilde{\boldsymbol{C}} \boldsymbol{U}_{\tilde{C}} \otimes \boldsymbol{U}_{\tilde{R}}^{\top} \frac{\partial}{\partial \theta_R} \tilde{\boldsymbol{R}} \boldsymbol{U}_{\tilde{R}} \right) \mathrm{vec}\hat{\boldsymbol{Y}}, \\
&= -\mathrm{vec}\hat{\boldsymbol{Y}}^{\top} \left( \boldsymbol{S}_{\tilde{C}} \otimes \boldsymbol{U}_{\tilde{R}}^{\top} \frac{\partial}{\partial \theta_R} \tilde{\boldsymbol{R}} \boldsymbol{U}_{\tilde{R}} \right) \mathrm{vec}\hat{\boldsymbol{Y}}, \\
&= -\mathrm{vec}(\hat{\boldsymbol{Y}})^{\top} \mathrm{vec} \left( \boldsymbol{U}_{\tilde{\boldsymbol{R}}}^{\top} \left( \frac{\partial}{\partial \theta_R} \tilde{\boldsymbol{R}} \right) \boldsymbol{U}_{\tilde{\boldsymbol{R}}} \hat{\boldsymbol{Y}} \boldsymbol{S}_{\tilde{\boldsymbol{C}}} \right) \qquad (5.11)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathrm{vec}(\hat{\boldsymbol{Y}}) &= \left( \boldsymbol{S}_{\tilde{\boldsymbol{C}}} \otimes \boldsymbol{S}_{\tilde{\boldsymbol{R}}} + \boldsymbol{I} \otimes \boldsymbol{I} \right)^{-1} \left( \boldsymbol{U}_{\tilde{\boldsymbol{C}}} \otimes \boldsymbol{U}_{\tilde{\boldsymbol{R}}} \right)^{\top} \mathrm{vec}\tilde{\boldsymbol{Y}} \\
&= \left( \boldsymbol{S}_{\tilde{\boldsymbol{C}}} \otimes \boldsymbol{S}_{\tilde{\boldsymbol{R}}} + \boldsymbol{I} \otimes \boldsymbol{I} \right)^{-1} \mathrm{vec} \left( \boldsymbol{U}_{\tilde{\boldsymbol{R}}}^{\top} \tilde{\boldsymbol{Y}} \boldsymbol{U}_{\tilde{\boldsymbol{C}}} \right).
\end{aligned}
$$

Finally, plugging the derivative of the log determinant and the squared form together again, we obtain

$$
\begin{aligned}
\frac{\partial}{\partial \theta_R} \mathcal{L} = &-\frac{1}{2} \mathrm{Tr} \left( \left( \mathbf{S}_{\tilde{\mathbf{C}}} \otimes \mathbf{S}_{\tilde{\mathbf{R}}} + \mathbf{I} \otimes \mathbf{I} \right) \left( \mathbf{S}_{\tilde{\mathbf{C}}} \otimes \mathbf{U}_{\tilde{\mathbf{R}}}^{\top} \left( \frac{\partial}{\partial \theta_{\mathbf{R}}} \tilde{\mathbf{R}} \right) \mathbf{U}_{\tilde{\mathbf{R}}} \right) \right) \\
&+ \frac{1}{2} \mathrm{vec}(\hat{\boldsymbol{Y}})^{\top} \mathrm{vec} \left( \boldsymbol{U}_{\tilde{\boldsymbol{R}}}^{\top} \left( \frac{\partial}{\partial \theta_R} \tilde{\boldsymbol{R}} \right) \boldsymbol{U}_{\tilde{\boldsymbol{R}}} \hat{\boldsymbol{Y}} \boldsymbol{S}_{\tilde{\boldsymbol{C}}} \right), \qquad (5.12)
\end{aligned}
$$

Analogous gradients can be derived for all other covariance parameters $\theta \in \{\boldsymbol{\theta}_C, \boldsymbol{\theta}_{\Sigma}, \boldsymbol{\theta}_{\Omega}\}$. The proposed speed-ups also apply to the special cases where $\boldsymbol{\Sigma}$ is modeled as being diagonal as in (Bonilla et al., 2008), or for optimizing the parameters of a kernel function. Since the sum of Kronecker products generally can not be written as a single Kronecker product, the speed-ups cannot be generalized to larger sums of Kronecker products.

**Efficient prediction** Similarly, the mean predictor (Eqn. (5.3)) can be efficiently evaluated

$$\begin{aligned}
\text{vec } \boldsymbol{M}^* &= (\boldsymbol{C} \otimes \boldsymbol{R}^*) \left(\boldsymbol{C} \otimes \boldsymbol{R} + \boldsymbol{\Sigma} \otimes \boldsymbol{I}\right)^{-1} \text{vec}\boldsymbol{Y} \\
&= (\boldsymbol{C} \otimes \boldsymbol{R}^*) \left[\left(\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{\frac{1}{2}} \otimes \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{\frac{1}{2}}\right) \tilde{\boldsymbol{K}} \left(\boldsymbol{S}_\Sigma^{\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right)\right]^{-1} \text{vec}\boldsymbol{Y} \\
&= (\boldsymbol{C} \otimes \boldsymbol{R}^*) \left(\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}} \otimes \boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{-\frac{1}{2}}\right) \tilde{\boldsymbol{K}}^{-1} \left(\boldsymbol{S}_\Sigma^{-\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \otimes \boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right) \text{vec}\boldsymbol{Y} \\
&= \left(\boldsymbol{C}\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}} \otimes \boldsymbol{R}^*\boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{-\frac{1}{2}}\right) \tilde{\boldsymbol{K}}^{-1} \text{vec}\left(\left(\boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_\Omega^\top\right) \boldsymbol{Y} \left(\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}}\right)\right) \\
&= \left(\boldsymbol{C}\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}} \otimes \boldsymbol{R}^*\boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{-\frac{1}{2}}\right) \left(\tilde{\boldsymbol{K}}^{-1} \text{vec}\tilde{\boldsymbol{Y}}\right) \\
&= \left(\boldsymbol{C}\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}} \otimes \boldsymbol{R}^*\boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{-\frac{1}{2}}\right) \text{vec}\left(\boldsymbol{U}_{\tilde{R}} \hat{\boldsymbol{Y}} \boldsymbol{U}_{\tilde{C}}^\top\right) \\
&= \text{vec}\left(\left(\boldsymbol{R}^*\boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{-\frac{1}{2}}\right) \left(\boldsymbol{U}_{\tilde{\boldsymbol{R}}} \hat{\boldsymbol{Y}} \boldsymbol{U}_{\tilde{\boldsymbol{C}}}^\top\right) \left(\boldsymbol{C}\boldsymbol{U}_\Sigma \boldsymbol{S}_\Sigma^{-\frac{1}{2}}\right)^\top\right) \\
&= \text{vec}\left(\boldsymbol{R}^*\boldsymbol{U}_\Omega \boldsymbol{S}_\Omega^{-\frac{1}{2}} \boldsymbol{U}_{\tilde{R}} \hat{\boldsymbol{Y}} \boldsymbol{U}_{\tilde{C}}^\top \boldsymbol{S}_\Sigma^{-\frac{1}{2}} \boldsymbol{U}_\Sigma^\top \boldsymbol{C}^\top\right). \qquad (5.13)
\end{aligned}$$

**Gradient-based parameter inference** The closed-form expression of the marginal likelihood (Eqn. (5.8)) and gradients with respect to covariance parameters (Eqn. (5.12)) allow for use of gradient-based parameter inference. In the experiments, we employ a variant of L-BFGS-B (Zhu et al., 1997).

**Computational cost.** While the naive approach has a runtime of $O(N^3 \cdot T^3)$ and memory requirement of $O(N^2 \cdot T^2)$, as it explicitly computes and inverts the Kronecker products, our inference scheme reduces the runtime to $O(N^3 + T^3)$ and the memory requirement to $O(N^2 + T^2)$, making it applicable to large numbers of samples and phenotypes. The empirical runtime savings over the naive approach are explored in the next section.

## 5.3 Experiments

We investigated the performance of the proposed GP-kronsum model on simulated data, as well as on phenotype prediction. To investigate the benefits of structured residual covariances, we compared the GP-kronsum model to a Gaussian process (GP-kronprod) with iid noise (Stegle et al., 2011)

$$p(\text{vec}\boldsymbol{Y} | \boldsymbol{C}, \boldsymbol{R}, \sigma_e^2) = \mathcal{N}\left(\text{vec}\boldsymbol{Y} \,\middle|\, \boldsymbol{0}, \boldsymbol{C} \otimes \boldsymbol{R} + \sigma_e^2 \boldsymbol{I}\right), \qquad (5.14)$$

as well as independent modeling of traits using a standard Gaussian process (GP-single)

$$p(\mathrm{vec}\boldsymbol{Y}|\boldsymbol{R}, \boldsymbol{\sigma}_g^2, \boldsymbol{\sigma}_e^2) = \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{y}_t \left| \mathbf{0}, {\sigma_{gt}}^2 \boldsymbol{R} + {\sigma_{et}}^2 \boldsymbol{I}\right.\right) \tag{5.15}$$

and joint modeling of all phenotypes using a standard Gaussian on a pooled dataset, naively merging data from all tasks (GP-pool).

$$p(\mathrm{vec}\boldsymbol{Y}|\boldsymbol{R}, \sigma_g^2, \sigma_e^2) = \mathcal{N}\left(\mathrm{vec}\boldsymbol{Y} \left| \mathbf{0}, \sigma_g^2 \mathbf{1} \otimes \boldsymbol{R} + \sigma_e^2 \boldsymbol{I}\right.\right), \tag{5.16}$$

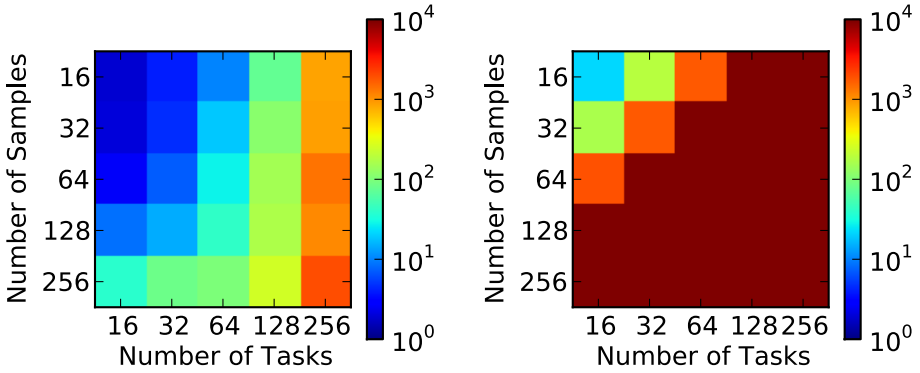
The predictive performance of individual models was assessed through 10-fold cross-validation. For each fold, model parameters were fit on the training data only. To avoid local optima during training, parameter fitting was carried out using five random restarts of the parameters on 90% of the training instances.

The remaining 10% of the training instances were used for out of sample selection using the maximum log likelihood as criterion. Unless stated otherwise, in the multi-task models the relationship between tasks was parameterized as $\boldsymbol{z}\boldsymbol{z}^\top + \sigma^2 \boldsymbol{I}$, the sum of a rank-1 matrix and a constant diagonal component. The rank-1 matrix allows for effects that are shared over all phenotypes, while the identity matrix allows for trait-specific effects. Both parameters, $\boldsymbol{z}$ and $\sigma^2$, were learnt by optimizing the marginal likelihood. Finally, we measured the predictive performance of the different methods via the averaged square of Pearson's correlation coefficient $r^2$ between the true and the predicted output, averaged over all phenotypes (Ober et al., 2012).

### 5.3.1 Simulations

First, we considered simulated experiments to explore the runtime behavior and to find out if there are settings in which GP-kronsum performs better than existing methods.

**Runtime evaluation** As a first experiment, we examined the runtime behavior of our method as a function of the number of samples and of the number of phenotypes. Both parameters were varied in the range $\{16, 32, 64, 128, 256\}$. The simulated dataset was drawn from the GP-kronsum model (Eqn. (5.2)) using a linear kernel for the sample-sample covariance matrix $\boldsymbol{R}$ and rank-1 matrices for the trait-trait covariances $\boldsymbol{C}$ and $\boldsymbol{\Sigma}$. The runtime of this model was assessed for a single likelihood optimization on an AMD Opteron Processor 6,378 using a single core (2.4GHz,



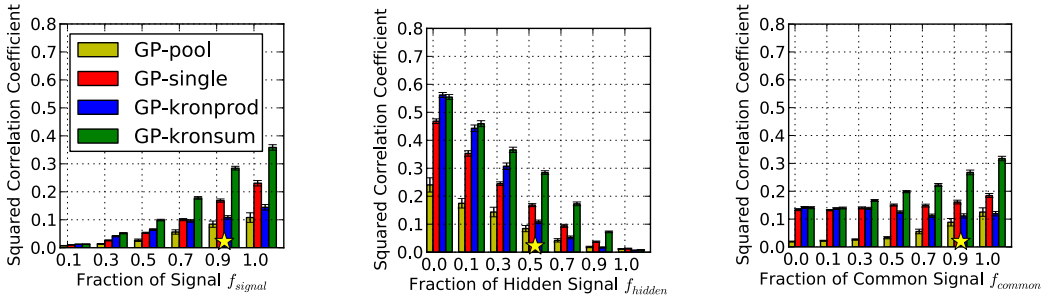

(a) Efficient Implementation (b) Naive Implementation

Figure 5.1: **Runtime comparison on synthetic data.** We compare our efficient GP-KS implementation (left) versus its naive counterpart (right). Shown is the runtime in seconds on a logarithmic scale as a function of the sample size and the number of traits. The optimization was stopped prematurely if it did not complete after $10^4$ seconds.

2,048 KB Cache, 512 GB Memory) and compared to a naive implementation. The optimization was stopped prematurely if it did not converge within $10^4$ seconds.

In the experiments, we used a standard linear kernel on the SNPs as sample-sample covariance while learning the trait-trait covariances. This modeling choice results in a steeper runtime increase with the number of traits, due to the increasing number of model parameters to be estimated. Figure 5.1 demonstrates the significant speed-up. While our algorithm can handle 256 samples/256 tasks with ease, the naive implementation failed to process more than 32 samples/32 tasks.

**Hidden confounding induces correlated residuals** A common source of structured residuals are hidden confounders such as environmental (Hunter, 2005) demographic (Rodwell et al., 2004) and technical factors (Kerr and Churchill, 2001). While this is well acknowledged when studying gene expression data (Leek and Storey, 2007; Listgarten et al., 2010; Fusi et al., 2012), we here explore the consequences of hidden confounding for multi-trait prediction. For this, we simulated the phenotypes as




(a) Total Signal (b) Hidden Signal (c) Shared Signal

Figure 5.2: **Evaluation of alternative methods for different simulation settings. (a)** Evaluation for varying signal strength. **(b)** Evaluation for variable impact of the hidden signal. **(c)** Evaluation for different strength of relatedness between the tasks. In each simulation setting, all other parameters were kept constant at default parameters marked with the yellow star symbol.

follows:

$$\boldsymbol{y}_t = \underbrace{\boldsymbol{X}\boldsymbol{w}_t}_{\text{genetic signal}} + \underbrace{\boldsymbol{Z}\boldsymbol{u}_t}_{\text{hidden confounder}} + \underbrace{\boldsymbol{\epsilon}_t}_{\text{noise}}, \tag{5.17}$$

where $\boldsymbol{Z} \in \mathbb{R}^{N \times P}$ are the unobserved confounders with $P$ being the number of unobserved confounders and $\boldsymbol{u}_t \in \mathbb{R}^P$ being the weights. The prior on the weights are specified as follows:

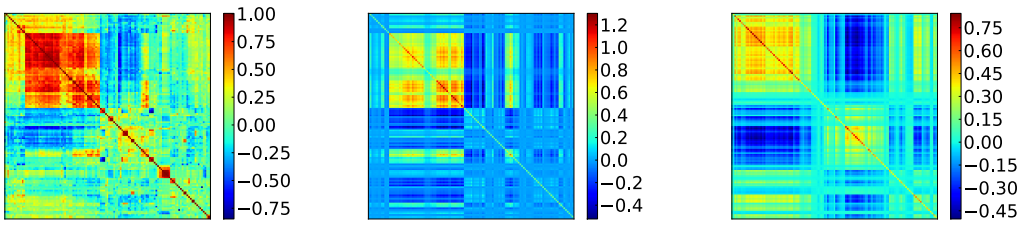
$$p(\boldsymbol{W}^\top) = \prod_{m=1}^{M} \mathcal{N}\left(\boldsymbol{w}_m \,|\, \boldsymbol{0}, \mu_{\text{signal}} \cdot (1 - \mu_{\text{hidden}}) \cdot \left[\mu_{\text{common}} \boldsymbol{r}\boldsymbol{r}^\top + (1 - \mu_{\text{common}})\boldsymbol{I}\right]\right)$$
$$p(\boldsymbol{U}^\top) = \prod_{p=1}^{P} \mathcal{N}\left(\boldsymbol{u}_p \,|\, \boldsymbol{0}, \mu_{\text{signal}} \cdot \mu_{\text{hidden}} \cdot \left[\mu_{\text{common}} \boldsymbol{s}\boldsymbol{s}^\top + (1 - \mu_{\text{common}})\boldsymbol{I}\right]\right),$$

where $\boldsymbol{r}$ and $\boldsymbol{s}$ are drawn from the standard normal distribution and represent the shared phenotypic effects. The trade-off parameter $\mu_{\text{common}}$ determines the extent of relatedness between the phenotypes, the parameter $\mu_{\text{hidden}}$ controls the ratio between the genetic and the confounding effect, and the parameter $\mu_{\text{signal}}$ defines the ratio between noise and signal. Finally, the noise is drawn from an isotropic Gaussian distribution $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, (1 - \mu_{\text{signal}})\boldsymbol{I})$.

To investigate the impact of the different trade-off parameters, we considered a series of datasets varying one of the parameters while keeping others fixed. We varied $\mu_{\text{signal}}$ in the range $\{0.1, 0.3, 0.5, 0.7, \mathbf{0.9}, 1.0\}$, $\mu_{\text{common}} \in \{0.0, 0.1, 0.3, 0.5, 0.7, \mathbf{0.9}, 1.0\}$ and $\mu_{\text{hidden}} \in \{0.0, 0.1, 0.3, \mathbf{0.5}, 0.7, 0.9, 1.0\}$, with default values marked in bold. Note that the best possible explained variance for the default setting is 45%, as the signal is split up equally between the genetic and the confounding process. For all simulation experiments, we created datasets with 200 samples and 10 phenotypes. The number of SNPs was set to 200, as well as the number of hidden confounders. For each such simulation setting, we created 30 datasets.

First, we considered the impact of variation in signal strength $\mu_{\text{signal}}$ (Figure 5.2a), where the overall signal was divided up equally between the two processes. Both GP-single and GP-kronsum performed better as the overall signal strength increased. The performance of GP-kronsum was superior, as the model can exploit the relatedness between the different phenotypes. Second, we explored the ability of the different methods to cope with hidden confounding (Figure 5.2b). In the absence of confounding factors ($\mu_{\text{hidden}} = 0$), GP-kronprod and GP-kronsum had very similar performances, as both methods leverage the shared genetic signal, thereby outperforming the single-task Gaussian processes. However, as the magnitude of the confounder increases, GP-kronprod falsely explains the task correlation completely by the genetic covariance term which leads to loss of predictive power. Last, we examined the ability of different methods to exploit the relatedness between the traits (Figure 5.2c). Since GP-single assumed independent phenotypes, the model performed very similarly across the full range of experiments. GP-kronprod suffered from the same limitations as previously described, since the correlation between the confounding effects increases synchronously with the correlation between the genetic effects as $\mu_{\text{common}}$ increases. In contrast, GP-kronsum could take advantage of the shared component between the phenotypes, as knowledge is transferred between them. GP-pool was consistently outperformed by all competitors as two of its main assumptions are heavily violated: samples of different phenotypes do not share the same signal and the residuals are neither independent of each other, nor do they have the same noise level.

In summary, the proposed model is robust to a range of different settings and clearly outperforms its competitors when the tasks are related to each other and hidden confounders are present.




(a) Empirical (b) Signal (c) Noise

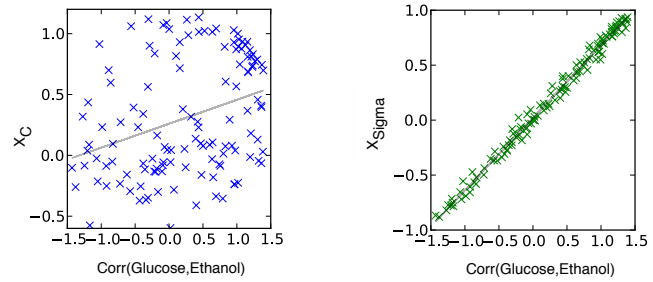
Figure 5.3: **Fitted task covariance matrices for gene expression levels in yeast. (a)** Empirical covariance matrix of the gene expression levels. **(b)** Signal covariance matrix learnt by GP-kronsum. **(c)** Noise covariance matrix learnt by GP-kronsum. The ordering of the tasks was determined using hierarchical clustering on the empirical covariance matrix.

### 5.3.2 Applications to phenotype prediction

We next applied our method to developmental phenotypes in *Arabidopsis thaliana* and gene expression levels in yeast demonstrating that hidden confounders play an important role in phenotype prediction and hence warrant greater attention.

**Gene expression prediction in yeast** We considered gene expression levels from a yeast genetics study (Smith and Kruglyak, 2008). The dataset comprised of gene expression levels of $5,493$ genes and $2,956$ SNPs, measured for 109 yeast crosses. Expression levels for each cross were measured in two conditions (glucose and ethanol as carbon source), yielding a total of 218 samples. In this experiment, we treated the condition information as a hidden factor instead of regressing it out, which is analogous to the confounding factors in the simulation experiments. The goal of this experiment was to investigate how alternative methods can deal and correct for this hidden covariate. We normalized the markers and all phenotypes to zero mean and unit variance. Subsequently, we filtered out all genes that were not consistently expressed in at least 90% of the samples ($z$-score cutoff 1.5). We also discarded genes having a heritability $h^2$ of less than 0.1 or were complete heritable $h^2 > 0.9$, reducing the number of genes to 123, which we considered as tasks in our experiment. The heritability was estimated by a univariate Gaussian process model. We used a linear kernel calculated on the markers for the sample-sample covariance.

Figure 5.3 shows the empirical covariance and the learnt task covariances by GP-kronsum. Both learnt covariances are highly structured, demonstrating that the




(a) Signal (b) Noise

Figure 5.4: **Correlation between the mean difference of the two conditions and the latent factors on the yeast dataset.** Shown is the strength of the latent factor of **(a)** the genetic signal and **(b)** the noise trait-trait covariance matrix as a function of the mean difference between the two environmental conditions. Each dot corresponds to one gene expression level.

assumption of iid noise in the GP-kronprod model is violated in this dataset. While the signal task covariance matrix reflects genetic signals that are shared between the gene expression levels, the noise covariance matrix mainly captures the mean shift between the two conditions the gene expression levels were measured in (Figure 5.4). To investigate the robustness of the reconstructed latent factor, we repeated the training 10 times. The mean latent factors and its standard errors were $0.2103 \pm 0.0088$ (averaged over factors, over the 10 best runs selected by out-of-sample likelihood), demonstrating robustness of the inference.

When considering alternative methods for out of sample prediction, the proposed Kronecker Sum model ($r^2$(GP-kronsum)=$0.3322 \pm 0.0014$) performed significantly better than previous approaches ($r^2$(GP-pool)=$0.0673 \pm 0.0004$, $r^2$(GP-single)=$0.2594 \pm 0.0011$, $r^2$(GP-kronprod)=$0.1820 \pm 0.0020$). The results are averaged over 10 runs and $\pm$ denotes the corresponding standard errors.

**Multi-phenotype prediction in *Arabidopsis thaliana*** As a second dataset, we considered a genome-wide association study in *Arabidopsis thaliana* (Atwell et al., 2010) to assess the prediction of developmental phenotypes from genomic data. This dataset consisted of 147 samples and 216,130 single nucleotide polymorphisms (SNPs, here used as features). We considered the phenotypes *flowering period duration*, *life cycle period*, *maturation period* and *reproduction period*. To avoid outliers and issues

| | Flowering period duration | Life cycle period | Maturation period | Reproduction period |
|---|---|---|---|---|
| GP-pool | $0.0502 \pm 0.0025$ | $0.1038 \pm 0.0034$ | $0.0460 \pm 0.0024$ | $\mathbf{0.0478 \pm 0.0013}$ |
| GP-single | $0.0385 \pm 0.0017$ | $\mathbf{0.3500 \pm 0.0069}$ | $0.1612 \pm 0.0027$ | $0.0272 \pm 0.0024$ |
| GP-kronprod | $0.0846 \pm 0.0021$ | $\mathbf{0.3417 \pm 0.0062}$ | $\mathbf{0.1878 \pm 0.0042}$ | $\mathbf{0.0492 \pm 0.0032}$ |
| GP-kronsum | $\mathbf{0.1127 \pm 0.0049}$ | $\mathbf{0.3485 \pm 0.0068}$ | $\mathbf{0.1918 \pm 0.0041}$ | $\mathbf{0.0501 \pm 0.0033}$ |

Table 5.1: **Predictive performance of the different methods on the** *Arabidopsis thaliana* **dataset.** Shown is the squared correlation coefficient and its standard error (measured by repeating 10-fold cross-validation 10 times).

due to non-Gaussianity, we preprocessed the phenotypic data by first converting it to ranks and squashing the ranks through the inverse cumulative Gaussian distribution. The SNPs in *Arabidopsis thaliana* are homozygous and we discarded all markers with a minor allele frequency of less than 10%, resulting in 176,436 SNPs. Subsequently, we normalized the markers to zero mean and unit variance. Again, we used a linear kernel on the SNPs as sample covariance.

Since the causal processes in *Arabidopsis thaliana* are complex, we allowed the rank of the signal and noise matrix to vary between 1 and 3. The appropriate rank complexity was selected on the 10% hold out data of the training fold. We considered the average squared correlation coefficient on the holdout fraction of the training data to select the model for prediction on the test dataset. Notably, for GP-kronprod, the selected task complexity was $\text{rank}(\boldsymbol{C}) = 3$, whereas GP-kronsum selected a simpler structure for the signal task covariance ($\text{rank}(\boldsymbol{C}) = 1$) and chose a more complex noise covariance, $\text{rank}(\boldsymbol{\Sigma}) = 2$.

The cross validation prediction performance of each model is shown in Table 5.1. For *reproduction period*, GP-single is outperformed by all other methods. For the phenotype *life cycle period*, the noise estimates of the univariate GP model were close to zero, and hence all methods, except of GP-pool, performed equally well since the measurements of the other phenotypes do not provide additional information. For *maturation period*, GP-kronsum and GP-kronprod showed improved performance compared to GP-single and GP-pool. For *flowering period duration*, GP-kronsum outperformed its competitors.

## 5.4 Summary

Here, we presented an efficient inference scheme for learning multi-trait Gaussian process models. In our experiments, we concentrated on phenotype prediction, but

the method, and the proposed speed-ups, can equally be used for cross-heritability estimation (Lee et al., 2012) and as background model for association testing in multivariate linear mixed models (Segura et al., 2012; Zhou and Stephens, 2013).

In our applications, we demonstrated the advantages of our proposed model over a broad range of different settings. However and as we noted before, the benefits of multi-trait modeling are lost if a) the phenotypes are fully determined by the genetic covariance matrix b) the signal and noise trait-trait covariance matrices are identical to each other or c) the phenotypes are not correlated to each other. In these circumstance, it is likely that the simpler single-trait model is working better than our proposed model as the number of parameters to be learnt is smaller. In practice, we recommend to use cross-validation to determine which model is the most appropriate one.

# Chapter 6

# Discussion and outlook

In this thesis, we have studied various methods to learn the relationship between genotypes and phenotypes in complex traits. This chapter provides a summary of the findings we have retained and also gives an outlook to open problems we want to tackle in the future.

## 6.1 Thesis summary

Early work in statistical genetics dates back to Georg Mendel who discovered patterns of inheritance for certain qualitative traits in peas in 1865, henceforth called Mendelian inheritance (Mendel, 1866). In the beginning of the twentieth century, his laws were rediscovered, leading to controversies concerning the universality of the Mendelian laws; his supporters accepted that most traits can be explained by a single gene, while biometricians argued that Mendelian patterns could not account for many quantitative traits following a normal distribution (Plomin et al., 2009). In 1918, Ronald Fisher unified both theories by showing that the aggregation of multiple genetic effects leads to normally distributed phenotypes, whereas each gene is inherited according to Mendel's laws (Fisher, 1918). With the advent of genomic data, it is now possible to study this polygenic architecture of complex traits at an unprecedented level of detail (Purcell et al., 2009; Yang et al., 2010). However, many models that are used today are still overly simple, and do not take advantage of the wealth of data that is available.

In Chapter 1, we have given a short introduction to the field of genomics and presented the challenges this thesis is concerned with. Chapter 2 provides the mathematical background for the remaining thesis.

In Chapter 3, we have introduced a new mixed model approach, the LMM-Lasso,

which allows for the inclusion of markers with large effect sizes as fixed effects. The random effect of the mixed model can hereby either represent confounding effects, such as population structure, or genetic effects that are too small to be traced down to individual markers. As a result, LMM-Lasso is able to recover individual genetic effects better than other existing methods in challenging settings with complex genetic architectures, weak effects of individual markers or in the presence of strong confounding effects. Compared to a pure Lasso regression model, the coefficients are easier to interpret as the confounding factors are picked up by the random effect. This helps to resolve the ambiguity between individual genetic effects and phenotypic variability due to population structure.

In Chapter 4 and 5, we have studied the benefits of incorporating prior knowledge and of leveraging the data over multiple correlated traits. We have used out-of-sample prediction accuracy as our main evaluation criterion as it is almost assumption-free and therefore a reasonably objective criterion. In Chapter 4, we have compared a number of different models which either couple the coefficients of markers that are connected in a given biological network and/or the coefficients between the phenotypic traits that are related to each other. By means of simulations, we have verified that both is of help if the assumptions made by the model are met by the data. However, we could not confirm the same benefits on the yeast dataset we have analyzed. This leads us to the following conclusions.

1. Using biological knowledge to couple the coefficient does not improve the solution in general, while multi-task learning can increase the accuracy of phenotype predictions.

2. Coupling the magnitudes of the weight coefficients across multiple phenotypes can lead to biases if the scales of the phenotypes are different. In practice, it is hard to determine the best scaling, since the signal-to-noise ratio of the different phenotypes cannot be known in advance.

3. The grouping of the phenotypes must be executed with care, since the phenotypes can also be statistically correlated due to shared hidden confounders.

In Chapter 5, we have designed the multi-trait model GP-kronsum that avoids the shortcomings of the multi-trait models we have considered in the previous chapter. This has been achieved by learning the correlations between the weight vectors and allowing for correlated residuals. While the model is not new *per se* but has been used before (Henderson, 1984; Zhang, 2007; Korte et al., 2012), we are first in showing that efficient inference is possible for that type of models: our algorithm reduces the runtime burden from $O(N^3T^3)$ to $O(N^3 + T^3)$ and the memory requirement from

$O(N^2T^2)$ to $O(N^2+T^2)$. This makes applications with a large number of individuals and phenotypes feasible. In our experiments, we have demonstrated that our method outperforms simpler alternatives in terms of predictive power. In addition, our model is not restricted to applications in phenotype predictions, but can also be used for variance component modeling or as a background model for association testing in correlated traits.

## 6.2 Future work

This thesis has introduced new approaches to learn the mapping between genotypes and phenotypes. Based on these results, we identify several future research directions. First, we discuss extensions that are directly linked to the work presented in this thesis. Second, we provide some new and more general ideas for genomic association studies.

### 6.2.1 Combining multi-trait models with feature selection

The positive results with the LMM-Lasso and the GP-kronsum model indicate that there is potential to combine the merits of both. We can exploit the same algorithmic steps as in the LMM-Lasso algorithm, that are

1. fit a null-model
2. whiten the phenotypes and the markers
3. solve a Lasso problem on the transformed data.

The null-model fitting (step 1) is done by learning the trait-trait covariance matrices using the inference scheme presented in Chapter 5. When whitening the data, it is important to preserve its Kronecker structure. Otherwise, the feature matrix inflates into a $TN \times M$ matrix leading to an increased memory requirement and an increased runtime for the subsequent steps. This can be achieved by making use of the Kronecker tricks once again. As a result, the consequential optimization problem can no longer be cast into the standard Lasso form. However, the resulting optimization problem is still convex, and adopting an existing $\ell_1$-solver to the new task should be feasible. We have already completed some experiments combining the multi-trait model with greedy forward selection, demonstrating the synergetic effects of marker selection and multi-trait modeling. We will try to further expand this by using the Lasso instead of performing a sequential selection strategy.

### 6.2.2 Significance estimates for the LMM-Lasso

A second direction of further work is to assess the markers in terms of statistical significance. Meinshausen et al. (2009) developed an algorithm for estimating $p$-values for Lasso methods that is related to stability selection, since it also involves randomized splitting of the dataset. However, it has not yet been investigated how strong the sample size splitting affects the power of Lasso-based methods. More recently, Lockhart et al. (2013) and Javanmard and Montanari (2013) proposed further approaches for estimating the significance in Lasso models. However, none of these three methods provides an out-of-the-box solution for estimating the statistical significance of the markers in the LMM-Lasso model on a genome-wide scale. A more straight-forward approach can be obtained by the following two-step procedure: First, we train the LMM-Lasso on the complete set of markers. Second, the selected SNPs are included as additional fixed effects in a standard linear mixed model, retaining their $\ell_1$-penalty. Significance estimates can then be obtained via standard univariate testing in the linear mixed model setting (Yu et al., 2006).

The only difference is that we remove a selected marker from the model if the tested SNP is in close proximity (Listgarten et al., 2012). With this scheme, one needs to solve one Lasso problem for each test which is conducted. However, the runtime overload is small as only a limited number of markers is included as fixed effects, and the coefficients of the selected markers are similar between the different tests. One can exploit this by initializing the coefficients with the solution obtained by the initial training, i.e. warm-starts, which allows to quickly update them. The approach is conceptually close to the work of Segura et al. (2012), but avoids a sequential selection of markers.

### 6.2.3 Extending multi-trait models to more than one kernel

In Section 5, we have shown that efficient inference is feasible for a sum of two Kronecker products. Unfortunately, one cannot straightforwardly extend these results to larger sums as the sum of Kronecker products is in general not a Kronecker product itself. Models of this type are required, for example, for association testing between a group of markers and a group of correlated phenotypes, as we studied in Casale et al. (2013). A similar approach was also introduced by Price et al. (2011) for studying *cis*- and *trans*-specific gene expression levels in different tissues. Currently, analyses of this type can only be performed on small datasets for which computing the explicit covariance matrix is feasible. This leads to a cubic runtime in the number of samples and number of traits $O(N^3T^3)$ and quadratic memory requirement $O(N^2T^2)$.

We want to overcome this by employing iterative solvers which can exploit the Kronecker structure of the covariance matrix.

### 6.2.4 From multiple to complex phenotypes

Earlier in this work, we proposed methods for analyzing multiple phenotypes jointly. In the future, we would like to extend this to more complex phenotypes, like microscopy images (Meijon et al., 2014) or electronic medical records (Denny et al., 2013). Amongst others, Karaletsos et al. (2012) and Parts et al. (2011) have presented promising work in this field. They propose to first learn a latent feature representation of the phenotype and then perform association mapping between the markers and the inferred factors. By designing new association mapping techniques that allow for non-continuous phenotypes, such as graphs or trees (Feragen et al., 2013), we want to directly test for associations instead.

### 6.2.5 Association testing in linear time

In this thesis, we have worked on approaches that work well for datasets with up to several thousands of samples and hundreds of thousands of markers. While this is good enough for most datasets that are publicly available at this moment, it will not suffice for new datasets that are currently created. Large consortia, like the Genetic Investigation of Anthropometric Traits, have already gathered data for over hundreds of thousands of individuals (Berndt et al., 2013), requiring new algorithms that scale linearly with the number of samples and the number of markers. Fortunately, a number of fast and accurate sparse approximations to Gaussian processes have been proposed recently, which scale well and allow for parallel inference (Hensman et al., 2013; Gal et al., 2014). We think that similar inference schemes as presented there can also be used to accelerate association testing in linear mixed models. Moreover, we are exploring new covariance functions which might allow for faster inference. For instance, Davies and Ghahramani (2014) suggest to construct a kernel based on random partitioning of the data. The similarity between two points is thereby defined as the probability that the two points are assigned to the same cluster. In genetics, SNP-based clustering is inevitably linked to inferring the population structure (Pritchard et al., 2000), making this the perfect choice for designing a covariance function that can capture the relatedness between samples. We also started exploring a permutation-based approach to correct for population structure (Huang et al., 2013). The key idea behind this is that the exchange probability between two samples is proportional to their similarity, making it more likely that samples are shuffled

within the same population. The method is generic in two ways: firstly, we can use an arbitrary covariance function to measure the similarity between the samples, and secondly, we can choose any desired test statistic for association testing.

# Appendix A

# Appendix

In this appendix, we provide a brief refresher to probability theory and linear algebra. For a more detailed introduction of probability theory, we refer the reader to (Wasserman, 2004; Bishop, 2006) and for linear algebra to (Lay, 2012; Golub and Van Loan, 1996).

## A.1 Probability theory

In probability theory, we are given an experiment and want to assign a probability to a possible event $A$. An event is thereby a set of possible outcomes, and the set of of all possible outcomes is called the sample space $\Sigma$.

Formally, we can then define a probability function $p$, as as any function that fulfills the following three axioms:

1. $p(A) \leq 0$ for all events $A \in \Sigma$.
2. $p(\Omega) = 1$.
3. if $A_1, A_2, \ldots \in \Sigma$ are disjoint, then

$$p\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} p(A_i). \tag{A.1}$$

Probabilities can either be interpreted as "relative frequencies with which an event occurs in the long run" (Bulmer, 1979) or as a "subjective degree of belief" (Koller and Friedman, 2009), resulting in the frequentists and the Bayesian school of thought. While there has been and still is much dispute about the different merits of the two

approaches, a deeper discussion lies beyond the scope of this thesis, and can for instance be found in (Efron, 1986; MacKay, 2002). In this work, we use a pragmatic approach and apply techniques from both fields, depending on our subjective opinion what suits best for the problem at hand.

**Basic properties** Let $A$ and $B$ be two events. Then, their joint distribution is denoted as $p(A, B)$. The two events are independent, if

$$p(A, B) = p(A)p(B). \tag{A.2}$$

If $p(B) > 0$, the conditional probability of $A$ given $B$ is

$$p(A|B) = \frac{p(A, B)}{p(B)}. \tag{A.3}$$

The product rule is then obtained by rewriting the definition of the conditional probability as

$$p(A, B) = p(A)p(B|A), \tag{A.4}$$

and the sum rule is given by

$$p(A) = \sum_B p(A, B). \tag{A.5}$$

By combining the sum and product rule, we directly get to Bayes theorem

$$p(A|B) = \frac{p(B|A)p(A)}{\sum_B p(B|A)p(A)}. \tag{A.6}$$

**Random variable** A random variable describes a mapping $X : \Omega \to \mathbb{R}$ from the sample space to the real numbers. Random variables can either be discrete, taking a countable list of values, or continuous, taking any value in one or multiple intervals. From now on, we concentrate on the continuous case.

The cumulative distribution function $F_X : \mathbb{R} \to [0, 1]$ of a random variable $X$ is given by

$$F_X(x) = p(X \leq 0). \tag{A.7}$$

If the random variable $X$ is continuous, the corresponding probability density function $f_X$ fulfills the following properties:

1. $f_X(x) \geq 0$
2. $\int_{-\infty}^{\infty} f_X(x)dx = 1$
3. $p(a < X < b) = \int_a^b f_X(x)dx$ for $a \leq b$.

The cumulative distribution function $F_X$ and the probability density function $f_X$ are then linked via the identity

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt. \tag{A.8}$$

**Moments** The expected value of a random variable $X$ is defined as

$$\mathbb{E}[X] = \int x f(x)dx. \tag{A.9}$$

and its variance as

$$\mathrm{Var}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]. \tag{A.10}$$

The standard deviation $\sigma$ of the random variable $X$ is defined as the square root of its variance.

**Gaussian distribution** The Gaussian distribution is parameterized by the mean $\mu$ and the variance $\sigma^2$, and defined by the probability density function

$$\mathcal{N}\left(x \,|\mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{1}{2\sigma^2}(x-\mu)^2. \tag{A.11}$$

The distribution is bell-shaped around the mean and its width is determined by the variance parameter, see also Figure A.1. Its extension to $d$ dimensions is given by

$$\mathcal{N}(\boldsymbol{x}\,|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d\,|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{A.12}$$

where $\boldsymbol{\mu}$ is now the $d$-dimensional mean vector, and $\boldsymbol{\Sigma}$ is the $d \times d$ covariance matrix.

The importance of the Gaussian distribution is grounded by the central limit theorem. It states that, under mild conditions, the sum of a large number of independent random variables, that need not necessarily be Gaussian distributed by themselves, will be approximately Gaussian distributed (Bulmer, 1979).

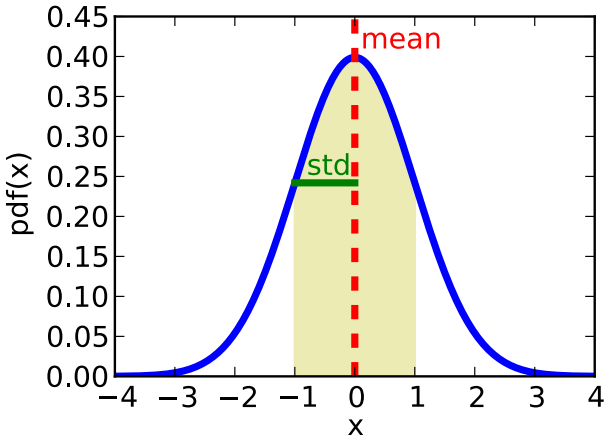



Figure A.1: **One-dimensional Gaussian distribution.** Gaussian probability density function (pdf), with mean $\mu = 0$ and variance $\sigma^2 = 1$, is shown in blue. The mean is depicted with red, the standard deviation $\sigma$ with green. 68.27% of the mass of the distribution is within one standard deviation of the mean (yellow-shaded area).

Apart from that, the Gaussian distribution is also loved for its analytically tractability. If the marginal distribution $p(\boldsymbol{x})$ and the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ are normally distributed

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}\,|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right), \tag{A.13}$$
$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{y}\,|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}\right), \tag{A.14}$$

it is easy to show that the marginal distribution of $p(\boldsymbol{y})$ and the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y})$ are again Gaussians, and have a closed-form solution for the mean and the covariance (Bishop, 2006)

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}\,|\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{A}\boldsymbol{\Lambda}^{-1}\boldsymbol{A}^{\top}\right), \tag{A.15}$$
$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{x}\,|\boldsymbol{\Sigma}\left[\boldsymbol{A}^{\top}\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\right], \boldsymbol{\Sigma}\right), \tag{A.16}$$

where $\boldsymbol{\Sigma} = \left(\boldsymbol{\Lambda} + \boldsymbol{A}^{\top}\boldsymbol{L}\boldsymbol{A}\right)^{-1}$.

**Prior, posterior and likelihood** In Bayesian statistics, the parameters $\boldsymbol{\theta}$ are not fixed, but random variables. They are endowed with a prior distribution $p(\boldsymbol{\theta})$ that reflects our beliefs, before we have seen any data. The likelihood function $p(D|\boldsymbol{\theta})$ connects the parameters with the data: it describes how good the parameters $\boldsymbol{\theta}$ can explain the dataset $D$. After having observed the data, we can then update our beliefs of the parameters $\boldsymbol{\theta}$ by using Bayes theorem

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} \tag{A.17}$$

yielding the posterior distribution $p(\boldsymbol{\theta}|D)$. The maximum a posteriori probability (MAP) of $\boldsymbol{\theta}$ is the mode of the posterior. If the prior is flat, that means it assigns the same probability to all parameter values, the MAP estimate coincides with the maximum likelihood solution. The normalization constant $\mathrm{p}(D)$ is often also called the evidence of the data

$$p(D) = \int p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{A.18}$$

and can be used for model comparisons. For instance, a popular way to determine the hyperparameters, that are the parameters of the prior, is to maximize the evidence function.

## A.2 Linear algebra

A vector $\boldsymbol{a}$ is of the form

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \tag{A.19}$$

where $a_i \in \mathbb{R}$ is the $i$th entry of $\boldsymbol{a}$. A matrix $\boldsymbol{A}$ is the concatenation of vectors

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_m, \end{pmatrix}. \tag{A.20}$$

The matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ consists of $m$ columns and $n$ rows. The columns of $\boldsymbol{A}$ are described by the vectors $\boldsymbol{a}_i$. The $(i, j)$ entry of the matrix $\boldsymbol{A}$ is indexed by $A_{ij}$. A diagonal matrix $\boldsymbol{D}$ is a square $n \times n$ matrix, that is zero for all non-diagonal entries. The identity matrix $\boldsymbol{I}$ is a diagonal matrix, whose diagonal entries are one.

**Matrix multiplication** Let $\boldsymbol{A}$ be a $m \times n$ matrix, $\boldsymbol{B}$ an $n \times p$ and $\boldsymbol{C}$ a $p \times q$ matrix. Then, the following identities are satisfied:

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{BC}) &= (\boldsymbol{AB})\boldsymbol{C} && \text{(A.21)} \\ \boldsymbol{A}(\boldsymbol{B}+\boldsymbol{C}) &= \boldsymbol{AB}+\boldsymbol{AC} && \text{(A.22)} \\ (\boldsymbol{B}+\boldsymbol{C})\boldsymbol{A} &= \boldsymbol{BA}+\boldsymbol{CA} && \text{(A.23)} \end{aligned}$$

In general, $\boldsymbol{AB} = \boldsymbol{BA}$ does not hold.

**Transpose** The transpose of a $m \times n$ matrix $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_m, \end{pmatrix}$ is obtained by writing the columns of $\boldsymbol{A}$ as the rows,

$$\boldsymbol{A}^\top = \begin{pmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \\ \vdots \\ \boldsymbol{a}_m^\top \end{pmatrix}, \tag{A.24}$$

yielding a $n \times m$ matrix. Let $\boldsymbol{B}$ be a $n \times p$ matrix. Then $(\boldsymbol{AB})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top$. A matrix is called symmetric if $\boldsymbol{A} = \boldsymbol{A}^\top$.

**Inverse** The matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is invertible if there exists a matrix $\boldsymbol{A}^{-1} \in \mathbb{R}^{n \times n}$, such that

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I} \tag{A.25}$$
$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I} \tag{A.26}$$

If a matrix is not invertible, it is also called singular. Let $\boldsymbol{B}$ be a second $n \times n$ matrix. Provided both matrices are not singular, the inverse of its product is given by

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}. \tag{A.27}$$

Let $\boldsymbol{A}$ be an invertible $n \times n$ matrix, and $\boldsymbol{U}, \boldsymbol{V}$ be two $n \times p$ matrices. The Woodbury identity (Golub and Van Loan, 1996) can significantly speed-up computations, if $\boldsymbol{A}$ is easy to invert and $p \ll n$:

$$\left(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{V}^\top\right)^{-1} = \boldsymbol{A}^{-1} - \left[\boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{I} + \boldsymbol{V}^\top\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}^\top\boldsymbol{A}^{-1}\right]. \tag{A.28}$$

**Trace** The trace of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is defined as the sum over its diagonal entries. The following equalities hold for $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{n \times n}$

$$\mathrm{Tr}(\boldsymbol{A}) = \mathrm{Tr}\left(\boldsymbol{A}^\top\right) \tag{A.29}$$
$$\mathrm{Tr}(\boldsymbol{ABC}) = \mathrm{Tr}(\boldsymbol{CBA}) = \mathrm{Tr}(\boldsymbol{BAC}). \tag{A.30}$$

**Determinants** The determinant $|\cdot|$ of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is defined as

$$|\boldsymbol{A}| = \sum_\sigma (\pm 1)^{N_\sigma} A_{1,i_1} \cdot A_{1,i_2} \cdot \ldots \cdot A_{1,i_n} \tag{A.31}$$

where the sum is going over all permutations $\sigma = (i_1, \ldots, i_n)$ and $N_\sigma$ is the number of pairwise permutations needed to go from $1, 2, \ldots, n$ to $i_1, i_2, \ldots, i_n$. Given $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}, c \in \mathbb{R}$, we can exploit the following identities

$$\begin{aligned} |\boldsymbol{AB}| &= |\boldsymbol{A}| \cdot |\boldsymbol{B}| && \text{(A.32)} \\ \left|\boldsymbol{A}^{-1}\right| &= |\boldsymbol{A}|^{-1} && \text{(A.33)} \\ |c\boldsymbol{A}| &= c^n \, |\boldsymbol{A}| \, . && \text{(A.34)} \end{aligned}$$

**Eigendecomposition** An eigenvector of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is a nonzero vector $\boldsymbol{u}$ such that

$$\boldsymbol{Au} = \lambda \boldsymbol{u}, \tag{A.35}$$

where $\lambda$ is the corresponding eigenvalue. If the matrix $\boldsymbol{A}$ is in addition symmetric, then there exists a orthogonal matrix $\boldsymbol{U}$, such that

$$\boldsymbol{A} = \boldsymbol{USU}^\top, \tag{A.36}$$

where $\boldsymbol{S} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix, containing the eigenvalues of $\boldsymbol{A}$, and $\boldsymbol{U} = \begin{pmatrix} \boldsymbol{u}_1 & \ldots \boldsymbol{u}_n \end{pmatrix}$ contains as columns the eigenvectors of $\boldsymbol{A}$. The eigenvalues of a general matrix $\boldsymbol{A}$ are complex, while they are real for any symmetric matrix. Given the eigenvalue decomposition of $\boldsymbol{A}$, we can compute its determinant, trace and inverse as

$$|\boldsymbol{A}| = \prod_{i=1}^{n} \lambda_i, \qquad \operatorname{Tr}\boldsymbol{A} = \textstyle\sum_{i=1}^{n} \lambda_i, \qquad \boldsymbol{A}^{-1} = \boldsymbol{US}^{-1}\boldsymbol{U}^\top. \tag{A.37}$$

**Non-negative matrices** A square $n \times n$ matrix $\boldsymbol{A}$ is said to be positive-semidefinite, if

$$\boldsymbol{x}^\top \boldsymbol{Ax} \geq \boldsymbol{0} \tag{A.38}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$. If a matrix is positive-semidefinite, all of its eigenvalues are non-negative.

**Singular value decomposition** Let $\boldsymbol{A}$ be an arbitrary $m \times n$ matrix. Then, its singular value decomposition is given by

$$\boldsymbol{A} = \boldsymbol{USV}^\top, \tag{A.39}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ contains the left-singular vectors, $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ the right-singular vectors, and $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ the singular values of $\boldsymbol{A}$ on its diagonal. The eigenvalue decomposition and the singular value decomposition are closely linked, as the left singular vectors are the eigenvectors of $\boldsymbol{A}^\top \boldsymbol{A}$, the right singular vectors the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^\top$, and the non-zero singular values are the square roots of the non-zero eigenvalues of $\boldsymbol{A}\boldsymbol{A}^\top$ and $\boldsymbol{A}\boldsymbol{A}^\top$.

**Gradient** The derivative of a vector $\boldsymbol{x} \in \mathbb{R}^m$ with respect to a scalar $y$ is a again a vector of size $m$, for which the $i$th entry is defined as

$$\left(\frac{\partial \boldsymbol{x}}{\partial y}\right)_i = \frac{\partial x_i}{\partial y} \tag{A.40}$$

Analogously, the derivative of a scalar $x$ with respect to a vector $\boldsymbol{y} \in \mathbb{R}^n$ is a vector of size $n$, with the entries

$$\left(\frac{\partial x}{\partial \boldsymbol{y}}\right)_i = \frac{\partial x}{\partial y_i}. \tag{A.41}$$

The same rules apply for derivatives of a vector $\boldsymbol{x}$ with respect to a vector $\boldsymbol{y}$, and their extensions to general matrices. The gradient vector of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\nabla_{\boldsymbol{x}} f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}. \tag{A.42}$$

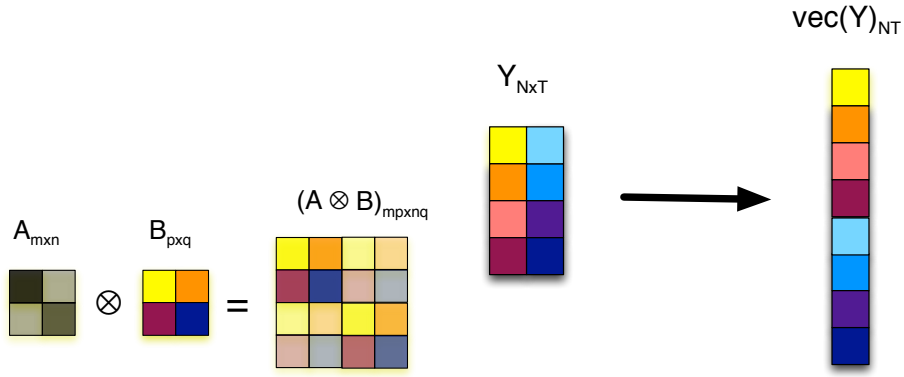
In the following, we provide a short list of well known derivative rules that were used throughout the thesis. For a more exhaustive enumeration, we refer the reader to (Petersen and Pedersen, 2012).

$$\frac{\partial}{\partial \theta}(\boldsymbol{X} + \boldsymbol{Y}) = \left(\frac{\partial}{\partial \theta}\boldsymbol{X}\right) + \left(\frac{\partial}{\partial \theta}\boldsymbol{Y}\right) \tag{A.43}$$

$$\frac{\partial}{\partial \theta}(\boldsymbol{X} \otimes \boldsymbol{Y}) = \left(\frac{\partial}{\partial \theta}\boldsymbol{X} \otimes \boldsymbol{Y}\right) + \boldsymbol{X} \otimes \left(\frac{\partial}{\partial \theta}\boldsymbol{Y}\right) \tag{A.44}$$

$$\frac{\partial}{\partial \theta}\boldsymbol{X}^{-1} = -\boldsymbol{X}^{-1}\left(\frac{\partial}{\partial \theta}\boldsymbol{X}\right)\boldsymbol{X}^{-1} \tag{A.45}$$

$$\frac{\partial}{\partial \theta}\log\left|\boldsymbol{X}^{-1}\right| = \mathrm{Tr}\left(\boldsymbol{X}^{-1}\left(\frac{\partial}{\partial \theta}\boldsymbol{X}\right)\right) \tag{A.46}$$




(a) Kronecker Product (b) vec operator

Figure A.2: **Graphical description of the Kronecker product and the vec operator.** Left: The Kronecker product is a matrix product that multiplies each element of $\boldsymbol{A}$ with the complete matrix $\boldsymbol{B}$. Right: The vec operator reshapes a matrix into a vector by concatenating the columns of the matrix.

## A.3 Kronecker product

Let $\boldsymbol{A}$ be a $m \times n$ matrix, and $\boldsymbol{B}$ be a $p \times q$ matrix. The Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ is a $mp \times nq$ matrix and defined as follows:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} A_{11}\boldsymbol{B} & \dots & A_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ A_{m1}\boldsymbol{B} & \dots & A_{mn}\boldsymbol{B} \end{pmatrix} \tag{A.47}$$

The following equalities hold (Bernstein, 2009):

$$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = \boldsymbol{AC} \otimes \boldsymbol{BD} \tag{A.48}$$

$$(\boldsymbol{A} \otimes \boldsymbol{B})^\top = \boldsymbol{A}^\top \otimes \boldsymbol{B}^\top \tag{A.49}$$

$$(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = \boldsymbol{A}^{-1} \otimes \boldsymbol{B}^{-1} \tag{A.50}$$

$$|\boldsymbol{A} \otimes \boldsymbol{B}| = |\boldsymbol{A}|^p \cdot |\boldsymbol{B}|^n \tag{A.51}$$

$$(\boldsymbol{A} \otimes \boldsymbol{B})\,\mathrm{vec}(\boldsymbol{Y}) = \mathrm{vec}\left(\boldsymbol{B}\boldsymbol{Y}\boldsymbol{A}^\top\right), \tag{A.52}$$

where $\mathrm{vec}(\boldsymbol{Y})$ is obtained by vertical concatenation of the columns of $\boldsymbol{Y}$. A graphical description of the Kronecker product and the vec operator is given in A.2.

Let $\boldsymbol{U}_A \boldsymbol{S}_A \boldsymbol{U}_A^\top$ be the eigenvalue decomposition of $\boldsymbol{A}$ and $\boldsymbol{U}_B \boldsymbol{S}_B \boldsymbol{U}_B^\top$ the eigenvalue decomposition of $\boldsymbol{B}$, then

$$(\boldsymbol{U}_A \otimes \boldsymbol{U}_B) \left(\boldsymbol{S}_A \otimes \boldsymbol{S}_B + \sigma^2 \boldsymbol{I}\right) \left(\boldsymbol{U}_A^\top \otimes \boldsymbol{U}_B^\top\right) \tag{A.53}$$

is the eigenvalue decomposition of $\boldsymbol{A} \otimes \boldsymbol{B} + \sigma^2 \boldsymbol{I}$, where $\sigma^2$ is a non-negative scalar.

# List of figures

# List of tables

# Bibliography

G. R. Abecasis, A. Auton, L. D. Brooks, M. A. DePristo, R. M. Durbin, R. E. Handsaker, H. M. Kang, G. T. Marth, G. A. McVean, D. M. Altshuler, and et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491 (7422):56–65, Nov 2012.

M. A. Álvarez and N. D. Lawrence. Sparse convolved gaussian processes for multi-output regression. In *NIPS*, pages 57–64, 2008.

C. Archembeau, S. Guo, and O. Zoeter. Sparse Bayesian Multi-Task Learning, 2011.

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

S. Atwell, Y. S. Huang, B. J. Vilhjalmsson, G. Willems, M. Horton, Y. Li, D. Meng, A. Platt, A. M. Tarone, T. T. Hu, and et al. Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines. *Nature*, 465(7298):627–631, Jun 2010.

C. A. Azencott, D. Grimm, M. Sugiyama, Y. Kawahara, and K. M. Borgwardt. Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics*, 29(13):i171–179, Jul 2013.

D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA, 2012. ISBN 0521518148, 9780521518147.

S. I. Berndt, S. Gustafsson, R. Magi, A. Ganna, E. Wheeler, M. F. Feitosa, A. E. Justice, K. L. Monda, D. C. Croteau-Chonka, F. R. Day, T. Esko, and et al. Genome-wide meta-analysis identifies 11 new loci for anthropometric traits and provides insights into genetic architecture. *Nat. Genet.*, 45(5):501–512, May 2013.

D. S. Bernstein. *Matrix mathematics : theory, facts, and formulas*. Princeton University Press, Princeton, N.J., Woodstock, 2009. ISBN 978-0-691-14039-1. URL `http://opac.inria.fr/record=b1128603`. Prcdente ed.: 2005.

J. Bigler, J. Whitton, J. W. Lampe, L. Fosdick, R. M. Bostick, and J. D. Potter. CYP2C9 and UGT1A6 genotypes modulate the protective effect of aspirin on colon adenoma risk. *Cancer Res.*, 61(9):3566–3569, May 2001.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

J. S. Bloom, I. M. Ehrenreich, W. T. Loo, T. L. Lite, and L. Kruglyak. Finding the sources of missing heritability in a yeast cross. *Nature*, 494(7436):234–237, Feb 2013.

E. V. Bonilla, K. M. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008. URL `nips08.pdf`.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL `http://dx.doi.org/10.1561/2200000016`.

J. R. Broach. Ras-regulated signaling processes in Saccharomyces cerevisiae. *Curr. Opin. Genet. Dev.*, 1(3):370–377, Oct 1991.

P. Buehlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer, 2011. ISBN 9783642201929. URL `http://books.google.de/books?id=S6jYXmh988UC`.

G. Bulmer. *Principles of Statistics*. Dover Books on Mathematics Series. Dover Publications, 1979. ISBN 9780486637600. URL `http://books.google.de/books?id=dh24EaSrmBkC`.

P. Carbonetto and M. Stephens. Scalable Variational Inference for Bayesian Variable Selection in Regression, and its Accuracy in Genetic Association Studies. *Bayesian Analysis*, 7:73–108, 2012. ISSN 1931-6690. doi: 10.1214/12-ba703.

R. Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997. ISSN 0885-6125. doi: 10.1023/A:1007379606734. URL `http://dx.doi.org/10.1023/A:1007379606734`.

F. P. Casale, B. Rakitsch, C. Lippert, and O. Stegle. A mixed-model approach for association studies of multiple variants to a set of correlated traits, 2013. MLSB 2013: The seventh workshop on Machine Learning in Systems Biology, Berlin, July 19-20, 2013.

A. T. Chan, G. J. Tranah, E. L. Giovannucci, D. J. Hunter, and C. S. Fuchs. Genetic variants in the UGT1A6 enzyme, aspirin use, and the risk of colorectal adenoma. *J. Natl. Cancer Inst.*, 97(6):457–460, Mar 2005.

S. S. Chen, D. L. Donoho, Michael, and A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

X. Chen et al. A two-graph guided multi-task lasso approach for eqtl mapping. *Journal of Machine Learning Research - Proceedings Track*, 22:208–217, 2012.

J. M. Cherry, E. L. Hong, C. Amundsen, R. Balakrishnan, G. Binkley, E. T. Chan, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. R. Engel, D. G. Fisk, J. E. Hirschman, B. C. Hitz, K. Karra, C. J. Krieger, S. R. Miyasato, R. S. Nash, J. Park, M. S. Skrzypek, M. Simison, S. Weng, and E. D. Wong. Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic acids research*, 40(Database issue):D700–5, Jan. 2012.

G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `http://igraph.sf.net`.

A. Dahl, V. Hore, V. Iotchkova, and J. Marchini. Network inference in matrix-variate Gaussian models with non-independent noise. *ArXiv e-prints*, Dec. 2013.

A. Davies and Z. Ghahramani. The Random Forest Kernel and other kernels for big data from random partitions. *ArXiv e-prints*, Feb. 2014.

G. de los Campos, D. Gianola, and D. B. Allison. Predicting genetic predisposition in humans: the promise of whole-genome markers. *Nat. Rev. Genet.*, 11(12):880–886, Dec 2010.

I. J. Deary, J. Yang, G. Davies, S. E. Harris, A. Tenesa, D. Liewald, M. Luciano, L. M. Lopez, A. J. Gow, J. Corley, P. Redmond, H. C. Fox, S. J. Rowe, P. Haggarty, G. McNeill, M. E. Goddard, D. J. Porteous, L. J. Whalley, J. M. Starr, and P. M. Visscher. Genetic contributions to stability and change in intelligence from childhood to old age. *Nature*, 482(7384):212–215, Feb 2012.

J. C. Denny, L. Bastarache, M. D. Ritchie, R. J. Carroll, R. Zink, J. D. Mosley, J. R. Field, J. M. Pulley, A. H. Ramirez, E. Bowton, and et al. Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. *Nat. Biotechnol.*, 31(12):1102–1110, Dec 2013.

P. Domingos. A unified bias-variance decomposition for zero-one and squared loss. In H. A. Kautz and B. W. Porter, editors, *AAAI/IAAI*, pages 564–569. AAAI Press / The MIT Press, 2000. ISBN 0-262-51112-6.

V. Ducrocq and H. Chapuis. Generalizing the use of the canonical transformation for the solution of multivariate mixed model equations. *Genetics Selection Evolution*, 29(2):205–224, 1997.

B. Efron. Why Isn't Everyone a Bayesian? *The American Statistician*, 40(1):1–5, Feb. 1986. ISSN 00031305. doi: 10.2307/2683105. URL `http://dx.doi.org/10.2307/2683105`.

B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Ann Stat*, 32:407–499, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.8168`.

E. E. Eichler, J. Flint, G. Gibson, A. Kong, S. M. Leal, J. H. Moore, and J. H. Nadeau. Missing heritability and strategies for finding the underlying causes of complex disease. *Nat. Rev. Genet.*, 11(6):446–450, Jun 2010.

L. Fahrmeir, T. Kneib, and S. Lang. *Regression*. Statistik und ihre Anwendungen. Springer, 2009. ISBN 9783642018374.

A. Feragen, J. Petersen, D. Grimm, A. Dirksen, J. Holst Pedersen, K. Borgwardt, and M. de Bruijne. Geometric tree kernels: Classification of COPD from airway tree geometry. *ArXiv e-prints*, Mar. 2013.

R. Fisher. The correlation between relatives on the supposition of mendelian inheritance. *Trans. Roy. Soc. Edinb.*, 52:399–433, 1918.

J. Flint and E. Eskin. Genome-wide association studies in mice. *Nat. Rev. Genet.*, 13(11):807–817, Nov 2012.

S. Foster, A. Verbyla, and W. Pitchford. Incorporating lasso effects into a mixed model for quantitative trait loci detection. *J Agric Biol Environ Stat*, 12:300–314, 2007. ISSN 1085-7117. URL `http://dx.doi.org/10.1198/108571107X200396`. 10.1198/108571107X200396.

J. L. Foulley and D. A. van Dyk. The PX-EM algorithm for fast stable fitting of Henderson's mixed model. *Genet. Sel. Evol.*, 32(2):143–163, 2000.

A. Franceschini et al. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.*, 41(D1):D808–815, Jan 2013.

K. A. Frazer, D. G. Ballinger, D. R. Cox, D. A. Hinds, L. L. Stuve, R. A. Gibbs, J. W. Belmont, A. Boudreau, P. Hardenbol, S. M. Leal, and et al. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449(7164):851–861, Oct 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008. ISSN 1468-4357. doi: 10.1093/biostatistics/kxm045.

W. J. Fu. Penalized Regressions: The Bridge versus the Lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998. ISSN 10618600. doi: 10.2307/1390712. URL `http://dx.doi.org/10.2307/1390712`.

N. Fusi, O. Stegle, and N. D. Lawrence. Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies. *PLoS Comput. Biol.*, 8(1):e1002330, Jan 2012.

J. Gagneur, O. Stegle, C. Zhu, P. Jakob, M. M. Tekkedil, R. S. Aiyar, A. K. Schuon, D. Pe'er, and L. M. Steinmetz. Genotype-environment interactions reveal causal pathways that mediate genetic effects on phenotype. *PLoS Genet.*, 9(9):e1003803, 2013.

Y. Gal, M. van der Wilk, and C. E. Rasmussen. Distributed Variational Inference in Sparse Gaussian Process Regression and Latent Variable Models. *ArXiv e-prints*, Feb. 2014.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Comput.*, 4(1):1–58, Jan. 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.1.1. URL `http://dx.doi.org/10.1162/neco.1992.4.1.1`.

Z. Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2013.

M. E. Goddard, N. R. Wray, K. Verbyla, and P. M. Visscher. Estimating effects and making predictions from genome-wide marker data. *Stat Sci*, 24(4):517–529, Nov. 2009. doi: 10.1214/09-STS306.

A. Goldberger. Best Linear Unbiased Prediction in the Generalized Linear Regression Model. *JASA*, 57, 1962.

G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

B. J. Hayes, P. M. Visscher, and M. E. Goddard. Increased accuracy of artificial selection by using the realized relationship matrix. *Genet Res (Camb)*, 91(1): 47–60, Feb 2009.

Q. He and D. Y. Lin. A variable selection method for genome-wide association studies. *Bioinformatics*, 27(1):1–8, Jan 2011.

C. Henderson. Estimation of genetic parameters. *Biometrics*, 6:186, 1950.

C. Henderson. *Applications of Linear Models in Animal Breeding*. University of Guelph, 1984. ISBN 9780889550308. URL `http://books.google.de/books?id=3uB6QgAACAAJ`.

C. Henderson, O. Kempthorne, S. Searle, and C. V. Krosigk. The estimation of environmental and genetic trends from records subject to culling. *Statistical Genetics and Plant Breeding*, 15:192–218, 1959.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *CoRR*, abs/1309.6835, 2013.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

C. J. Hoggart, J. C. Whittaker, M. De Iorio, and D. J. Balding. Simultaneous analysis of all SNPs in genome-wide and re-sequencing association studies. *PLoS Genet.*, 4(7):e1000130, 2008.

M. W. Horton, A. M. Hancock, Y. S. Huang, C. Toomajian, S. Atwell, A. Auton, N. W. Muliyati, A. Platt, F. G. Sperone, B. J. Vilhjalmsson, M. Nordborg, J. O. Borevitz, and J. Bergelson. Genome-wide patterns of genetic variation in world-wide Arabidopsis thaliana accessions from the RegMap panel. *Nat. Genet.*, 44(2): 212–216, Feb 2012.

H. Huang, G. Peloso, B. Rakitsch, J. Goldstein, S. Purcell, M. Daly, K. Borgwardt, and B. Neale, 2013. WCPG XXI, World Congress of Psyciatric Genetics, Boston, October 17-21, 2013.

X. Huang, X. Wei, T. Sang, Q. Zhao, Q. Feng, Y. Zhao, C. Li, C. Zhu, T. Lu, Z. Zhang, M. Li, D. Fan, Y. Guo, A. Wang, L. Wang, L. Deng, W. Li, Y. Lu, Q. Weng, K. Liu, T. Huang, T. Zhou, Y. Jing, W. Li, Z. Lin, E. S. Buckler, Q. Qian, Q. F. Zhang, J. Li, and B. Han. Genome-wide association studies of 14 agronomic traits in rice landraces. *Nat. Genet.*, 42(11):961–967, Nov 2010.

D. J. Hunter. Gene-environment interactions in human diseases. *Nat. Rev. Genet.*, 6(4):287–298, Apr 2005.

A. Javanmard and A. Montanari. Confidence Intervals and Hypothesis Testing for High-Dimensional Regression. *ArXiv e-prints*, June 2013.

R. Jelier, J. I. Semple, R. Garcia-Verdugo, and B. Lehner. Predicting phenotypic variation in yeast from individual genome sequences. *Nature Genetics*, pages 1–7, Nov. 2011.

Y. Jia and J. L. Jannink. Multiple-trait genomic selection methods increase genetic value prediction accuracy. *Genetics*, 192(4):1513–1522, Dec 2012.

C. Jiang and Z. B. Zeng. Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics*, 140(3):1111–1127, Jul 1995.

A. A. Kalaitzis and N. D. Lawrence. Residual components analysis. In *ICML*, 2012.

H. M. Kang, N. A. Zaitlen, C. M. Wade, A. Kirby, D. Heckerman, M. J. Daly, and E. Eskin. Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709–1723, Mar 2008.

H. M. Kang, J. H. Sul, S. K. Service, N. A. Zaitlen, S. Y. Kong, N. B. Freimer, C. Sabatti, and E. Eskin. Variance component model to account for sample structure in genome-wide association studies. *Nat. Genet.*, 42(4):348–354, Apr 2010.

T. Karaletsos, O. Stegle, C. Dreyer, J. Winn, and K. M. Borgwardt. ShapePheno: unsupervised extraction of shape phenotypes from biological image collections. *Bioinformatics*, 28(7):1001–1008, Apr 2012.

M. K. Kerr and G. A. Churchill. Experimental design for gene expression microarrays. *Biostatistics*, 2(2):183–201, Jun 2001.

B. Khor, A. Gardet, and R. J. Xavier. Genetics and pathogenesis of inflammatory bowel disease. *Nature*, 474(7351):307–317, Jun 2011.

S. Kim and E. P. Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genet.*, 5(8):e1000587, Aug 2009.

S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, pages 543–550, 2010.

S. Kim, K.-A. Sohn, and E. Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics (Oxford, England)*, 25(12):12, 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp218. URL `http://dx.doi.org/10.1093/bioinformatics/btp218`.

S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research*, 2007, 2007.

Y. Kirino, G. Bertsias, Y. Ishigatsubo, N. Mizuki, I. Tugal-Tutkun, E. Seyahi, Y. Ozyazgan, F. S. Sacli, B. Erer, H. Inoko, and et al. Genome-wide association analysis identifies new susceptibility loci for Behet's disease and epistasis between HLA-B*51 and ERAP1. *Nat. Genet.*, 45(2):202–207, Feb 2013.

E. Knight. *Improved Iterative Schemes for REML Estimation of Variance Parameters in Linear Mixed Models*. University of Adelaide, School of Agriculture, Food and Wine, Discipline of Biometrics SA, 2008. URL `http://books.google.de/books?id=kbr6PgAACAAJ`.

S. A. Knott and C. S. Haley. Multitrait least squares for quantitative trait loci detection. *Genetics*, 156(2):899–911, Oct 2000.

M. A. Kohli, S. Lucae, P. G. Saemann, M. V. Schmidt, A. Demirkan, K. Hek, D. Czamara, M. Alexander, D. Salyakina, S. Ripke, and et al. The neuronal transporter gene SLC6A15 confers risk to major depression. *Neuron*, 70(2):252–265, Apr 2011.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

A. Korte, B. J. Vilhjalmsson, V. Segura, A. Platt, Q. Long, and M. Nordborg. A mixed-model approach for genome-wide association studies of correlated traits in structured populations. *Nat. Genet.*, 44(9):1066–1071, Sep 2012.

E. S. Lander and N. J. Schork. Genetic dissection of complex traits. *Science*, 265 (5181):2037–2048, Sep 1994.

E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, and et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, Feb 2001.

H. Lango Allen, K. Estrada, G. Lettre, S. I. Berndt, M. N. Weedon, F. Rivadeneira, C. J. Willer, A. U. Jackson, S. Vedantam, S. Raychaudhuri, and et al. Hundreds of variants clustered in genomic loci and biological pathways affect human height. *Nature*, 467(7317):832–838, Oct 2010.

D. C. Lay. *Linear Algebra and its Applications*. Addison-Wesley, 2012.

S. Lee and E. P. Xing. Leveraging input and output structures for joint mapping of epistatic and marginal eQTLs. *Bioinformatics*, 28(12):i137–146, Jun 2012.

S. Lee, J. Zhu, and E. P. Xing. Adaptive multi-task lasso: with application to eqtl detection. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *NIPS*, pages 1306–1314. Curran Associates, Inc., 2010.

S. H. Lee, J. Yang, M. E. Goddard, P. M. Visscher, and N. R. Wray. Estimation of pleiotropy between complex diseases using single-nucleotide polymorphism-derived genomic relationships and restricted maximum likelihood. *Bioinformatics*, 28(19): 2540–2542, Oct 2012.

S. I. Lee, A. M. Dudley, D. Drubin, P. A. Silver, N. J. Krogan, D. Pe'er, and D. Koller. Learning a prior on regulatory potential from eQTL data. *PLoS Genet.*, 5(1): e1000358, Jan 2009.

J. T. Leek and J. D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet.*, 3(9):1724–1735, Sep 2007.

C. Li and H. Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, May 2008.

J. Li, K. Das, G. Fu, R. Li, and R. Wu. The bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523, 2011.

C. Lippert, J. Listgarten, Y. Liu, C. M. Kadie, R. I. Davidson, and D. Heckerman. FaST linear mixed models for genome-wide association studies. *Nat. Methods*, 8 (10):833–835, 2011.

C. Lippert, G. Quon, E. Y. Kang, C. M. Kadie, J. Listgarten, and D. Heckerman. The benefits of selecting phenotype-specific variants for applications of mixed models in genomics. *Sci Rep*, 3:1815, 2013.

J. Listgarten, C. Kadie, E. E. Schadt, and D. Heckerman. Correction for hidden confounders in the genetic analysis of gene expression. *Proc. Natl. Acad. Sci. U.S.A.*, 107(38):16465–16470, Sep 2010.

J. Listgarten, C. Lippert, C. M. Kadie, R. I. Davidson, E. Eskin, and D. Heckerman. Improved linear mixed models for genome-wide association studies. *Nat. Methods*, 9(6):525–526, Jun 2012.

D. Liu, X. Lin, and D. Ghosh. Semiparametric regression of multidimensional genetic pathway data: least-squares kernel machines and linear mixed models. *Biometrics*, 63(4):1079–1088, Dec 2007.

R. Lockhart, J. Taylor, R. J. Tibshirani, and R. Tibshirani. A significance test for the lasso. *ArXiv e-prints*, Jan. 2013.

D. J. C. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, NATO ASI Series, pages 133–166. Kluwer, 1998.

D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. ISBN 0521642981.

T. F. Mackay, E. A. Stone, and J. F. Ayroles. The genetics of quantitative traits: challenges and prospects. *Nat. Rev. Genet.*, 10(8):565–577, Aug 2009.

B. Maher. Personal genomes: The case of the missing heritability. *Nature*, 456(7218): 18–21, Nov 2008.

T. A. Manolio. Bringing genome-wide association findings into clinical use. *Nat. Rev. Genet.*, 14(8):549–558, Aug 2013.

T. A. Manolio, F. S. Collins, N. J. Cox, D. B. Goldstein, L. A. Hindorff, D. J. Hunter, M. I. McCarthy, E. M. Ramos, L. R. Cardon, A. Chakravarti, J. H. Cho, A. E. Guttmacher, A. Kong, L. Kruglyak, E. Mardis, C. N. Rotimi, M. Slatkin, D. Valle, A. S. Whittemore, M. Boehnke, A. G. Clark, E. E. Eichler, G. Gibson, J. L. Haines, T. F. Mackay, S. A. McCarroll, and P. M. Visscher. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, Oct 2009.

J. Marchini, L. R. Cardon, M. S. Phillips, and P. Donnelly. The effects of human population structure on large genetic association studies. *Nat. Genet.*, 36(5):512–517, May 2004.

M. Meijon, S. B. Satbhai, T. Tsuchimatsu, and W. Busch. Genome-wide association study using cellular traits identifies a new regulator of root development in Arabidopsis. *Nat. Genet.*, 46(1):77–81, Jan 2014.

N. Meinshausen and P. Bühlmann. Stability selection. *J R Stat Soc Series B Stat Methodol*, 72:417–473, 2010. doi: 10.1111/j.1467-9868.2010.00740.x.

N. Meinshausen, L. Meier, and P. Bühlmann. p-values for high-dimensional regression. *Journal of the American Statistical Association*, 104(488):1671–1681, 2009. doi: 10.1198/jasa.2009.tm08647.

G. Mendel. Versuche über pflanzenhybriden. *Verhandlungen des naturforschenden Vereines in Brünn, Bd. IV für das Jahr, 1865*, pages 3–47, 1866.

T. H. Meuwissen, B. J. Hayes, and M. E. Goddard. Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, 157(4):1819–1829, Apr 2001.

K. Meyer. Restricted maximum likelihood to estimate variance components for animal models with several random effects using a derivative-free algorithm. *Genetics Selection Evolution*, 21(1):1–24, 1989.

K. Meyer. Estimating variances and covariances for multivariate animal models by restricted maximum likelihood. *Genetics Selection Evolution*, 23(1):67–83, 1991.

K. Meyer. PXxAI: algorithmics for better convergence in restricted maximum likelihood estimation. *CD-ROM Eighth World Congr. Genet. Appl. Livest. Prod., August 1318 2006, Belo Horizonte, Brasil, Communication*, 2006.

K. Meyer and M. Kirkpatrick. Restricted maximum likelihood estimation of genetic principal components and smoothed covariance matrices. *Genet. Sel. Evol.*, 37(1): 1–30, 2005.

A. Miller. *Subset Selection in Regression*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2002. ISBN 9781420035933. URL `http://books.google.de/books?id=7p59iir822sC`.

K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, Cambridge, MA, 2012. ISBN 0262018020.

I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent gaussian models. In *NIPS*, pages 1732–1740, 2010.

D. L. Newman, M. Abney, M. S. McPeek, C. Ober, and N. J. Cox. The importance of genealogy in determining genetic associations with complex traits. *Am. J. Hum. Genet.*, 69(5):1146–1148, Nov 2001.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, Aug. 2000. ISBN 0387987932.

M. Nordborg and D. Weigel. Next-generation genetics in plants. *Nature*, 456(7223): 720–723, Dec 2008.

U. Ober, J. F. Ayroles, E. A. Stone, S. Richards, and et al. Using Whole-Genome Sequence Data to Predict Quantitative Trait Phenotypes in Drosophila melanogaster. *PLoS Genetics*, 8(5):e1002685+, May 2012. ISSN 1553-7404. doi: 10.1371/journal.pgen.1002685. URL `http://dx.doi.org/10.1371/journal.pgen.1002685`.

G. Obozinski, M. J. Wainwright, and M. I. Jordan. High-dimensional support union recovery in multivariate regression. In *NIPS*, pages 1217–1224, 2008.

R. O'Hara and M. Sillanpaa. A review of Bayesian variable selection methods: What, how, and which. *Bayesian Analysis*, 4:85–118, 2009.

J. Ott, Y. Kamatani, and M. Lathrop. Family-based designs for genome-wide association studies. *Nat. Rev. Genet.*, 12(7):465–474, Jul 2011.

J. H. Park, S. Wacholder, M. H. Gail, U. Peters, K. B. Jacobs, S. J. Chanock, and N. Chatterjee. Estimation of effect size distribution from genome-wide association studies and implications for future discoveries. *Nat. Genet.*, 42(7):570–575, Jul 2010.

T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

L. Parts, O. Stegle, J. Winn, and R. Durbin. Joint genetic analysis of gene expression data with inferred cellular phenotypes. *PLoS Genet.*, 7(1):e1001276, 2011.

H. D. Patterson and R. Thompson. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554, Dec. 1971. doi: 10.1093/biomet/58.3.545. URL `http://dx.doi.org/10.1093/biomet/58.3.545`.

K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. Version 20121115.

V. T. Phan, V. W. Ding, F. Li, R. J. Chalkley, A. Burlingame, and F. McCormick. The RasGAP proteins Ira2 and neurofibromin are negatively regulated by Gpb1 in yeast and ETEA in humans. *Mol. Cell. Biol.*, 30(9):2264–2279, 2010.

A. Platt, M. Horton, Y. S. Huang, Y. Li, A. E. Anastasio, N. W. Mulyati, J. Agren, O. Bossdorf, D. Byers, K. Donohue, M. Dunning, E. B. Holub, A. Hudson, V. Le Corre, O. Loudet, F. Roux, N. Warthmann, D. Weigel, L. Rivero, R. Scholl, M. Nordborg, J. Bergelson, and J. O. Borevitz. The scale of population structure in Arabidopsis thaliana. *PLoS Genet.*, 6(2):e1000843, Feb 2010a.

A. Platt, B. Vilhjalmsson, and M. Nordborg. Conditions Under Which Genome-wide Association Studies Will be Positively Misleading. *Genetics*, 2010b.

R. Plomin, C. M. Haworth, and O. S. Davis. Common disorders are quantitative traits. *Nat. Rev. Genet.*, 10(12):872–878, Dec 2009.

A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.*, 38(8):904–909, Aug 2006.

A. L. Price, N. A. Zaitlen, D. Reich, and N. Patterson. New approaches to population stratification in genome-wide association studies. *Nat. Rev. Genet.*, 11(7):459–463, Jul 2010.

A. L. Price, A. Helgason, G. Thorleifsson, S. A. McCarroll, A. Kong, and K. Stefansson. Single-tissue and cross-tissue heritability of gene expression via identity-by-descent in related or unrelated individuals. *PLoS Genet.*, 7(2):e1001317, Feb 2011.

J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, Jun 2000.

S. M. Purcell, N. R. Wray, J. L. Stone, P. M. Visscher, M. C. O'Donovan, P. F. Sullivan, P. Sklar, S. M. Purcell, N. R. Wray, J. L. Stone, and et al. Common polygenic variation contributes to risk of schizophrenia and bipolar disorder. *Nature*, 460(7256):748–752, Aug 2009.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

N. Risch and K. Merikangas. The future of genetic studies of complex human diseases. *Science*, 273(5281):1516–1517, Sep 1996.

G. K. Robinson. That BLUP is a Good Thing: The Estimation of Random Effects. *Statistical Science*, 6(1):15–32, 1991. ISSN 08834237. doi: 10.2307/2245695. URL `http://dx.doi.org/10.2307/2245695`.

R. T. Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

G. E. Rodwell, R. Sonu, J. M. Zahn, J. Lund, J. Wilhelmy, L. Wang, W. Xiao, M. Mindrinos, E. Crane, E. Segal, B. D. Myers, J. D. Brooks, R. W. Davis, J. Higgins, A. B. Owen, and S. K. Kim. A transcriptional profile of aging in the human kidney. *PLoS Biol.*, 2(12):e427, Dec 2004.

A. Rovelet-Lecrux, D. Hannequin, G. Raux, N. Le Meur, A. Laquerriere, A. Vital, C. Dumanchin, S. Feuillette, A. Brice, M. Vercelletto, and et al. APP locus duplication causes autosomal dominant early-onset Alzheimer disease with cerebral amyloid angiopathy. *Nat. Genet.*, 38(1):24–26, Jan 2006.

R. M. Sakia. The box-cox transformation technique: A review. *Statistician*, 41(2): 169, 1992.

J. Schelldorfer and P. Bühlmann. GLMMLasso: An Algorithm for High-Dimensional Generalized Linear Mixed Models Using L1-Penalization. Sept. 2011.

J. Schelldorfer, P. Bhlmann, and S. v. De Geer. Estimation for high-dimensional linear mixed-effects models using l1-penalization. *Scand Stat Theory Appl*, 38(2): 197–214, 2011. ISSN 1467-9469. doi: 10.1111/j.1467-9469.2011.00740.x. URL `http://dx.doi.org/10.1111/j.1467-9469.2011.00740.x`.

B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759.

L. J. Scott, K. L. Mohlke, L. L. Bonnycastle, C. J. Willer, Y. Li, W. L. Duren, M. R. Erdos, H. M. Stringham, P. S. Chines, A. U. Jackson, and et al. A genome-wide association study of type 2 diabetes in Finns detects multiple susceptibility variants. *Science*, 316(5829):1341–1345, Jun 2007.

V. Segura, B. J. Vilhjalmsson, A. Platt, A. Korte, U. Seren, Q. Long, and M. Nordborg. An efficient multi-locus mixed-model approach for genome-wide association studies in structured populations. *Nat. Genet.*, 44(7):825–830, 2012.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for l1 regularized loss minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 929–936, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553493. URL `http://doi.acm.org/10.1145/1553374.1553493`.

J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA, 1994.

D. Shriner. Moving toward System Genetics through Multiple Trait Analysis in Genome-Wide Association Studies. *Front Genet*, 3:1, 2012.

S. Sivakumaran, F. Agakov, E. Theodoratou, J. G. Prendergast, L. Zgaga, T. Manolio, I. Rudan, P. McKeigue, J. F. Wilson, and H. Campbell. Abundant pleiotropy in human complex diseases and traits. *Am. J. Hum. Genet.*, 89(5):607–618, Nov 2011.

E. N. Smith and L. Kruglyak. Gene-environment interaction in yeast gene expression. *PLoS Biology*, 6(4):e83, 2008.

N. Solovieff, C. Cotsapas, P. H. Lee, S. M. Purcell, and J. W. Smoller. Pleiotropy in complex traits: challenges and strategies. *Nat. Rev. Genet.*, 14(7):483–495, Jul 2013.

S. Sonnenburg, K. Rieck, F. F. Ida, and G. Rätsch. Large scale learning with string kernels. *LARGE SCALE KERNEL MACHINES*, pages 73–103, 2007. doi: 10.1.1.84.6387. URL `http://dx.doi.org/10.1.1.84.6387`.

E. A. Stahl, D. Wegmann, G. Trynka, J. Gutierrez-Achury, R. Do, B. F. Voight, P. Kraft, R. Chen, H. J. Kallberg, F. A. Kurreeman, and et al. Bayesian inference analyses of the polygenic architecture of rheumatoid arthritis. *Nat. Genet.*, 44(5): 483–489, May 2012.

O. Stegle, C. Lippert, J. M. Mooij, N. D. Lawrence, and K. M. Borgwardt. Efficient inference in matrix-variate gaussian models with iid observation noise. In *NIPS*, pages 630–638, 2011.

P. Tao and L. An. Convex analysis approach to DC programming: Theoru,algorithms and applications. *Acta Math. Vietname*, 22(1):289–355, 1997.

R. Thompson. The estimation of variance and covariance components with an application when records are subject to culling. *Biometrics*, 29(3):pp. 527–550, 1973. ISSN 0006341X. URL `http://www.jstor.org/stable/2529174`.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

M. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5, 2009.

R. Tomioka, T. Suzuki, and M. Sugiyama. Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation. *J. Mach. Learn. Res.*, 12:1537–1586, July 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1953048.2021050`.

W. Valdar, L. Solberg, D. Gauguier, S. Burnett, P. Klenerman, W. Cookson, M. Taylor, J. Rawlins, R. Mott, and J. Flint. Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nat Genet*, 38(8):879–887, 2006a.

W. Valdar, L. C. Solberg, D. Gauguier, S. Burnett, P. Klenerman, W. O. Cookson, M. S. Taylor, J. N. Rawlins, R. Mott, and J. Flint. Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nat. Genet.*, 38(8):879–887, Aug 2006b.

S. Vattikuti, J. Guo, and C. C. Chow. Heritability and genetic correlations explained by common snps for metabolic syndrome traits. *PLoS Genet*, 8(3):e1002637, 03 2012. doi: 10.1371/journal.pgen.1002637.

P. M. Visscher, W. G. Hill, and N. R. Wray. Heritability in the genomics era–concepts and misconceptions. *Nat. Rev. Genet.*, 9(4):255–266, Apr 2008.

P. M. Visscher, J. Yang, and M. E. Goddard. A commentary on 'common SNPs explain a large proportion of the heritability for human height' by Yang et al. (2010). *Twin Res Hum Genet*, 13(6):517–524, Dec 2010.

von Mering et al. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, May 2002.

H. Wang, F. Nie, H. Huang, S. Kim, K. Nho, S. L. Risacher, A. J. Saykin, and L. Shen. Identifying quantitative trait loci via group-sparse multitask regression and feature selection: an imaging genetics study of the ADNI cohort. *Bioinformatics*, 28(2): 229–237, Jan 2012.

Warde-Farley et al. The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Res.*, 38 (Web Server issue):W214–220, Jul 2010.

L. Wasserman. *All of statistics : a concise course in statistical inference*. 2004.

A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In *ICML*, 2012.

A. G. Wilson, E. Gilboa, A. Nehorai, and J. P. Cunningham. GPatt: Fast Multidimensional Pattern Extrapolation with Gaussian Processes. *ArXiv e-prints*, oct 2013.

T. T. Wu, Y. F. Chen, T. Hastie, E. Sobel, and K. Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, Mar. 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp041. URL `http://dx.doi.org/10.1093/bioinformatics/btp041`.

J. Yang, B. Benyamin, B. P. McEvoy, S. Gordon, A. K. Henders, D. R. Nyholt, P. A. Madden, A. C. Heath, N. G. Martin, G. W. Montgomery, M. E. Goddard, and P. M. Visscher. Common SNPs explain a large proportion of the heritability for human height. *Nat. Genet.*, 42(7):565–569, Jul 2010.

J. Yang, T. Ferreira, A. P. Morris, S. E. Medland, P. A. Madden, A. C. Heath, N. G. Martin, G. W. Montgomery, M. N. Weedon, R. J. Loos, T. M. Frayling, M. I. McCarthy, J. N. Hirschhorn, M. E. Goddard, and P. M. Visscher. Conditional

and joint multiple-SNP analysis of GWAS summary statistics identifies additional variants influencing complex traits. *Nat. Genet.*, 44(4):369–375, Apr 2012a.

J. Yang, N. A. Zaitlen, M. E. Goddard, P. M. Visscher, and A. L. Price. Advantages and pitfalls in the application of mixed-model association methods. *Nat. Genet.*, 46(2):100–106, Feb 2014.

S. Yang et al. Feature grouping and selection over an undirected graph. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 922–930, New York, NY, USA, 2012b. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339675. URL `http://doi.acm.org/10.1145/2339530.2339675`.

J. Yu, W. Pressoir G. Briggs, I. Bi, M. Yamasaki, J. Doebley, M. McMullen, B. Gaut, M. Dahlia, J. Holland, S. Kresovich, and E. Buckler. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature Methods*, 38(2):203–208, 2006.

M. Yuan, M. Yuan, Y. Lin, and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68: 49–67, 2006.

N. Zaitlen and P. Kraft. Heritability in the genome-wide association era. *Hum. Genet.*, 131(10):1655–1664, Oct 2012.

H. Zhang. Maximum-likelihood estimation for multivariate spatial linear coregionalization models. *Environmetrics*, 18(2):125–139, 2007. doi: 10.1002/env.807.

Z. Zhang, E. Ersoz, C.-Q. Q. Lai, R. J. Todhunter, H. K. Tiwari, M. A. Gore, P. J. Bradbury, J. Yu, D. K. Arnett, J. M. Ordovas, and E. S. Buckler. Mixed linear model approach adapted for genome-wide association studies. *Nature genetics*, 42 (4):355–360, Apr. 2010.

K. Zhao, M. J. Aranzana, S. Kim, C. Lister, C. Shindo, C. Tang, C. Toomajian, H. Zheng, C. Dean, P. Marjoram, and M. Nordborg. An Arabidopsis example of association mapping in structured samples. *PLoS Genet.*, 3(1):e4, Jan 2007.

X. Zhou and M. Stephens. Genome-wide efficient mixed-model analysis for association studies. *Nat. Genet.*, 44(7):821–824, Jul 2012.

X. Zhou and M. Stephens. Efficient Algorithms for Multivariate Linear Mixed Models in Genome-wide Association Studies. *ArXiv e-prints*, May 2013.

X. Zhou, P. Carbonetto, and M. Stephens. Polygenic modeling with bayesian sparse linear mixed models. *PLoS Genet.*, 9(2):e1003264, 2013.

C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, Dec. 1997. ISSN 0098-3500. doi: 10.1145/279232.279236. URL `http://doi.acm.org/10.1145/279232.279236`.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

O. Zuk, E. Hechter, S. R. Sunyaev, and E. S. Lander. The mystery of missing heritability: Genetic interactions create phantom heritability. *Proc. Natl. Acad. Sci. U.S.A.*, 109(4):1193–1198, Jan 2012.