Universität zu Köln

Understanding complex traits by non-linear mixed models

Inaugural-Dissertation
zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von
Johannes Stephan
geboren in Gera

1. Gutachter: Prof. Dr. Andreas Beyer
2. Gutachter: Prof. Dr. Achim Tresch
3. Gutachter: Dr. Oliver Stegle

Tag der mündlichen Prüfung: 8. Oktober 2014

Meiner Familie gewidmet.

Acknowledgements

First of all, I like to thank my mentor Andreas Beyer, whose group I joined as a PhD student at the BIOTEC in Dresden. During the past years Andreas has not only been a great scientific advisor, but also proven to be a keen manager that always keep my focus on scientific tasks. Under his supervision I enjoyed the freedom to exploit the areas of science that I particularly liked!

Oliver Stegle, who I met as undergraduate student at the Max Planck institute of Tübingen, has significantly contributed to this work. He is father of many ideas that are exploited here. Beyond sharing his exceptional scientific expertise he provided plenty of practical advice. Thanks a lot Oli! I really enjoyed working with you for the past two years!

I like to thank Achim Tresch who joined our labmeetings with his group. I will miss the (scientific) discussions we had during our weekly lunch breaks. And thanks a lot for giving me a hand with mathematical problems!

Group members a invaluable for internal reviewing, scientific - and most importantly - social support. So are you! Thanks to Karl Köchert for open minded discussions while having record breaking restaurant bills. Thanks to Julia Harnisch and Susanne Reinhardt for being great in- and out of office mates. All the best to you Sinan Saraç and thanks for setting me up at the beginning. Thanks to all members of the fantastic room: Marit Ackermann, Kalaimathy Singaravelu, Eleni Christodoulou, Mathieu Clément-Ziza, Weronika Sikora-Wohlfeld, Betty Friedrich and Michael Kuhn. And finally, thanks to my lab members from the newly established Cologne group: Li Li, Cédric Debès, Jan Großbach, Mikael Kaminsky, Kay Heitplatz and Marius Garmhausen.

Being at work while feeling among friends never happened to me before. Thanks a lot to my host group at the EBI: Amelie Baud, Yuanhua Huang, Sung Hee Park and Paolo Casale. I spent a wonderful time with you in Cambridge.

Furthermore, I like to acknowledge people that guided and inspired me during my time as a diploma student at the University of Tübingen. I would like to thank my professors for their engagement in turning me into a scientist. In particular I would like to mention professor Hauck, as he is the greatest lecturer I know! My special thanks go to Jacquelyn Shelton for getting me in touch with Oliver Stegle, infecting me with “machine learning” and providing invaluable support during the application for my

PhD. Thanks a lot Jackie!

I like to thank Aylar Tafazzoli for her constant support and care during the past months when I was primarily occupied writing this thesis.

Finally, I like to thank my parents, grandparents, grandmother and my little brother for being with me throughout my life. It was my family's constant support that kept me on track and their warm and inspiring environment which enabled me to pursue a scientific career till now.

Abstract

Population structure and other nuisance factors represent a major challenge for the analysis of genomic data. Recent advances in statistical genetics have lead to a new generation of methods for quantitative trait mapping that also account for spurious correlation as caused by population structure. In particular, linear mixed models (LMMs) gained considerable attention as they enable easy black box-like control for population structure in a wide range of genetic designs and analysis settings.

The aim of this work is to transfer the advantages of LMMs into a random bagging framework in order to simultaneously address a second pressing challenge: the recovery of complex non-linear genetic effects. Existing methods that allow for identifying such relationships like epistasis typically do not provide any robust and interpretable means to control for population structure and other confounding effects.

The method we present here is based on random forests, a bagged variant of the well established decision trees. We show that the proposed method greatly improves over existing methods not only in identifying causal genetic markers but also in the prediction of held out phenotypic data.

Zusammenfassung

Populationsstrukturen sowie andere unerwünschte Faktoren erschweren häufig die Analyse genomischer Daten. Aufgrund von Fortschritten in der statistischen Genetik sind neuere Methoden in der Lage, unerwünschte Korrelationen, die z.B. durch Populationsstrukturen entstehen, zu korrigieren. Insbesondere haben lineare *Mixed Models* stark an Popularität gewonnen. Durch ihre anwenderfreundliche Kontrolle der Populationsstruktur sind sie für viele genetische Strukturen und in vielen Studiendesigns anwendbar.

Ziel dieser Arbeit ist es, die Vorteile der linearen *Mixed Models* mit denen eines *Random Bagging* Verfahrens zu vereinen, um das Finden komplexer genetischer Effekte, zu erleichtern. Bestehende Methoden, die solche Signale wie *Epistasis* erkennen, sind bisher nicht in der Lage, Populationsstrukturen und andere Störfaktoren zu berücksichtigen.

Die hier vorgestellte Methode ist eine Erweiterung des *Random Forests*, eines *Random Bagging*-Verfahrens welches auf Entscheidungsbäumen ba-

siert. Wie auch bei linearen *Mixed Models* korrigiert es Störfaktoren durch einen *Random Effect*. Mit Hilfe von simulierten und realen Daten zeigen wir, dass diese neue Methode nicht nur mehr kausale genetische Marker gegenüber bestehenden Ansätzen findet, sondern auch die Vorhersage ungesehener Phenotypen verbessert.

Contents

Motivation

During the past decades biomedical research has received increasing attention as it significantly furthered our understanding of complex medical conditions such as type-II diabetes, Alzheimers- or Crohns disease. In one of its main directions, researchers aim to unveil inherited and environmental contributions to a specific phenotype, as for instance, the state of a disease they are interested in. With steadily increasing amount of genomic data, the application computational tools to guide researchers becomes more and more important. Among these, quantitative trait locus (QTL) mapping methods assess the strength of a link between a genotypic region to quantitative phenotypic condition (trait).

Whereas most QTL mapping methods model phenotypes as a simple linear function of the genotype, it is assumed that for many of the complex diseases multiple genetic factors contribute in a non-linear fashion. In addition, individuals in a sample can be related by means of population structure. Not correcting for such sources of confounding effects leads to an increase in false positive hypotheses and therefore more recent work focuses on correcting for population effects when the underlying genotype to phenotype relationship is linear [28, 30, 34, 50, 66]

On the other hand, numerous alternative approaches have been developed in order to detect non-linear effects like gene-gene epistasis. For example, linear models with interaction terms can be fit using a greedy algorithms [14, 44] or by sampling techniques, e.g. [12]. Also, random bagging techniques [9] have gained considerable attention. In particular, random forests [10] have been shown to accurately capture epistatic effects (e.g. [41, 43, 49]).

All these approaches - including random forests - assume that correlations between genotype and phenotype are genuine and, unlike extensions of linear models, do not explicitly correct for population structure or other confounding effects. Thus, there is a lack of methods that can perform both tasks: correcting for population structure while accounting for epistasis.

Outline of this thesis

Following **Part I** of this thesis, the reader will learn about linear QTL mapping methods and their extensions to correct for population structure as well as decision tree based approaches.

These methodological concepts are required for **Part II**, where we show how a decision tree based approach can be extended to a simple yet efficient correction for population structure maintaining its ability to map multivariate non-linear associations.

We compare our new approach, termed mixed random forest, to alternative state of the art methods using simulations as well as data obtained from a large scale study in mice [62].

Part I.

Introduction to genomic association mapping

1

Univariate linear models

Linear models are established tools for QTL mapping. The simplest and probably most commonly used is the so called *univariate linear model* where a single independent variable like the state of a gene is used to explain a continuous outcome. More elaborate variants are capable of including several features in a linear-additive fashion. We review the univariate model in the following before we discuss several ways to obtain multivariate linear models in Chapter 2.

Assume we are given a continuous phenotypic trait measured for N individuals that is stored within a N-dimensional vector $\mathbf{y}$. Our goal is to explain $\mathbf{y}$ as a *linear function* of genomic information. If we allow for additive noise $\boldsymbol{\psi}$ on our measurements we can write our model as follows

$$\mathbf{y} = \boldsymbol{\beta}_0 + \mathbf{x}\beta + \boldsymbol{\psi}. \tag{1.1}$$

Here, $\mathbf{x}$ is a real- or integer-valued vector of size N that encodes the genetic state for each of individual. Remaining parameters of our model in Equation (1.1) are the weight or slope of our linear function β and the intercept with the y-axis $\boldsymbol{\beta}_0$. We shall not concern ourselves with the intercept and thus set $\boldsymbol{\beta}_0 = 0$.

For now, β remains the only parameter we want to fit. We optimize this linear model by minimizing the mean squared error between phenotype $\mathbf{y}$ and its linear reconstruction $\mathbf{x}\beta$ w.r.t. β, i.e.

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{N} \|\mathbf{y} - \mathbf{x}\beta\|^2 = \arg\min_{\beta} \frac{2}{N} \underbrace{\frac{1}{2} \sum_{i=1}^{N} (y_i - x_i\beta)^2}_{\mathrm{E}(\beta)}. \tag{1.2}$$

In Section 1.3, we will learn which assumptions are made about the nature of the error $\boldsymbol{\psi}$ when using this so called *least squares* optimization.

The solution of Equation (1.2) can be found analytically by setting the

derivative of the *sum of squares error function*[1] $\mathrm{E}(\beta)$ w.r.t. β to zero

$$\hat{\beta} = \left(\sum_{i=1}^{N} x_i^2 \right)^{-1} \left(\sum_{i=1}^{N} y_i x_i \right). \tag{1.3}$$

We can now evaluate the goodness of the learned model by plugging the optimal solution $\hat{\beta}$ back into Equation (1.2)

$$\mathrm{E}(\hat{\beta}) = \frac{1}{2} \left\| \mathbf{y} - \mathbf{x}\hat{\beta} \right\|^2, \tag{1.4}$$

where a low error indicates a good fit. An algorithm for univariate linear association mapping can be established as follows:

1. For each of the given genetic features, optimize the linear model in Equation (1.3) and
2. evaluate the error function (Equation (1.4))
3. Return the error for each feature

The lower the associated error, the stronger we consider the association between genetic feature and phenotype.

Feature scoring using the error function. Importance measures such as the sum of squares error above only tell about the relative importance of genetic features in a given analysis. This becomes an issue if we intend to compare our results to those of other studies. To obtain more comparable scores, we can evaluate the ratio of the sum of squares errors before and after fitting a model, i.e.

$$\Delta\, \mathrm{E}(\hat{\beta}) = \frac{\mathrm{E}(0)}{\mathrm{E}(\hat{\beta})} \tag{1.5}$$

where

$$\mathrm{E}(0) = \frac{1}{2} \sum_{i=1}^{N} (y_i - \bar{y})^2 \tag{1.6}$$

and $\bar{y}$ is the mean of our phenotype across all individuals. This ratio of errors turns out to be proportional to the *logarithm of odds* (LOD) that is introduced at the end of this chapter.

[1] Also referred to as *residual sum of squares*

1.1. Including covariates

Phenotypes are usually driven by other non-genetic factors like the time at which a measurement was taken, environmental variables such as temperature and humidity or even the experimenter herself/himself. We refer to these factors as confounders as there are not of interest to the study. Careful modelling of such variables is essential to avoid false positive discoveries. To illustrate this, suppose there exists a genetic feature which happens to be correlated to a confounder that explains a large fraction of phenotypic variance. If this confounder is not included into our model, the correlated genotypic feature will take its place in explaining phenotypic variance and therefore receive a good score. It is, on the other hand, very unlikely that this particular genetic feature is relevant for the observed trait. So what we consider to be genetic driver of our phenotype is probably a false positive.

We can deal with covariates the same way as for genotypic information by including them as linear additive factors into our model, i.e.

$$\mathbf{y} = \mathbf{c}_1\beta_{c_1} + \mathbf{c}_2\beta_{c_2} + \cdots + \mathbf{c}_k\beta_{c_k} + \mathbf{x}\beta + \boldsymbol{\psi}. \tag{1.7}$$

This turns our univariate- into a multivariate linear model. In the context of (genomic) association mapping, however, it is still considered to be a univariate model as only the importance of a single (genetic) feature $\mathbf{x}$ is of interest.

To evaluate the improvement in residual error that can be attributed to $\mathbf{x}$ we need to optimize the full model in Equation (1.7) w.r.t. all weights and compute the corresponding error. We do the analogue for the baseline model, which does not include the genomic feature to be tested for association.

Starting with the error functions

$$\mathrm{E}_0(\boldsymbol{\beta}_c) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{C}\boldsymbol{\beta}_c\right\|^2 \qquad \text{baseline model} \tag{1.8}$$

$$\mathrm{E}(\boldsymbol{\beta}_c, \beta) = \frac{1}{2}\left\|\mathbf{y} - (\mathbf{C}\boldsymbol{\beta}_c + \mathbf{x}\beta)\right\|^2 \qquad \text{alternative model} \tag{1.9}$$

where we introduced the matrix containing the covariates as columns $\mathbf{C} = \left[\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k\right]$ and the weight vector $\boldsymbol{\beta}_c = \left[\beta_{c_1}, \beta_{c_2}, \cdots, \beta_{c_k}\right]^t$. After taking the derivative of Equation (1.8) w.r.t. $\boldsymbol{\beta}_c$, setting it to zero

and manipulating the expressions algebraically, we obtain for the baseline model

$$\hat{\boldsymbol{\beta}}_{\mathbf{c}} = \left(\mathbf{C}^T\mathbf{C}\right)^{-1}\mathbf{C}^T\mathbf{y}. \tag{1.10}$$

We run the analogue optimization for our alternative model $\mathrm{E}(\boldsymbol{\beta}_{\mathbf{c}}, \beta)$ and report

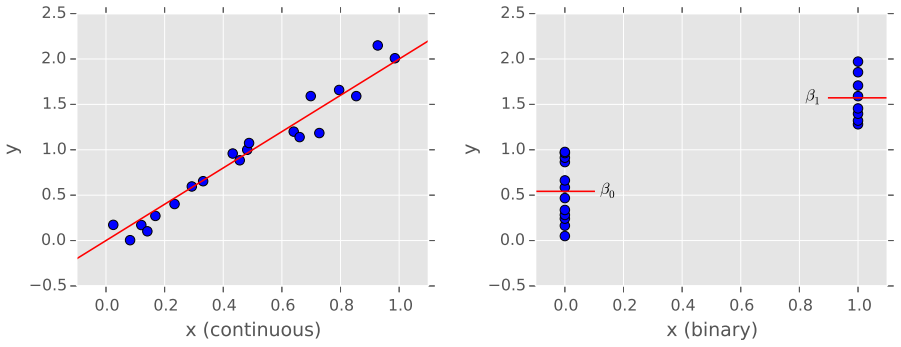
$$\Delta\,\mathrm{E} = \frac{\mathrm{E}_0(\hat{\boldsymbol{\beta}}_{\mathbf{c}})}{\mathrm{E}(\hat{\boldsymbol{\beta}}_{\mathbf{c}}, \hat{\beta})} \tag{1.11}$$

as score for the association strength of the genetic feature $\mathbf{x}$. Note, that the minimized squared error for the baseline model $\mathrm{E}_0(\hat{\boldsymbol{\beta}}_{\mathbf{c}})$ only needs to be computed once.

Fitting the intercept. We can equally regard the intercept $\boldsymbol{\beta}_0$ of our model from the beginning (Equation (1.1)) as a $N \times 1$ vector multiplied by the (scalar) weight β_0

$$\mathbf{y} = \mathbf{1}\beta_0 + \mathbf{x}\beta + \boldsymbol{\psi}, \tag{1.12}$$

which allows us to run optimization as before (Equations (1.10) and (1.11)) where the one-valued vector enters as additional covariate.



(a) **(b)**

Figure 1.1.: Univariate linear models. (a) 20 individuals are simulated by first sampling uniformly between 0 and 1 in predictor-space x. Corresponding responses in y are then computed as a linear function of x plus a small amount of independent Gaussian noise. (b) the same sample where the predictor values have been "binarized" such that all $x_i \leq 0.5$ are set to 0 and the remaining $x_i > 0.5$ to 1. Red lines indicate the means β_0 and β_1 as fitted for the two resulting groups of samples.

1.2. Binary predictors

Genetic variants of some stretch of DNA (called 'alleles') are often distinguished by measuring single nucleotide polymorphisms (SNPs) in such a region. Assume that Adenin is the most frequent base at a given position in our population and Cytosin being the less common alternative. We say that individuals that contain Adenin are carrier of the *major allele*, whereas the rest of our population has the *minor allele* (Cytosin). A common way of encoding is to use "0" to refer to major- and "1" to minor alleles.

For continuous $\mathbf{x}$ a linear model can be visualized by a regression line indicating slope and intercept that have been fitted (Figure 1.1(a)). The special case of binary predictors, enables us to take an alternative point of view where the model computes means within two distinct groups: the individuals that contain $x_i = 0$ and the ones that have $x_i = 1$, respectively. As shown in Figure 1.1(b), these will turn out to be the intercept β_0 and the weight β_1 from our linear model in Equation (1.12).

Note, that most species have more than a single copy of a chromosome in each cell. Since each copy of a chromosome can harbor a different allele of every marker, in general we have more than two states per marker. For example, in humans (with two copies per chromosome) we have three possible states: both copies are major, both copies are minor, or one copy is the major and the other the minor allele. However, for task of fitting regression trees, we can still take advantage of this binary perspective by converting a single predictor (that has more than two values) into an equivalent *set* of binary predictors (see Chapter 4).

1.3. Gaussian interpretation

Now we turn to a probabilistic interpretation of what we referred to as *least squares* linear model so far. probability distributions. We start by constraining our measurement noise from Equation (1.1) $\boldsymbol{\psi}$ to be Gaussian, specifically we have identical and independently (i.i.d.) distributed Gaussian noise with zero mean

$$\boldsymbol{\psi} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{I}\right). \tag{1.13}$$

Including the linear relationship as mean into the noise model, we can regard the phenotype as sample from a Gaussian distribution

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{C}\boldsymbol{\beta}_c + \mathbf{x}\beta, \sigma^2\mathbf{I}\right). \tag{1.14}$$

Taking the log transformed

$$\log(p(\mathbf{y}|\mathbf{x}, \mathbf{C}, \boldsymbol{\beta}_c, \beta, \sigma^2) = \frac{N}{2}\log\left(\frac{1}{2\pi\sigma^2}\right) - \frac{1}{\sigma^2}\underbrace{\frac{1}{2}\sum_{i=0}^{N}(y_i - (\mathbf{c}_i\boldsymbol{\beta}_c + x_i\beta))^2}_{\mathrm{E}(\beta_c,\beta)}. \tag{1.15}$$

and our goal becomes to maximize this so-called *log likelihood* w.r.t. the model parameters $\boldsymbol{\beta}_\mathbf{c}$, β and σ^2. This leads to the same optimal solution as by working on the Gaussian directly but the log transformed is preferred for reasons of numerical stability. Furthermore, we notice that the weights $\boldsymbol{\beta}_\mathbf{c}$ and β only depend on the log likelihood only through the negative of the error function $-\mathrm{E}(\boldsymbol{\beta}_\mathbf{c}, \beta)$. In other words, maximizing the log likelihood of a linear model is equivalent to minimizing the least squares in the case of i.i.d. Gaussian noise. We therefore reuse Equation (1.10) to find $\hat{\boldsymbol{\beta}}_\mathbf{c}$ and $\hat{\beta}$. Putting these optimal weights back into Equation (1.15) and setting the derivative w.r.t. σ^2 to zero we obtain

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=0}^{N}(y_i - (\mathbf{c}_i\hat{\boldsymbol{\beta}}_c + x_i\hat{\beta}))^2. \tag{1.16}$$

Now we can use our optimal parameters to reevaluate the log likelihood

$$\log(p(\mathbf{y}|\mathbf{x}, \mathbf{C}, \hat{\boldsymbol{\beta}}_\mathbf{c}, \hat{\beta}, \hat{\sigma}^2) = N\log\left(\frac{1}{2\pi\hat{\sigma}^2}\right) - \frac{N}{2}. \tag{1.17}$$

As before we also need to maximize the log likelihood of our baseline model $\log(p(\mathbf{y}|\mathbf{C}, \hat{\boldsymbol{\beta}}_c, \hat{\sigma}_0^2)$.

Logarithm of odds. The logarithm of odds (LOD) is defined as

$$\mathrm{LOD} = \log\frac{p(\mathbf{y}|\mathbf{x}, \mathbf{C}, \hat{\beta}, \hat{\boldsymbol{\beta}}_\mathbf{c}, \hat{\sigma}^2)}{p(\mathbf{y}|\mathbf{C}, \hat{\boldsymbol{\beta}}_\mathbf{c}, \hat{\sigma}_0^2)} = \frac{N}{2}\log(\hat{\sigma}_0^2) - \frac{N}{2}\log(\hat{\sigma}^2) \tag{1.18}$$

and can equivalently be computed as difference of the optimized likelihoods for baseline- and alternative model. Associations that have a LOD greater

than 3 are commonly considered as statistically significant, but in general it is hard to define such a meaningful threshold [3].

P-Values. As alternative to the LOD, we can use the fitted variances of baseline- and alternative model in order to compute the *F-Score*. This measure can subsequently be used to evaluate the significance of a genetic association by means of a P-Value, see e.g. [3]. P-Values should be handled with care as this approach is very sensitive towards non-normality on the data [7].

2

Multivariate linear models

Here we give a brief review on multivariate linear models. Instead of considering a single genetic feature for a given phenotype, multivariate linear models account for multiple effects in a linear-additive fashion, i.e.

$$\mathbf{y} = \mathbf{C}\boldsymbol{\beta}_c + \mathbf{x}_1\beta_1 + \mathbf{x}_2\beta_2 + \cdots + \mathbf{x}_M\beta_M + \boldsymbol{\psi}. \tag{2.1}$$

Once we selected all features to be included, optimization is analog to the linear model with covariates (see Section 1.1).

In most association studies we expect only a small fraction of features to be relevant for the phenotype. A model containing all genotypic information might explain a great proportion of variance, but it is useless if we are to select the subset of features driving phenotypic changes. Therefore, most (if not all) multivariate linear models used today, implement a mechanism to control for the number of features included.

In the following, we present three popular variants of multivariate linear models, a greedy approach called *forward selection*, its counterpart *backward elimination* and the *LASSO*.

2.1. Forward selection and backward elimination

In forward selection we start testing each feature in a univariate linear model for association (using some statistic like the LOD). The feature obtaining the highest score is included as covariate into the (updated) baseline model. This procedure of testing is repeated for the remaining features. Again, the best feature will be included into the baseline model and so on. . . . This way, covariates are added to the model until it cannot be improved by a predefined threshold.

Backward elimination, on the other hand, starts with the model including all covariates. The feature that leads to the least reduction in a predefined score is removed. Covariates are being eliminated as long as reduction in score stays **below** a predefined threshold.

For both of these *stepwise regression methods* (see e.g. [24]) the final model contains the set of genetic features considered important.

2.2. The LASSO

The **L**east **A**bsolute **S**hrinkage and **S**election **O**perator (LASSO) [60] is a linear model that, depending on the adjustment of a so called shrinkage parameter, includes a limited number of features.

We derive LASSO extending the linear model's optimization function by a regularizer that sums over the absolute values of the weights

$$\mathrm{E}(\lambda, \boldsymbol{\beta}) = \frac{1}{N} \underbrace{\frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2}_{\mathrm{E}(\boldsymbol{\beta})} + \underbrace{\lambda \left(\sum_{j=1}^{M} |\beta_j| \right)}_{\text{regularizer}}. \tag{2.2}$$

To minimize the total error $\mathrm{E}(\lambda, \boldsymbol{\beta})$ our aim is to keep the regularizers contribution low. Depending on the choice of λ we encourage weights β_i that are zero. So, rather than explicitly including or excluding features like in forward selection or backward elimination, LASSO controls the number of active features through their weights.

There is no analytical solution to find the optimum for our objective (Equation (2.2)). Nevertheless, we are dealing with a *convex optimization problem* and efficient numerical methods are available to obtain good approximations. Most LASSO solvers use gradient descent which makes a local quadratic approximation to the optimization function [15].

Bayesian interpretation. We conclude this review of LASSO giving a Bayesian interpretation. By taking the exponent of its negative objective (Equation (2.2))

$$\underbrace{p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X})}_{\text{posterior}} \propto \underbrace{\mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{X}\boldsymbol{\beta},\, \sigma_v^2 \mathbf{I}\right)}_{\text{likelihood}} \underbrace{\prod_{j=1}^{M} \exp\left(-\frac{1}{2}\lambda\,|\beta_j|\right)}_{\text{prior}}. \tag{2.3}$$

we find that the regularizer can be seen as Laplace prior over weights β_j whereas our error function $\mathrm{E}(\boldsymbol{\beta})$ turned into a Gaussian likelihood. As

the marginal $p(\mathbf{X}, \mathbf{y})$ is constant, the product of prior and likelihood is proportional to the posterior over the weights. In other words, minimizing LASSO's objective in Equation (2.2) is equivalent to maximizing the posterior over $\boldsymbol{\beta}$ in this probabilistic view.

2.3. Bagging for feature selection in multivariate linear models

A problem common to all these methods is that we end up with a set of active predictors while having little information about their relative importance. Note, that considering the LOD or similar statistical measures when including features using forward selection can be misleading. Imagine that the baseline model already includes a covariate which is correlated to a feature we intend to test/include. The feature under consideration is prone to receive a relatively low score, as a significant proportion of phenotypic variance is already accounted for by the other correlated covariate. In the worst case, we may not consider this feature after all. Using the absolute of the weights β_j as importance measure when fitted in a LASSO approach (see Equation (2.2)) is problematic for similar reasons.

There are several methods that address this problem. A conceptually simple way to obtain more robust feature scores uses bagging [9]. That is, we randomly sample N times with replacement from the whole set of individuals. We obtain what is called a *bootstrap sample* which is used to fit the multivariate linear model of choice. We repeat bootstrapping and fitting several times and record the features selected at each run. The fraction of bootstraps in which a given feature was included in the model is used as importance measure.

Bagging does not come without issues. We need to fit our model several times which can easily exhaust computational resources. In addition, the interpretability of a single linear model is sacrificed for a collection of noisy variants.

For LASSO feature selection, bagging is encouraged by the work of Meinshausen and Bühlmann [40] in which they show that resulting scores are admissible w.r.t. false discovery rates. Bagging has also been successfully applied in combination with forward selection to map complex traits of heterogeneous mouse populations [62].

3

Linear mixed models

Linear mixed models are an extension to linear models containing additional summands $\mathbf{u}_1, \mathbf{u}_2, \dots \mathbf{u}_K$, called random effects

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + z_1\mathbf{u}_1 + z_2\mathbf{u}_2 + \cdots z_k\mathbf{u}_K + \boldsymbol{\psi}, \tag{3.1}$$

were $z_1, z_2, \dots z_K$ weight their relative contributions. Random effects can be regarded as random variables, each following some probability distribution. Here, we will focus on the class of mixed models containing a single Gaussian random effect $\mathbf{u}$, i.e.

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u} + \boldsymbol{\psi} \tag{3.2}$$

where

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}).$$

Omitting the weight z is not constraining the model as we can simply include z as a factor into the covariance matrix $\boldsymbol{\Sigma}$. To keep naming short and simple, we refer to this variant as linear mixed model in the following.

Throughout this work, we use random effects to model those parts of sample variance that are not in the focus of our study. Including known covariates is one way to account for confounding variance (see Chapter 1.1). However, with large heterogeneous populations, the number of additional factors needed can increase rapidly. Unregularized linear models are then prone to overfit. Random effect modelling, on the other hand, provides a robust way to deal with such complex covariate structures.

3.1. Bayesian motivation of the random effect

We start considering a linear model that just includes our matrix of covariates $\mathbf{C}$

$$\mathbf{y} = \mathbf{C}\boldsymbol{\beta}_c + \boldsymbol{\psi}. \tag{3.3}$$

Setting the so called *fixed effect* $\mathbf{X}\boldsymbol{\beta}$ to zero is not compromising the important insights of this Bayesian view, allowing us to update the following derivation for non-zero fixed effects later.

We further assume, that $\mathbf{C}$ is a $N \times M$ matrix where M is in the range of N. As mentioned before, joint inference of all covariates using ordinary least squares is prone to result in overfitting. In case of $M < N$ we are guaranteed to run in numerical problems while computing the matrix inversion in Equation (1.10). A common way to address these issues is to introduce a prior distribution over the weights $\boldsymbol{\beta}_c$. Here we use an i.i.d. Gaussian

$$\boldsymbol{\beta}_c \sim \mathcal{N}\left(\mathbf{0},\, \sigma_g^2 \mathbf{I}\right). \tag{3.4}$$

Multiplying our model for $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}_c)$ with the prior, we arrive at the joint distribution of $\boldsymbol{\beta}_c$ and the observations $\mathbf{y}$

$$p(\mathbf{y}, \boldsymbol{\beta}_c) = p(\mathbf{y}|\boldsymbol{\beta}_c)p(\boldsymbol{\beta}_c). \tag{3.5}$$

Direct maximization of this joint distribution w.r.t. $\boldsymbol{\beta}_c$ would lead to an instance of the so called *ridge regression*, a method developed in order to more robustly solve ill posed least-squares problems (see for instance [61]).

When modelling nuisance factors, we are not interested in reporting values for the weights $\boldsymbol{\beta}_c$ (or other types of importance measures). This allows us to take Bayesian modelling one step further marginalizing over $\boldsymbol{\beta}_c$. The resulting model accounts for the whole range of possible configurations of $\boldsymbol{\beta}_c$ weighted by their prior distribution. Following the usual steps of marginalization would require integrating the joint distribution w.r.t. $\boldsymbol{\beta}_c$, i.e.

$$p(\mathbf{y}) = \int p(\mathbf{y}|\boldsymbol{\beta}_c)p(\boldsymbol{\beta}_c)d\boldsymbol{\beta}_c. \tag{3.6}$$

Our particular case allows a simpler derivation: As both, $p(\mathbf{y}|\boldsymbol{\beta}_c)$ and $p(\boldsymbol{\beta}_c)$ are Gaussian, it follows that the joint distribution $p(\mathbf{y}, \boldsymbol{\beta}_c)$ is Gaussian as well. Joining the individual quadratic forms

$$\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}_c\right)^{\mathrm{T}} \mathbf{I}\sigma_v^{-2} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}_c\right) \text{ and } \boldsymbol{\beta}^{\mathrm{T}}\mathbf{I}\sigma_g^{-2}\boldsymbol{\beta}^{\mathrm{T}}$$

in the exponents to of $p(\mathbf{y}|\boldsymbol{\beta}_c)$ and $p(\boldsymbol{\beta}_c)$ into an equivalent quadratic form over the concatenated vector $[\mathbf{y}, \boldsymbol{\beta}_c]^{\mathrm{T}}$ we note that the matrix in this quadratic form must be the inverse covariance of the joint distribution.

We have

$$\begin{aligned} p(\mathbf{y}, \boldsymbol{\beta}_c) &= \mathcal{N}\left(\mathbf{y} \,|\, \mathbf{C}\boldsymbol{\beta}_c,\, \sigma_v \mathbf{I}\right) \mathcal{N}\left(\boldsymbol{\beta}_c \,\middle|\, \mathbf{0},\, \sigma_g^2 \mathbf{I}\right) \\ &= \mathcal{N}\left(\begin{bmatrix} \mathbf{y} \\ \boldsymbol{\beta}_c \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{I}\sigma_v^{-2} & \mathbf{C}\sigma_v^{-2} \\ \mathbf{C}^{\mathrm{T}}\sigma_v^{-2} & \mathbf{C}^{\mathrm{T}}\mathbf{C}\sigma_v^{-2} + \mathbf{I}\sigma_g^{-2} \end{bmatrix}^{-1} \right). \end{aligned} \tag{3.7}$$

Applying the rules for Gaussian marginalization (Equation (B.4)) and inversion of partitioned matrices (Equation (B.6)), we find the marginal distribution of $\mathbf{y}$

$$p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{0},\, \sigma_g^2 \mathbf{C}\mathbf{C}^{\mathrm{T}} + \sigma_v^2 \mathbf{I}\right). \tag{3.8}$$

Individual samples are now related through the covariance matrix $\sigma_g^2 \mathbf{C}\mathbf{C}^{\mathrm{T}} + \sigma_v^2 \mathbf{I}$. In contrast to our model in Equation (3.5) this marginalized version contains $M - 1$ parameters less.

Let's get back to our full linear model including a linear fixed effect

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{C}\boldsymbol{\beta}_c + \boldsymbol{\psi}. \tag{3.9}$$

Applying the same Gaussian prior and doing the analogue marginalization over $\boldsymbol{\beta}_c$, we note that the fixed effect $\mathbf{X}\boldsymbol{\beta}$ only enters the mean and not the covariance of the joint distribution Equation (3.7). We find for the updated marginal distribution

$$p(\mathbf{y}|\boldsymbol{\beta}) = \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{X}\boldsymbol{\beta},\, \sigma_g^2 \boldsymbol{\Sigma} + \sigma_v^2 \mathbf{I}\right), \tag{3.10}$$

where we defined $\boldsymbol{\Sigma} \coloneqq \mathbf{C}\mathbf{C}^{\mathrm{T}}$. Note, that the fixed effect can be replaced by any function $f(\mathbf{X})$ which is independent of the confounders contained in $\mathbf{C}$.

3.2. Efficient inference in linear mixed models

Linear mixed models can be optimized analytically, but in contrast to the vanilla linear model from before, computational complexity increases drastically. The main reason lies in the inversion of the covariance matrix $\sigma_g^2 \boldsymbol{\Sigma} + \sigma_v \mathbf{I}$ that is required to evaluate the derivative of our objective function (log of Equation (3.10)) w.r.t. $\boldsymbol{\beta}$. As a result, runtime scales cubic in the number of samples, and if we intend to test M features for

association, the total computational cost multiplies up to $\mathcal{O}(MN^3)$. So using the naïve analytical approach will render large scale studies with millions of SNP features infeasible.

More efficient methods joined the scene recently. These reduce runtime by clever algebraic tricks [29, 34, 35, 66] and making weak assumptions about confounding effects [28, 34, 35, 65, 66]. Here, we present in parts the work of Lippert and others [34]. Their proposed FaST LMM utilizes a *singular value decomposition* (SVD) of $\mathbf{\Sigma}$ in order to reduce the total runtime complexity to $\mathcal{O}(n^3 + n^2 m)$[1]. The same trick is later applied to the method that is in the focus of this thesis.

We start with the log likelihood of the linear mixed model (Equation (3.10)), this time including all parameters of the conditioning set

$$\mathrm{LL}(\boldsymbol{\beta}, \sigma_v^2, \sigma_g^2) = \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \sigma_g^2, \sigma_v^2) = \log \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{X}\boldsymbol{\beta},\, \sigma_g^2\mathbf{\Sigma} + \sigma_v^2\mathbf{I}\right). \tag{3.11}$$

Our first step is to substitute δ for $\frac{\sigma_v}{\sigma_g}$ to rewrite Equation (3.11) as

$$\mathrm{LL}(\boldsymbol{\beta}, \sigma_g^2, \delta) = \log \frac{1}{Z} - \frac{1}{2} \log \left(\frac{1}{\sigma_g^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} (\mathbf{\Sigma} + \delta\mathbf{I})^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right). \tag{3.12}$$

Here, Z denotes the Gaussian's normalizing constant. Second, we replace $\mathbf{\Sigma}$ by a singular value decomposition $\mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}$ is a orthonormal- and $\mathbf{S}$ is a diagonal matrix. Utilizing that $\mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}$, we rearrange Equation (3.12) to

$$\mathrm{LL}(\boldsymbol{\beta}, \sigma_v^2, \sigma_g^2) = \log \tfrac{1}{Z} - \tfrac{1}{2} \log \left(\tfrac{1}{\sigma_g^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} (\mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{T}} + \delta\mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right). \tag{3.13}$$

After factoring out $\mathbf{U}$ and $\mathbf{U}^{\mathrm{T}}$ in the next step, we manipulate the inverse covariance $\left(\mathbf{U}(\mathbf{S} + \delta\mathbf{I})\mathbf{U}^{\mathrm{T}}\right)^{-1}$ algebraically to obtain $\mathbf{U}(\mathbf{S}+\delta\mathbf{I})^{-1}\mathbf{U}^{\mathrm{T}}$. Plugging this back into Equation (3.13) we have

$$\mathrm{LL}(\boldsymbol{\beta}, \sigma_v^2, \sigma_g^2) = \log \tfrac{1}{Z} + \tfrac{1}{2} \log \left(\tfrac{1}{\sigma_g^2} (\mathbf{U}^{\mathrm{T}}\mathbf{y} - \mathbf{U}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} (\mathbf{S} + \delta\mathbf{I})^{-1} (\mathbf{U}^{\mathrm{T}}\mathbf{y} - \mathbf{U}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta}) \right). \tag{3.14}$$

A closer look on this reformulated log-likelihood reveals a quadratic form in $\mathbf{U}^{\mathrm{T}}\mathbf{y}$. In other words, we can alternatively express Equation (3.14) as

[1] In their work they further show that, when using a low rank approximation of $\mathbf{\Sigma}$, runtime can even be further reduced.

Gaussian under orthonormal projection by $\mathbf{U}^{\mathrm{T}}$

$$\begin{aligned}\mathrm{LL}(\boldsymbol{\beta}, \sigma_g^2, \delta) &= \log \mathcal{N}\left(\mathbf{U}^T\mathbf{y} \,\middle|\, \mathbf{U}^T\mathbf{X}\boldsymbol{\beta},\, \sigma_g^2\left(\mathbf{S}+\delta\mathbf{I}\right)\right) \\ &= \log \prod_{i=1}^{N} \mathcal{N}\left(\left[\mathbf{U}^{\mathrm{T}}\mathbf{y}\right]_i \,\middle|\, \left[\mathbf{U}^{\mathrm{T}}\mathbf{X}\right]_{i:}\boldsymbol{\beta},\, \sigma_g^2\left(\left(\mathbf{S}_{i,i}+\delta\right)\right)\right).\end{aligned} \tag{3.15}$$

The covariance $\sigma_g^2(\mathbf{S}+\delta\mathbf{I})$ is a diagonal matrix which allows us to factorize our likelihood model. Also note, that the normalization of a Gaussian distribution is invariant w.r.t. orthonormal projection such that Z correctly normalizes this reformulated likelihood. We use this factorized version (Equation (3.15)) for efficient optimization and evaluation in the following.

Optimization w.r.t. $\boldsymbol{\beta}$. Setting the derivative of Equation (3.15) with respect to $\boldsymbol{\beta}$ to zero we obtain:

$$\hat{\boldsymbol{\beta}} = \left[\sum_{i=1}^{N} \frac{1}{\left(\mathbf{S}_{i,i}+\delta\right)} \left[\mathbf{U}^{\mathrm{T}}\mathbf{X}\right]_{i:}^{\mathrm{T}} \left[\mathbf{U}^{\mathrm{T}}\mathbf{X}\right]_{i:}\right]^{-1} \left[\sum_{i=1}^{N} \frac{1}{\left(\mathbf{S}_{i,i}+\delta\right)} \left[\mathbf{U}^{\mathrm{T}}\mathbf{X}\right]_{i:}^{\mathrm{T}} \left[\mathbf{U}^{\mathrm{T}}\mathbf{y}\right]_i\right]. \tag{3.16}$$

Optimization w.r.t. σ_g^2. Optimization of Equation (3.15) requires evaluation of the normalization constant Z which we can conveniently write in terms of our diagonal covariance from Equation (3.15)

$$Z = (2\pi)^{\frac{N}{2}} \left|\sigma_g^2\left(\mathbf{S}+\delta\mathbf{I}\right)\right|^{\frac{1}{2}} = (2\pi)^{\frac{N}{2}} \sigma_g^2 \left(\sum_{i=0}^{N} \left(\mathbf{S}_{i,i}+\delta\right)\right)^{\frac{1}{2}}. \tag{3.17}$$

After substituting Z in the log-likelihood by the right hand side of Equation (3.17) we take its derivative w.r.t. σ_g^2. We find our maxima at

$$\hat{\sigma}_g^2 = \frac{1}{N} \sum_{i=0}^{N} \frac{\left(\left[\mathbf{U}^{\mathrm{T}}\mathbf{y}\right]_i - \left[\mathbf{U}^{\mathrm{T}}\mathbf{X}\right]_{i:}\hat{\boldsymbol{\beta}}\right)^2}{\left(\mathbf{S}_{i,i}+\delta\right)}, \tag{3.18}$$

which can be seen as the weighted mean squared error between projected response and (projected) linear prediction.

Optimization w.r.t. δ. Plugging the expressions for $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}_g^2$ back in to Equation (3.15) the log-likelihood only depends on δ

$$\begin{aligned}\mathrm{LL}(\delta) = &-\frac{1}{2}\left[\sum_{i=1}^{N}\log\left(\mathbf{S}_{i,i}+\delta\right) + N\log\frac{1}{N}\sum_{i=0}^{N}\frac{\left(\left[\mathbf{U}^{\mathrm{T}}\mathbf{y}\right]_i - \left[\mathbf{U}^{\mathrm{T}}\mathbf{X}\right]_{i:}\hat{\boldsymbol{\beta}}(\delta)\right)^2}{\left(\mathbf{S}_{i,i}+\delta\right)}\right] \\ &- \frac{N}{2}\log(2\pi) - \frac{N}{2}.\end{aligned} \tag{3.19}$$

Finding the maximum w.r.t. δ is a non-convex optimization problem for which no analytical solution is available. However, once our data $\{\mathbf{y}, \mathbf{X}\}$ has been rotated by $\mathbf{U}^{\mathrm{T}}$, a single evaluation of the log likelihood is linear in sample size N, as compared to $\mathcal{O}(N^3)$ using Equation (3.12). Thus, we revert to numerical optimization which requires multiple evaluations of the log likelihood for different values of δ. Lippert and others [34] use Brent's method [11], a simple one-dimensional optimization technique on a fine grid of predefined intervals over δ.

Univariate association tests. Here, we follow in outline the procedure used for the vanilla linear model (see section 1.3). The genetic feature to be tested for association and a one valued vector fitting the intercept are included in the design matrix $\mathbf{X}$ of the alternative model. Our baseline- or *null model* just includes the intercept. We optimize and evaluate the log likelihood using Equation (3.19) for both models and compute the LOD or P-Value as measure for association strength (see Equation (1.18)).

Lippert and others fit δ once on the null model keeping it fixed for all subsequent association tests. So, in order to assess each genetic feature, they just require a single evaluation of the log likelihood (Equation (3.19)). This trick was first implemented by Kang and others [28] helping to drastically reduce the overall runtime.

3.3. Multivariate linear mixed models

The multivariate linear models presented in Section 2 can be extended to also include random effects. Here, we present the linear mixed model LASSO (LMM LASSO) [50] referring to the work of Segura and others ([55]) for a treatment on forward selection. Based on the probabilistic interpretation we introduced in Equation (2.3), LASSO is completed to a

random effect model marginalizing over additional covariates as shown for unregularized linear models in Section 3.1

$$p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) \propto \mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{X}\boldsymbol{\beta},\, \sigma_g^2\boldsymbol{\Sigma} + \sigma_v^2\mathbf{I}\right) \prod_{j=1}^{M} \exp\left(-\frac{1}{2}\lambda\,|\beta_j|\right). \tag{3.20}$$

Applying the same algebraic tricks introduced in Section 3.2 we can write this posterior as

$$p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) \propto \mathcal{N}\left(\tilde{\mathbf{y}} \,\middle|\, \tilde{\mathbf{X}}\boldsymbol{\beta},\, \sigma_g^2\mathbf{I}\right) \prod_{j=1}^{M} \exp\left(-\frac{1}{2}\lambda\,|\beta_j|\right) \tag{3.21}$$

$$\text{where } \tilde{\mathbf{y}} = (\mathbf{S} + \delta\mathbf{I})^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{y} \text{ and } \tilde{\mathbf{X}} = (\mathbf{S} + \delta\mathbf{I})^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{X},$$

and we also rescaled the transformed data by $(\mathbf{S} + \delta\mathbf{I})^{-1}$ to obtain an isotropic Gaussian likelihood. In the space of the transformed data $\{\tilde{\mathbf{y}}, \tilde{\mathbf{X}}\}$ we arrive at the Bayesian view of the vanilla LASSO Equation (2.3).

Using LASSO solvers on Equation (3.21) requires that δ has been fixed beforehand. A reasonable estimation can be obtained by fitting δ once on the null model (i.e. the model that only includes the intercept) as applied in the univariate case (see Section 3.2).

The shrinkage parameter λ can be found, for instance, using cross-validation.

4

Decision tree based approaches

Decision trees are among the oldest machine learning methods and initially developed for use in medical- and military applications [8]. With the introduction of ensemble based approaches such as boosting [19–22] and bagging [9], recently, they regained popularity particularly in the fields of bioinformatics (e.g. [6, 41, 49, 59]) and image analysis (e.g. [5, 47, 56])

Decision trees can either be used for classification or regression [8]. Here, we restrict ourselves to a review of *least squares regression trees* [8] as they are extended by our method that is introduced in Part II. Furthermore, we show that, while being a non-linear method at the global level, building trees can be seen as fitting (multiple) linear models when considering individual operations during learning (splits).

4.1. Least squares regression trees

We follow the notation of [8] and define a least squares regression tree by its set of nodes $T = \{t_0, \dots, t_n\}$, where t_0 denotes the root. Note, that indexes of the elements in T are sufficient to specify the exact structure of the tree (see Figure 4.1). Given our response vector of phenotypic observations $\mathbf{y}$ in N dimensions and our $N \times M$ matrix of genetic features $\mathbf{X}$, let $\mathbf{y}(t)$ be the subvector of observations associated to a node t and let $N(t)$ denote the total number of observations at t.

Learning

Learning starts with the tree having only a single node $T^{(0)} = \{t_0\}$ to which all observations are associated, i.e. $\mathbf{y}(t_0) = \mathbf{y}$. When growing the tree at root node t_0, the best splitting point $\hat{s}$ in the set of possible partitions $\mathcal{S}$ is then determined by maximizing the reduction in variance:

$$\hat{s} = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \, \mathrm{R}(t_0) - \mathrm{R}(t_1|s) - \mathrm{R}(t_2|s), \tag{4.1}$$

which leads to daughter nodes t_1 and t_2. Here we defined for a given node t

$$\mathrm{R}(t) = \frac{1}{N(t)} \sum_{i \in t} (y_i - \bar{y}(t))^2$$

which corresponds to the within-node variance, and $\bar{y}(t)$ to the samples' mean at t.

Viable partitions $s \in \mathcal{S}$ are implied by the candidate features $\mathbf{x}_j$, as given by the column vectors of $\mathbf{X}$. If we assume that - for the sake of simplicity - we are having binary values, there exists at most one possibility to split per feature $\mathbf{x}_j$, i.e. all samples where $x_{ij} = 1$ will be separated from samples having $x_{ij} = 0$. So each feature contributes at most one split to $\mathcal{S}$ thus allowing us to operate on the features' indexes $j \in \{1, \ldots, M\}$ directly. We therefore rewrite equation 4.1 as

$$\hat{j} = \underset{j \in \{1,\ldots,M\}}{\operatorname{argmax}} \; \Delta\mathrm{R}'(\mathbf{x}_j, t_0) \tag{4.2}$$

where

$$\Delta\mathrm{R}'(\mathbf{x}_j, t_0) = \mathrm{R}(t_0) - \mathrm{R}(t_1 | x_{ij} = 0) - \mathrm{R}(t_2 | x_{ij} = 1).$$

The objective of variance minimization (Equation (4.1)) can equivalently be formulated as maximum likelihood selection of splitting points in a linear model

$$\hat{j} = \underset{j \in \{1,\ldots,M\}}{\operatorname{argmax}} \; \mathrm{LL}(\mathbf{y} \,|\, \hat{\beta}_b, \hat{\beta}_j, \hat{\sigma}_v^2, \mathbf{x}_j) \tag{4.3}$$

where

$$\mathrm{LL}(\mathbf{y} \,|\, \hat{\beta}_b, \hat{\beta}_j, \hat{\sigma}_v^2, \mathbf{x}_j) = \log \mathcal{N} \left(\begin{bmatrix} \mathbf{y}(t_1) \\ \mathbf{y}(t_2) \end{bmatrix} \middle| \, \hat{\beta}_b \begin{bmatrix} \mathbf{1}(t_1) \\ \mathbf{1}(t_2) \end{bmatrix} + \hat{\beta}_j \underbrace{\begin{bmatrix} \mathbf{0}(t_1) \\ \mathbf{1}(t_2) \end{bmatrix}}_{\mathbf{x}_j}, \hat{\sigma}_v^2 \mathbf{I} \right). \tag{4.4}$$

Here, $\mathbf{y}(t_1)$ and $\mathbf{y}(t_2)$ are reordered versions of $\mathbf{y}$, such that they correspond to all samples with $x_{ij} = 0$ and $x_{ij} = 1$ respectively. In this representation, $\hat{\beta}_b$ and $\hat{\beta}_j$ denote sample bias at node t and splitting weight of $\mathbf{x}_j$. We summarize this equivalency in

Proposition 1. *Let* $\mathbf{x}_j \in \{\mathbf{x}_0, \ldots, \mathbf{x}_M\}$ *be a binary predictor and* $\mathbf{y}(t)$ *be*

the measurements associated with node $t \in T$, then

$$\hat{j} = \underset{j \in \{1,\ldots,M\}}{\operatorname{argmax}} \; \Delta R'(\mathbf{x}_j, t) = \underset{j \in \{1,\ldots,M\}}{\operatorname{argmax}} \; LL(\mathbf{y}(t) \,|\, \hat{\boldsymbol{\beta}}, \hat{\sigma}_v^2, \mathbf{x}_j).$$

A proof of this statement is given in Appendix C.

Splitting proceeds recursively from the subtrees rooted by the daughter nodes t_1 and t_2. Growing of the regression tree is stopped at terminal (or leaf) node t_T if, either no more splitting is possible (i.e. t_T contains a minimal number of samples) or the scoring function shows no improvement (i.e. $\Delta R'(\mathbf{x}_j, t) \approx 0; \forall j$).

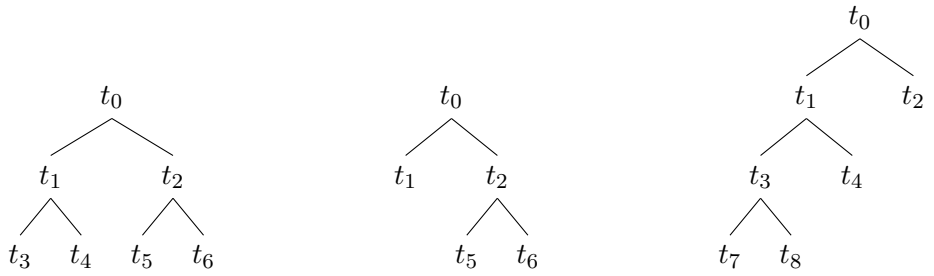


Figure 4.1.: Examples of binary regression using labelling introduced in [8]. Nodes t are indexed starting with zero and subsequently labelled breath-first. Nodes that do not exist are still counted by the index (as it is the case for the tree in the middle). Using this labelling a node's index specifies its exact position in the tree.

Features having more than two levels. In cases where we have more than two levels per predictor, we can as well convert this problem into our linear model perspective. The idea is to transform our matrix of features $\mathbf{X}$ into an equivalent matrix $\tilde{\mathbf{X}}$ that leads to same set of splits S.

Suppose that a predictor $\mathbf{x}$ has three levels, say 0, 1 and 2[1]. In order to partition a given node t in a *binary* regression tree we have two possibilities: joining all individuals that have $\mathbf{x}_i < 1$ into the first daughter node and the remainder into the second daughter node, or splitting the individuals that have $\mathbf{x}_i < 2$ from the rest.

[1]This encoding is common for organisms that have two copies of a chromosome such as humans. Here we have three possible combinations of alleles for a given marker: minor-minor, major-minor and major-major.

Instead of using $\mathbf{x}_j$ directly we can construct two binary features, where for the first we set all entries having $\mathbf{x}_j < 1$ to 0, the rest being set 1. For the second, all entries in $\mathbf{x}_j$ smaller than 2 are set to 0 (again, the rest being set to 1). Now, we can apply our splitting objective (Equation (4.1)) and our linear model equivalent (Equation (4.3)) on the alternative features which will lead to the same splits considered than working on $\mathbf{x}$ directly.

It is straightforward to extend this procedure to features having more than three levels. For each additional level we have to introduce a new binary feature. Consequently, a single feature having k levels can be replaced by $k - 1$ binary features.

Prediction

Using a tree T to predict an outcome (such as disease sate) to given input $\mathbf{x}_\star$ (e.g. the genetic makeup of an individual) is the result of a series of binary decisions. Starting with the root node t_0, the first decision is based on its feature selected for splitting during learning. Let j be the index of this feature, we proceed to child t_1 if $x_{j\star} = 0$ otherwise to t_2. We apply this procedure recursively for the subtrees rooted by t_1 or t_2 until a terminal node t_T is reached. The prediction (response) $y_\star$ is given by the average over the training samples associated to t_T, i.e. $y_\star = \bar{y}(t_T)$.

4.2. Random Forests

Random forests (RF) [10] are among the most commonly used ensemble methods. Individual learners - here decision trees - are built on a noisy variant of the training sample (bootstrap). To add further variation to individual trees, splits are performed on a random subset of all available features.

Predictions are obtained by aggregating responses of individual trees. This procedure of bootstrapping, fitting several weak learners and aggregation makes random forest an instance of a bagged[2] learning method [9].

In our particular case of random forest regression, let $\{T_b\}_1^B$ denote such an ensemble of B trees, whereas each learner is built on a random subsample[3] of the data $\{\mathbf{X}, \mathbf{y}\}$. For a test input $\mathbf{x}_\star$, prediction is defined to be the

[2] "bagging" is an acronym for **b**ootstrap **agg**regation

[3] Both versions, subsampling with and without replacement can be applied

average response of individual trees, i.e.

$$f_{\mathrm{rf}}^{B}(\mathbf{x}_*) = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x}_*). \tag{4.5}$$

Out of bag error

The out of bag error was introduced as an unbiased measure of predictive accuracy for bagged predictors alongside with the initial publication of random forests [10].

As a consequence of subsampling some part of the training data remains unused while building each tree T_b, the so called *out of bag sample*. Conversely, to each datum $\{\mathbf{x}_i, y_i\}$ we have a set of trees where $\{\mathbf{x}_i, y_i\}$ is part of the out of bag sample. We can use this subensemble to make a prediction for $\mathbf{x}_i$ (analog to Equation (4.5)) and take y_i to compute the corresponding prediction error.

Repeating this procedure for the rest of the training sample, we take the mean over individual prediction errors to obtain the *out of bag error*.

Feature scoring

Several feature scoring measures have been proposed for random forests, here we give an overview about the three that are most commonly used.

Breiman [10] suggested the so called *permutation importance*. That is, after learning the ensemble of trees, values of a specified feature are randomly permuted. The resulting difference in predictive accuracy (estimated with the out of bag error) serves as importance measure for the feature specified.

A simple alternative is the *selection frequency (SF)* which is the total number a given feature is selected for splitting during learning. Despite being prone to several biases [6, 59] it has proven to be the most sensitive score towards interacting features [41] when compared to alternative measures and methods.

In all our experiments we use the residual sums of squares (RSS) which is the default importance measure of the random forest R package [33].

Briefly, at each node in a tree we add the improvement in variance to the score of the feature chosen for splitting. The sum over all improvements for a given genetic feature is used as final score. This measure behaves similar

to the selection frequency when it comes to the detection of interacting features [41]. Being on a continuous scale, however, the RSS provides a higher resolution (as compared to selection frequency) which makes this measure is particularly attractive when large data sets with thousands of SNPs and individuals are considered.

Part II.

Mixed random forests

5

Mixed model regression trees

Mixed model regression trees extend least squares regression trees by including an additional random effect term (see Figure 5.1). It is the same modelling principle that we applied in Chapter 3 to obtain linear mixed models (LMMs). At every stage of building a mixed model regression tree, variation in the data is modelled by a split according to a selected feature and the random effect. As a consequence of this joint modelling approach, features tend to be selected that lead to splits minimizing those parts of variance which are not captured by the random effect.

In this chapter, we formalize our idea of the mixed model regression tree and explain how computational speed-ups introduced for LMMs (Section 3.2) can be adapted for our approach, thereby enabling applications to large (genetic) datasets.

5.1. A linear mixed model view

The key innovation of mixed model regression trees is to take advantage of the linear model perspective introduced with least squares regression trees (Equation (4.3)). If we subsequently extend this splitting model by a random effect we obtain following linear mixed model

$$\mathrm{LL}(\mathbf{y} \mid \beta_b, \beta_j, \boldsymbol{\Sigma}, \sigma_g^2, \delta, \mathbf{x}_j) = \log \mathcal{N}\left(\mathbf{y} \,\middle|\, \beta_b \mathbf{1} + \beta_j \mathbf{x}_j, \sigma_g^2 \left(\boldsymbol{\Sigma} + \delta \mathbf{I}\right)\right). \quad (5.1)$$

Given feature $\mathbf{x}_j$, our objective is to perform a least squares optimization w.r.t. weights β_b (bias), β_j (splitting weight) and the random effect variance σ_g^2. Like for the LMM and the LMM LASSO in Section 3.2 we estimate δ once on the null model, keeping it fixed during subsequent branching decisions.

In the following, we omit constant parameters on the conditioning set of the likelihood in order to simplify our notation. To assess feature $\mathbf{x}_j$ as

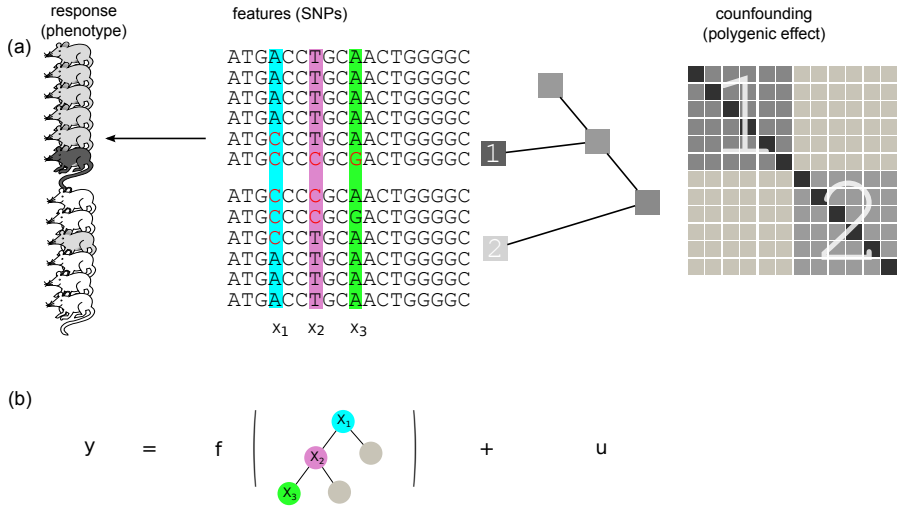


Figure 5.1.: Schematic overview of mixed model regression trees. **(a)** QTL data is considered where individuals are descendants from two different populations as illustrated by the phylogenetic tree. Confounding effects caused by relatedness between individuals are captured by the random effect term with covariance shown on the right. Submatrices corresponding to within relatedness of the two populations are indicated by numbers. Remaining parts of the matrix model cross-relatedness. **(b)** Response $\mathbf{y}$ (coat colour of mice) is modelled by the sum of a (genetic) fixed effect and a random effect. The fixed effect is captured using a decision tree; at the same time, the random effect $\mathbf{u}$ with the covariance derived from the population phylogeny explains confounding by population structure effects. As a result of this joint learning, splits in the regression tree are more likely to occur along informative features that are orthogonal (i.e. not correlated) to confounding effects. In this example the mouse coat colour is a non-linear function of the three polymorphic sites X_1, X_2 and X_3.

candidate for splitting, we evaluate

$$\mathrm{LL}(\mathbf{y} \,|\, \hat{\beta}_b, \hat{\beta}_j, \hat{\sigma}_g^2, \mathbf{x}_j) = \log(\mathcal{N}\left(\mathbf{y} \,\middle|\, \hat{\beta}_b \mathbf{1} + \hat{\beta}_j \mathbf{x}_j, \hat{\sigma}_g^2 \left(\mathbf{\Sigma} + \delta \mathbf{I}\right)\right)).$$

The index of the best predictor can then be found by

$$\hat{j} = \underset{j \in \{1,\ldots,M\}}{\operatorname{argmax}} \ \mathrm{LL}(\mathbf{y} \,|\, \hat{\beta}_b, \hat{\beta}_j, \hat{\sigma}_g^2, \mathbf{x}_j). \tag{5.2}$$

This optimization is equivalent to a (univariate) linear mixed model association test and thus the inference presented in Section 3.2 can be used.

Let $\hat{\mathbf{x}}_t := \mathbf{x}_{\hat{j}}$. As for least squares regression trees (Section 4.1), we apply

reordering by combining the samples for which $\hat{\mathbf{x}}_{it} = 0$ into $\mathbf{y}(t_1)$ and leave remaining observations to $\mathbf{y}(t_2)$.

In contrast to least squares regression trees, we note that further splitting of the samples in t_1 cannot be regarded independently from the samples assigned to t_2 and vice versa. Both branches of the tree remain coupled through the covariance $\mathbf{\Sigma}$ and hence the full dataset needs to be considered in subsequent splits. Tree growing, however, proceeds by recursively selecting further predictors while keeping $\hat{\mathbf{x}}_t$ and the bias in the model. For node t_2 we consider the following mixed model

$$\mathrm{LL}\left(\begin{bmatrix}\mathbf{y}(t_1)\\ \mathbf{y}(t_2)\end{bmatrix}\middle|\boldsymbol{\beta}_b, \beta_j, \sigma_g^2, \mathbf{x}_j\right) = \log \mathcal{N}\left(\begin{bmatrix}\mathbf{y}(t_1)\\ \mathbf{y}(t_2)\end{bmatrix}\middle|\, \boldsymbol{\beta}_b^{\mathrm{T}}\mathbf{C}_{t_2} + \beta_j \mathbf{1}_{[\hat{\mathbf{x}}_t=1]}(\mathbf{x}_j), \sigma_g^2(\mathbf{\Sigma} + \delta\mathbf{I})\right) \tag{5.3}$$

where the bias vector and $\hat{\mathbf{x}}_t$ are joined into the matrix

$$\mathbf{C}_{t_2} = \begin{bmatrix}\mathbf{1}(t_1) & \mathbf{0}(t_1)\\ \mathbf{1}(t_2) & \mathbf{1}(t_2)\end{bmatrix}$$

and $\mathbf{1}_{[\hat{\mathbf{x}}_t=1]}(\mathbf{x}_j)$ denotes the vector having x_{ij} at all the indices i where $\hat{x}_{it} = 1$, and 0 otherwise. Alternatively, one can regard $\mathbf{1}_{[\hat{\mathbf{x}}_t=1]}(\mathbf{x}_j)$ as modelling an interaction between predictors $\hat{\mathbf{x}}_t$ and $\mathbf{x}_j$. Importantly, all the previous weights (which are combined into $\boldsymbol{\beta}_b$) will be refitted and we find the index of the next splitting feature $\hat{\mathbf{x}}_{t_2}$ by

$$\hat{j} = \underset{j\in\{1,\ldots,M\}}{\operatorname{argmax}} \left\{\mathrm{LL}(\mathbf{y}|\hat{\boldsymbol{\beta}}_b, \hat{\beta}_j, \hat{\sigma}_g^2, \mathbf{x}_j)\right\}. \tag{5.4}$$

Optimization for node t_1 follows analogously replacing the last summand of the mean in Equation 5.3 with $\beta_j \mathbf{1}_{[\hat{\mathbf{x}}_1=0]}(\mathbf{x}_j)$ and $\mathbf{C}_{t_1}$ by

$$\mathbf{C}_{t_1} = \begin{bmatrix}\mathbf{1}(t_1) & \mathbf{1}(t_1)\\ \mathbf{1}(t_2) & \mathbf{0}(t_2)\end{bmatrix}.$$

5.2. Mixed random forests

The mixed random forest (mixed RF) is the method in focus of this thesis and obtained by bagging mixed model regression trees. We learn this ensemble analog to random forests creating random subsamples of the

data and building individual trees[1] Similarly, we only consider a random subset of all available features for each split (see Section 4.2).

Prediction

To a given test genotype $\mathbf{x}_\star$ we traverse each learned tree in the ordinary decision tree manner until we reach a terminal node and return its associated mean. Analog to a standard random forest, the response $m_\star$ is computed as the average over the means returned by the individual trees (see Equation (4.5)).

In addition, the population structure captured by the random effect term contributes to the predictive distribution. Under the random effect model, the joint distribution of training and test responses is a multivariate Gaussian

$$\begin{bmatrix} \mathbf{y} \\ y_\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m} \\ m_\star \end{bmatrix}, \sigma_g^2 \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{X},\mathbf{X}} + \delta\mathbf{I} & \mathbf{\Sigma}_{\mathbf{X},\mathbf{x}_\star} \\ \mathbf{\Sigma}_{\mathbf{x}_\star\mathbf{X}} & \mathbf{\Sigma}_{\mathbf{x}_\star,\mathbf{x}_\star} + \delta\mathbf{I} \end{bmatrix} \right) \tag{5.5}$$

where $\mathbf{m}$ is the training-fixed effect estimated during the forest building procedure. The training covariance $\mathbf{\Sigma}_{\mathbf{X},\mathbf{X}}$ and the cross covariance $\mathbf{\Sigma}_{\mathbf{x}_\star,\mathbf{X}}$ are obtained by subsetting $\mathbf{\Sigma}$ which has been estimated on the whole predictor matrix in advance. The predictive distribution for the unseen test phenotype y_* can be derived by conditioning on $\mathbf{y}$ and completing the square (see Equation (B.4) in Appendix B.2).

Fitting the optimal tree depth

Adjusting tree depth in the ensemble is a simple way to control model complexity [16] which can be efficiently done while training. For each tree T we use the *out of bag* sample (i.e. the part of the training set that is not used building T) to compute the *out of bag* prediction. Out of bag prediction for mixed random forests is the conceptual extension to that of random forests (see Section 4.2) when additionally accounting for a random effect. We formalize the joint model of in bag- and out of bag

[1] Note, that for each tree we have to provide the covariance matrix $\mathbf{\Sigma}_T$ considering the subsample, which is obtained by selecting corresponding rows and columns of the global covariance $\mathbf{\Sigma}$.

sample for each tree T as follows:

$$\begin{bmatrix} \mathbf{y}_b \\ \tilde{\mathbf{y}}_o \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_b \\ \mathbf{m}_o \end{bmatrix}, \sigma_g^2 \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{X}_b,\mathbf{X}_b} + \delta\mathbf{I} & \mathbf{\Sigma}_{\mathbf{X}_b,\mathbf{X}_o} \\ \mathbf{\Sigma}_{\mathbf{X}_o,\mathbf{X}_b} & \mathbf{\Sigma}_{\mathbf{X}_o,\mathbf{X}_o} + \delta\mathbf{I} \end{bmatrix} \right),$$

The estimation for the fixed effect is the vector composed of the tree's response $\mathbf{m}_o$ given out of bag sample features $\mathbf{X}_o$, and the mean fitted for in bag sample $\mathbf{m}_b$. The (cross)covariances of this model are obtained by subsetting $\mathbf{\Sigma}$ for in bag- and out of bag sample, respectively. We use the entire tree based model to make a prediction for $\tilde{\mathbf{y}}_o$, computing the conditional (Gaussian) distribution of $\tilde{\mathbf{y}}_o|\mathbf{y}_b$ (see Appendix B.2 for details). Averaging over all trees up to a particular depth, we obtain the out of bag prediction of the *whole* training vector $\tilde{\mathbf{y}}$ which is then compared to $\mathbf{y}$ computing the out of bag error.

We (re)evaluate this error after each cycle of the forest growing procedure increasing the depth of all trees by one. Tree growing is (usually) stopped if the out of bag error is not decreasing anymore. The (forest) depth resulting in the lowest error is returned.

6

Application of mixed random forest to genomic data

In this chapter, we evaluate the proposed mixed random forest. First, we simulate a wide range of genetic architectures to show that the model is able to improve detection of genuine (genetic) signals when compared to a standard random forest and linear mixed models. In the second part, we map hundreds of individual gene-expression levels measured in mice to demonstrate that associations uncovered by mixed RF are in better agreement with known pathway annotations than those detected by competing methods. Finally, we apply mixed RF to QTL data of the same mouse cohort, to show that our method is able to recover complex genetic models.

6.1. Association mapping on simulated data

In order to validate the mixed RF, we initially consider synthetic datasets where the ground truth genetic architecture is known. We use genotype data of *Arabidopsis thaliana* in form of single nucleotide polymorphisms (SNPs) covering a total of 1,179 samples [1]. These data are well suited for this test as the given *A. thalina* population is very structured and hence effects due to population structure can be effectively simulated (see also discussion in [1]).

We compared alternative association models in their ability to recover true causal genetic markers. In addition to mixed random forest (mixed RF), we include the standard random forest (RF), LASSO and LMM LASSO [50]. Introduced in Chapter 3.3, the LMM LASSO extends LASSO by a random effect that accounts for population structure. As further references, we consider the univariate linear model (LM) and the linear mixed model (LMM) both of which assume that a single causal locus underlies the trait (see Chapters 1.3 and 3.2).

Method parameter settings. In general, we aim to keep parameter settings between RF and mixed RF as consistent and comparable as possible. We therefore learn ensembles of 250 trees for both, RF and mixed RF. Each tree is grown on a bootstrap sample (sampling with replacement) of full training set size, whereas a random subsample of 2/5 of all available predictors is used to find each split.

To obtain feature scores for both the LASSO and LMM LASSO we follow the procedure of [50] and rank features by their order of inclusion into the LASSO model. Univariate linear models (LM and LMM) use the LOD as importance measure (see Chapters 3.2 and 1.3, respectively).

For methods correcting population structure (LMM, LMM LASSO and mixed RF), we set the covariance $\mathbf{\Sigma}$ to the realized relationship matrix as used for simulation.

Simulation setup. To generate a synthetic dataset, we randomly select 1,000 from a total of 214,553 genome wide SNPs with a minor allele frequency > 0.1 and simulate a total of 100 traits for 250 individuals in our **baseline setting** as follows: three SNPs are picked randomly to simulate linear additive and a further three pairs of SNPs contribute epistatic effects

$$\begin{aligned}\mathbf{y} = &\underbrace{\mathbf{x}_1\beta_1 + \mathbf{x}_2\beta_2 + \mathbf{x}_3\beta_3}_{\text{additive effects}} \\ &+ \underbrace{\text{int}(\mathbf{x}_4, \mathbf{x}_5)\beta_4 + \text{int}(\mathbf{x}_6, \mathbf{x}_7)\beta_5 + \text{int}(\mathbf{x}_8, \mathbf{x}_9)\beta_6}_{\text{epistatic interactions}} \\ &+ \mathbf{u} + \boldsymbol{\psi}\end{aligned} \tag{6.1}$$

where

$$\beta_i \sim \mathcal{N}\left(0,\, \sigma_\beta^2\right),\ \mathbf{u} \sim \mathcal{N}\left(0,\, \sigma_{\mathbf{u}}^2\mathbf{\Sigma}\right) \text{ and } \boldsymbol{\psi} \sim \mathcal{N}\left(0,\, \sigma_g^2\mathbf{I}\right).$$

Interactions are simulated by randomly picking two genetic features and taking the component-wise binary product as indicated by the "int" operator above. The resulting vector is multiplied by the simulated effect size β_i. The polygenic effect $\mathbf{u}$ is a sample from a multivariate Gaussian having the realized relationship matrix $\mathbf{\Sigma}$ (see Appendix B.1) as covariance. Here, $\mathbf{\Sigma}$ is constructed from another subsample of 1,000 SNPs. Contributions of fixed genetic-, polygenic effect and independent Gaussian noise to the total trait variance are split into 0.375:0.5:0.125 adjusting $\sigma_\beta^2, \sigma_g^2$ and σ_v^2 accordingly.

To simulate different genetic settings, we vary the number of additive terms (from 1–20 in Figure 6.1 (a)), the fraction of additive- to interaction terms (6:0 – 0:6, Figure 6.1 (b)) and relative contributions of population structure and independent Gaussian noise $\frac{\sigma_g}{\sigma_v}$, (0.1 – 0.9, Figure 6.1 (c)) adjusting the simulation setup (Equation (6.1)) accordingly.

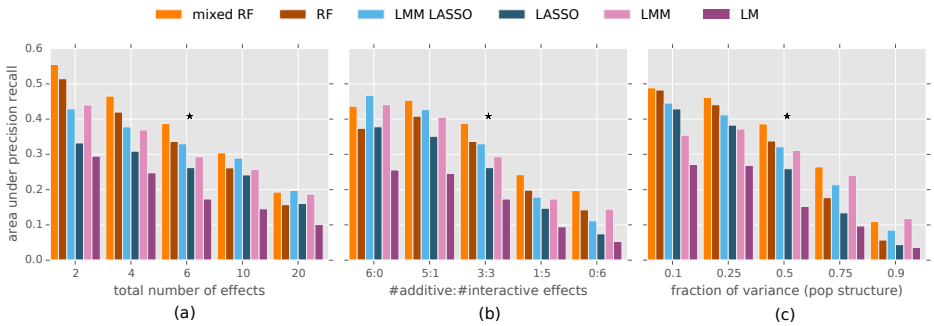


Figure 6.1.: Comparison of alternative methods to identify causal genetic loci on simulated datasets. The proposed mixed random forest (mixed RF) is compared to the standard random forest (RF), a univariate linear association test (LM) and a multivariate linear model (LASSO). We also consider extensions of both linear models to also account for polygenic background (LMM and LMM LASSO). Methods are assessed by their ability to recover true causal loci as measured by the area under the precision recall curve (PR). The asterix indicates our baseline setting including 3 additive and 3 multiplicative effects and 50% of the phenotypic variance explained by population effect (see Figure E.1 in Appendix E for PR curves of alternative methods on our baseline setting). In individual simulations we vary the total number of effects **(a)**, the ratio between direct additive and epistatic effects **(b)** and the relative contribution of population structure to the phenotypic variance **(c)**.

Results. For each of the 15 simulation settings, we compare the accuracy of all methods in terms of the area under the precision-recall curve (Figure 6.1). Briefly, this approach quantifies the precision (proportion of correct predictions) as a function of the recall (sensitivity), thereby avoiding choosing arbitrary cut-off values for significance when comparing alternative methods. See Figure E.1 in Appendix E for an example of a typical precision-recall curve.

As expected, the vanilla random forest (RF) is more accurate than any of the linear models in regimes where epistatic effects dominate the association signals (Figure 6.1 (b)). However, the performance of RF is severely

affected by population structure (Figure 6.1 (c)). RF is outperformed by linear models correcting for population structure, whenever traits are significantly affected by confounding effects. Unlike RF, our mixed RF is not affected by these artefacts, demonstrating how the proposed random effect extension is able to account for population structure. The mixed RF is also remarkably robust with respect to the trait architecture (Figure 6.1 (b)) and the number of of causal variants (Figure 6.1 (a)). Notably, our model performs favourably even in the limit of a purely additive architecture (Figure 6.1 (b)), where the mixed RF achieves a level of accuracy similar to that of the LMM LASSO, which *a priori* imposes a linear additive genetic architecture.

6.2. Mapping expression QTLs in mouse Hippocampus data

A significant drawback of simulations is that they inevitably require to make assumptions about the genetic architecture of traits and the nature of noise in the data. We therefore sought to additionally assess mixed RF based on quantitative trait loci (QTL) data without requiring any assumptions about how traits are affected by genetic variants. However, in real settings, accurate ground truth information for genotype-phenotype associations is difficult to obtain and hence it is necessary to revert to a bronze standard. Our approach is based on the notion that expression QTLs (eQTLs) at genes that are functionally related to the target genes whose expression they affect are more likely to be true than eQTLs not fulfilling this criterion [41]. This benchmark does not account for *cis* associations where the marker is close to the target gene itself (see Figure E.2 in Appendix E for the relative proportion of *cis* and *trans* effects by different methods). A possible concern of this approach is that several genuine associations may not be in agreement with the pathway databases, for example because of incomplete annotations. Nevertheless, this scheme to assess associations provides a robust test as results are aggregated over hundreds of target genes and individual eQTLs.

Reactome analysis. Here, we consider gene expression from mouse Hippocampus as phenotypes [25] and assess the plausibility of eQTLs using known pathways obtained by the Reactome database [27]. Because of the low number of unique genetic markers (12,545, from an inbred cross of eight

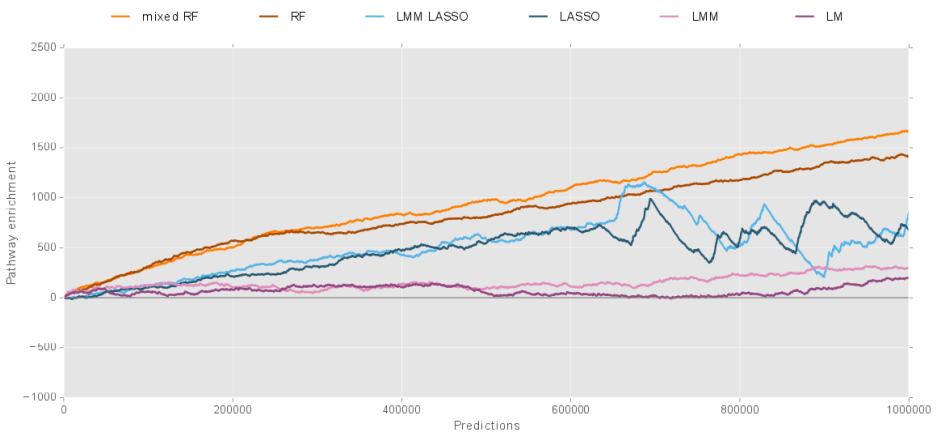


Figure 6.2.: Concordance of detected expression QTLs using alternative methods with annotated pathways from Reactome. For a total of 468 individuals a selection of 300 gene-expression traits are mapped to 12,545 available SNP features using the same methods as in Figure 6.1. Resulting feature scores are ranked for each trait (and method). We compute the *pathway enrichment* as the number of SNPs up to a given feature importance rank that are in consensus to the links inferred from Reactome. We normalize this measure by subtracting the number of consensus links that are to be expected under random ranking of SNPs. Shown is the averaged enrichment across all the expression traits. To exclude *cis* effects, SNPs located within a window of 500kB of the gene are discarded.

founders), we apply individual models to individual chromosomes where the population structure is estimated from all remaining chromosomes. While this leave-one out approach may miss inter-chromosomal epistatic interactions between markers, it has been shown to avoid proximal contamination when the same SNPs are used for mapping and for predicting population structure [35]. Jointly considering all genome-wide markers leads to similar overall conclusions, where the performance of all mixed-model based approaches is decreased (see Figure E.3 in Appendix E).

From a total of 19,892 expression traits we select the top 10 percentile when ranked by variance (1,989). From these, 373 could be associated to at least a single Reactome pathway. To elaborate our SNP to pathway memberships, we consider all (ENSEMBL) annotated genes within a 500kB window from the SNPs position in the genome. A SNP $\mathbf{x}_j$ is linked to a expression trait (*via* Reactome) if at least one of its associated pathways contains a gene that is in proximity to $\mathbf{x}_j$. We do not consider links that are induced by *cis* effects according to our 500kB window. We

further exclude expression traits that are linked to less than 10 or more than 1,000 SNPs.

For each method we rank SNPs by their scores for all of the 300 remaining traits. Each point on a curve as shown in Figure 6.2 reports the number of Reactome-consistent (i.e. plausible) associations that are recovered relative to the number of "hits" expected under random SNP ranking.

Method parameter settings. Method parameter settings and feature importance measures are the same as in the simulation study (Section 6.1), for RF, mixed RF and both univariate linear models (LM and LMM). In case of LASSO, we find that the recently proposed stability selection (see [40] and Chapter 2.3) is more robust than inclusion rank for scoring eQTL when given genetic features are in linkage disequilibrium (see also the discussion in [50]). We apply stability selection as follows: for each expression trait we randomly sample 90% of the data without replacement and learn the LASSO/LMM LASSO model such that it includes 20 features (adjusting the shrinkage parameter accordingly). We repeat random sampling and learning 1,000 times, reporting the fraction of times a feature is selected as importance score.

As for the simulation study, we require the inter-sample covariance $\mathbf{\Sigma}$ which can be estimated on the basis of genetic features $\mathbf{X}$, when (for instance) using the Realized Relationship matrix (RRM) [23].

To estimate $\mathbf{\Sigma}$, we first use a simple linear association test to rank all SNPs by their LOD-score [31] and subsequently select the top 1000 genetic features in order to build the RRM. This ranking avoids inclusion of features that explain little of the overall variance (see also [35] for a discussion on selecting subsets of features for building RRMs).

Results. Notably, all considered methods identify regulator-target gene associations with greater frequency than random assignment, which confirms that this assessment using the Reactome database is an informative criterion to evaluate alternative association methods. Moreover, the systematic difference between linear association models and the random forest approaches underlines once more the importance to account for epistatic effects [4, 13, 39, 49].

This analysis also confirms that modelling population structure is important: methods correcting for population structure perform better than their non-correcting counterparts (LMM *versus* LM; LMM LASSO *versus*

LASSO, mixed RF *versus* RF). Although this trend is weak, we observe it consistently for all three classes of mapping methods considered. Finally, the proposed mixed random forest yields associations that are more enriched among known pathway annotations than any other method, again demonstrating the merits of combining methods to correct for confounding effects with non-additive association mapping. The differences between RF and mixed RF are most visible in the tail of the association distribution, suggesting that in particular weak associations are obscured by population structure, if not accounted for.

6.3. Phenotype prediction

Complementary to evaluating methods in terms of recovering true associations, the ability to explain phenotypic variation has recently gained considerable attention. The task of phenotype prediction is also linked to "missing heritability" and several studies have noted that single marker association methods are not sufficient to fully explain the heritable component of phenotype variability [4, 45, 64, 67]. Indeed, more complex genetic models, for example considering epistatic effects, have been shown to significantly improve the fraction of explained phenotypic variance in out of sample prediction experiments [4, 67].

To this end, we also investigate the ability of mixed RF to predict phenotype from genotype. Our assessment is based on phenotypes measured in heterogeneous stock mice [62], which is the same outbred population of mice used in the Hippocampus eQTL study above (Section 6.2) and characterized by a strong family structure. As physiological and behavioural traits are highly complex, we expect them to be affected by a comparably large number of genetic variants. We compare our mixed RF to the vanilla RF, LASSO and, its counterpart modelling population effects, LMM LASSO [50]. Univariate linear approaches (i.e. LMM and LM) are not included in this analysis, as they are conceptually inappropriate for prediction tasks.

In addition, we compare all methods above to the **B**est **L**inear **U**nbiased **P**redictor (BLUP) [52] which corresponds to the mixed RF and the LMM LASSO when the estimation of direct genetic factors is dropped such that prediction is solely based on the (marginal linear) model of the polygenic background. Note, that we derived the BLUP (without explicitly naming it) in Chapter 3.1.

Methods

Mouse data. We select a total of 124 phenotypes (ranging from biochemical to behavioural traits) measured in a total of 1904 mouse HS individuals [57, 62] (see Table E.1 in Appendix E for a full list). Parts of the same cohort are used for eQTL mapping (see Section 6.2), thus we have the same genotype information of 12,545 genome-wide SNPs.

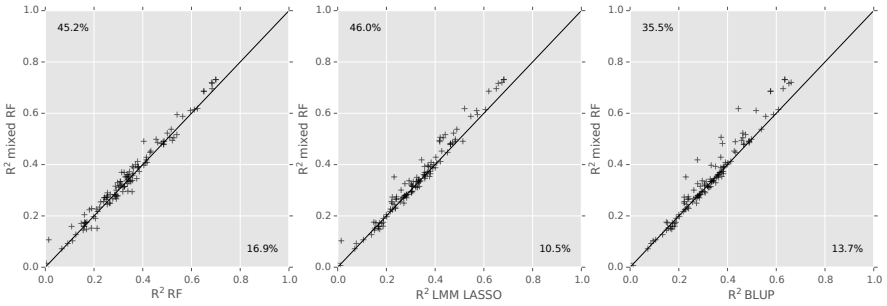
Model parameter settings. Compared to the eQTL study considered before (Section 6.2), we have a larger samplesize and in turn more features are likely to explain phenotypic variance. Thus, we use all 12,545 genetic features to estimate the realized relationship matrix. Nevertheless, a rank-based feature filtering as used before might further improve performance of the methods that handle population structure.

For the prediction of mouse phenotypes we use ensembles of 100 trees (mixed RF and RF). In case of the mixed RF, we learn each regression tree on a bootstrap sample of half the training set (drawn without replacement). This leaves the remaining half of the training data to adjust the depth of the trees. For the vanilla RF, we keep the bootstrapping as for feature selection (subsampling with replacement), since the used python package [48] does not provide subsampling without replacement.

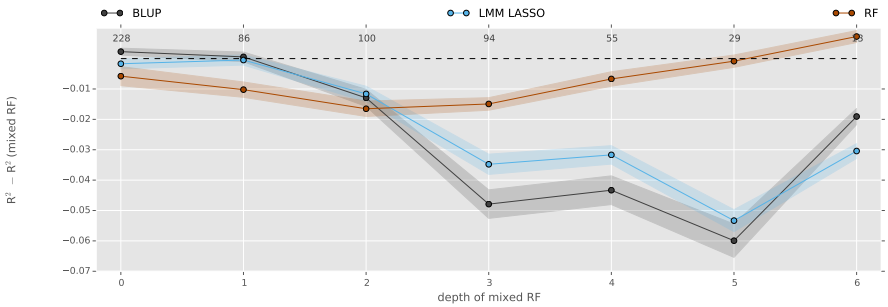
We give a runtime evaluation for this task of phenotype prediction in Chapter 8.

Assessment of alternative models. The performance of all models is quantified *via* randomized three-fold cross-validation. That is, for each prediction experiment we randomly sample two third of the data for training and use the remaining third of the sample for validation. Predictions are then assessed using the squared correlation coefficient (R^2) on the test set. For each phenotype and method, we repeat this procedure five times and report the average over correlation coefficients (Figure 6.3(a)). The full list of selected phenotypes with correlation coefficients for all methods is contained in Table E.1 of Appendix E. In order to quantify the relative performance of various methods, we report the fraction of phenotypes where one method performs better than another. We call two averaged R^2s significantly different if there is no overlap in intervals according to standard errors.

We use that notion to investigate the performance of mapping methods



(a)



(b)

Figure 6.3.: Prediction of mouse phenotypes. (a) Accuracy of alternative methods in predicting 124 traits is assessed in a randomized three-fold cross validation. For given phenotype and method we compute the squared correlation coefficient (R^2) to the test fold and report the averaged R^2 across five random restarts as final measure for accuracy in prediction. Averaged R^2s of mixed RF are then compared to RF, LMM LASSO and the best linear unbiased predictor (BLUP) by correlation plots. Univariate methods (LMM and LM in Figures 6.1 and 6.2) were excluded from this comparison as they are conceptually inappropriate for prediction tasks. (b) Performance of alternative models as a function of trait complexity estimated by the fitted mixed RF tree depth. For each method we consider all random restarts from (a) and group correlations to held out samples by the optimal depth found for the corresponding mixed RF. Shown is the average improvement in R^2 w.r.t. the mixed RF. Shaded areas indicate the bounds of the standard error and digits above the graphs the number of random restarts each given depth.

as a function of trait complexity (Figure 6.3(b)): we quantify each trait's complexity as the depth of the respective mixed RF model and subsequently analyse the performance of the association methods as a function of that measure. Note, as depth was estimated within each of 620 ran-

dom restarts of the cross-validation, some traits may end up in several complexity classes.

In general, we observe that the proposed mixed RF is the most robust predictor across the entire spectrum from simple to complex traits (Figure 6.3(b)). Whereas linear methods (BLUP and LMM LASSO) perform similarly to mixed RF for simple traits (Figure 6.3(b), tree depths ≤ 1), their predictive power breaks down for more complex traits (i.e. tree depth > 1). In particular, the improved performance in comparison to the LMM LASSO indicates that non-linear structure (like epistasis) is present in the data, which is better captured by the regression tree based approaches.

Furthermore, RF and mixed RF become similar in performance if the fitted depth of the ensemble gets large (Figure 6.3(b), tree depths > 4). A possible explanation is that the amount of variance attributable to population structure decreases in relative magnitude as trait complexity increases and/or that assuming linearity to model population structure is adverse in some of these cases. On this note, we find for tree depths of 6 (where the vanilla RF shows best average performance) the number of cases is low (13) when compared to the total of 605 random restarts considered (indicated by numbers on top of Figure 6.3(b)). Note that, 15 random restarts with tree depth > 6 were were excluded from this analysis as they constituted less than 10 cases per depth.

6.4. Author contributions and acknowledgements

Oliver Stegle conceived the general idea. I implemented, optimized and conducted theoretical analyses of our apporach. Andreas Beyer, Oliver Stegle and I designed the mapping and prediction experiments. I performed and refined the analysis. Andreas Beyer, Oliver Stegle and I wrote the related manuscript. I like to thank Barbara Rakitsch for her help with questions related to the LMM LASSO.

7

Mapping of rare genetic variants

With this chapter, we widen the field of possible applications of mixed random forests to the mapping of *rare genetic variants*. As "rare" suggests, we refer to cases where only a limited number of individuals carry the minor allele in a given study population (usually less than 1%–5%). Nevertheless, rare variants are assumed to explain a significant amount of missing heritability [18,38] as well as to play an important role in complex phenotypes [38, 54].

Univariate association test such as the linear- and the linear mixed model introduced in Part I are conceptually inappropriate for detection of rare variants. Assessing a single genetic feature in isolation, they will receive only a limited number of cases that contain the minor allele and are consequently prone to fail in finding (statistically) meaningful scores. Also reweighting schemes accounting for underrepresentation of such variants are problematic as rare variants are virtually indistinguishable from sequencing errors when testing for univariate association.

Apart from improving data quality by increasing study population sizes or sequencing depth, the only way to increase sensitivity for rare variants is to leverage information from their (local) genetic context. Purpose designed methods make the assumption that rare variants segregate together with other common- and/or rare variants that are in linkage disequilibrium. Instead of scoring individual features they summarize multiple genetic features within a predefined genomic window by means of a single aggregated score [32,37,42] or - in a more rigorous Bayesian fashion - using random effect modelling [46, 63]. Figure 7.1 illustrates common principles where highlighted parts indicate the regions that are tested for association. The basic window-based framework as implemented by [32,37,42,63] (Figure 7.1(a)) lacks in that it just considers (independent) noise and the genomic region to be tested.

Random effect modelling, on the other hand, provides a simple and robust way to also incorporate effects of features outside the genomic window [46]. Accounting for variance explained by the genomic background avoids that

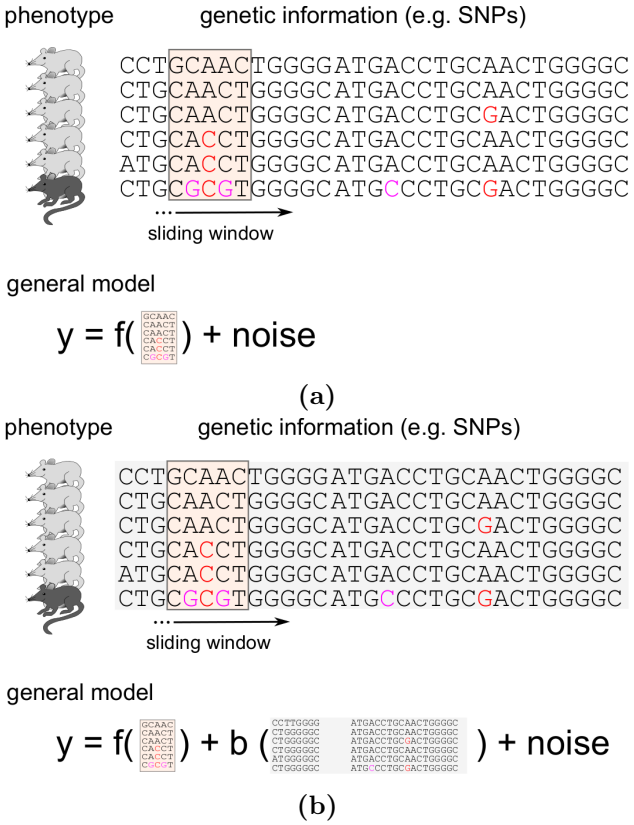


Figure 7.1.: Common principles of rare variant detection. Methods designed for rare variant testing usually employ a sliding window of fixed size (i.e. measured by number of bases). (a) genetic features within the window are used to predict phenotypic variance with a learned model f. Alternatively, scores that aggregate over features are used to assess linkage of a particular region. (b) extended version of the model in (a) that also accounts for the contribution of features in the genetic background extending the overall predictive model by another function b.

features within our window explain trait variance that can otherwise be attributed to correlated features within the genetic background.

This extension of the simple framework is also illustrated in Figure 7.1(b) and can be formalized as follows

$$\mathbf{y} = \mathrm{f}(\mathbf{X}_w) + \mathbf{u}(\mathbf{X}_r). \tag{7.1}$$

Here, $\mathrm{f}(\mathbf{X}_w)$ is the function of the region to be learned and $\mathbf{u}(\mathbf{X}_r)$ is the random effect correcting for the features which are not within the

foreground window.

So employing our mixed RF, we can use its random effect to also account for the genetic background and - unlike alternative methods ([32, 37, 42]) - being able to model non-linear effects within our window of genomic features considered (learning $\mathrm{f}(\mathbf{X}_w)$ through its regression tree component). In the following, we learn a separate mixed random forest for each window and score linkage of regions using the out of bag prediction error (see Chapter 5.2).

7.1. Experimental setup

We take genotype data from the same *Arabidopsis thaliana* population considered in Chapter 6.1 ([1]). Simulation of phenotypes, however, is fundamentally different.

From 386 individuals of *Arabidopsis thaliana* we consider a total of 50,435 SNPs from the first chromosome from which we uniformly sample five percent (2521) to simulate a genetic background. We take a sample of further 10 SNPs within a randomly selected region (r1) of 100kB to simulate direct additive effects as well as further 10 SNPs of another 100kB region (r2) to simulate five interactions. Simulation of a single phenotype can be formalized as follows

$$\begin{aligned}\mathbf{y} = \mathbf{X}^{(r1)}\boldsymbol{\beta}^{(r1)} &+ \operatorname{int}\left(\mathbf{x}_1^{(r2)}, \mathbf{x}_2^{(r2)}\right)\beta_1^{(r_2)} \\ &+ \operatorname{int}\left(\mathbf{x}_3^{(r2)}, \mathbf{x}_4^{(r2)}\right)\beta_2^{(r_2)} + \cdots + \operatorname{int}\left(\mathbf{x}_9^{(r2)}, \mathbf{x}_{10}^{(r2)}\right)\beta_5^{(r_2)} \\ &+ \mathbf{X}_{bg}\boldsymbol{\beta}_{bg} + \boldsymbol{\psi}.\end{aligned} \tag{7.2}$$

Here the superscripts r1 and r2 refer to the regions selected. The "int"-operator takes the component-wise product of selected (binary) feature vectors to simulate interactions. Furthermore, the background effect weight $\boldsymbol{\beta}_{bg}$ is a sample from a Gaussian distribution $\left(\boldsymbol{\beta}_{bg} \sim \mathcal{N}\left(\mathbf{0},\, 0.4^2\mathbf{I}\right)\right)$, and the region effect weights are both sampled from a bi-normal distribution, particularly

$$\beta_j^{(r1/r2)} = \mathcal{N}\left(1,\, 1\right) z - \mathcal{N}\left(1,\, 1\right)(1 - z), \tag{7.3}$$

where z is a Bernoulli distributed random variable taking value 0 or 1 with probability 0.5 . With this simulation we further ensure that effect sizes of individual regions are relatively small. Therefore, contributions of background-, foreground effects and independent Gaussian noise to the

total simulated trait variance are split into 0.5:0.1:0.4 (rescaling sampling distributions accordingly).

In comparison to mixed RF we consider the vanilla random forest, which does not account for genetic background but is also able to capture interactions, and two marginal linear models. The first variant tests for a linear kernel which is build on the features within the window (SKAT) [63]. The second variant, LMM SKAT [46][1], also considers the background through random effect modelling similar to that used for the mixed RF.

Alternative methods are used/implemented as follows:

Mixed random forest. Here, we use ensembles of 100 trees each fitted on the features considered in the (current) foreground window. To save computational resources, we reuse bootstraps that are created for the ensemble fitted on the first window. This allows us to reuse singular value decompositions of the trees' local covariances $\mathbf{\Sigma}_T$ (see Chapter 5.2) which would otherwise needed to be recomputed if new bootstrap samples are created.

We measure importance of each region by the out of bag prediction error which is obtained after fitting the optimal depth of the ensemble (see Chapter 5.2). To account for the genetic background signal, we construct the realized relationship matrix (RRM) (see Appendix B.1) on all features of the first chromosome.

Random forest. Here, we apply the vanilla random forest using our mixed random forest implementation by setting the covariance correcting for background ($\mathbf{\Sigma}$) to the identity matrix. The same parameter settings are applied as for the mixed random forest, including fitting of the optimal depth.

Marginal linear models. LMM SKAT uses the same RRM computed for the mixed RF to correct for genetic background. Features within each window are used to compute a second RRM (foreground covariance). Both foreground and background covariances are then combined into a marginal

[1]The method is originally termed ASKAT (**A**djusted **S**equence **K**ernel **A**ssociation **T**est) but is, for the sake of consistency to the other methods, referred to as LMM SKAT in the following.

log-likelihood model:

$$\mathrm{LL}(\mathbf{y}|\sigma_f^2, \sigma_b^2) = \log \mathcal{N}\left(\mathbf{y} \mid 0,\, \sigma_f^2 \mathbf{\Sigma}_f + \sigma_b^2 \mathbf{\Sigma}_b\right), \tag{7.4}$$

where weights for both, foreground and background covariances (σ_f^2 and σ_b^2) are fitted using python's limix package. For each window we obtain a marginal linear model, for which we compute the log-likelihood as the regions score.

Fitting of SKAT follows analog setting the RRM modelling the background effects to the identity matrix.

7.2. Results

We evaluate performance of alternative methods through summary plots which report the average number of recovered causal regions across 500 restarts of our simulation setup. Note, that we employ a sliding window, where "adjacent" regions are overlapping. We call all sliding windows which have overlap with any causal region as a true positive (if recovered), such that we arrive at a total of eight true positive windows (see Figure 7.2).

Considering all effects (Figure 7.2 (a)), we find that the LMM SKAT is superior over alternative methods, whereas our mixed RF ranks second before the (simple) variance component test (SKAT) and RF. Furthermore, we notice a comparably large difference between RF and mixed RF which underlines the importance to account for background signals, especially for the regression tree based approaches. In addition, we considered two alternative evaluations of our simulation experiment considering regions that carry additive effects as true positives (Figure 7.2 (b)) while ignoring regions carrying interactions and vice versa (Figure 7.2 (c)). Considering additive effects in isolation we find the same trends as before (Figure 7.2 (a)) while relative differences are larger. When it comes to the detection of interactions (Figure 7.2 (c)) alternative methods become similar in performance.

All in all, the proposed mixed RF is able to robustly assess the linkage of whole genomic regions. Nevertheless, using a state of the art method (LMM SKAT) gives similar or better results in the scenarios considered. An advantage of the LMM SKAT is its inherent Bayesian motivation of the (linear) foreground effect, which follows analog to that of the marginal

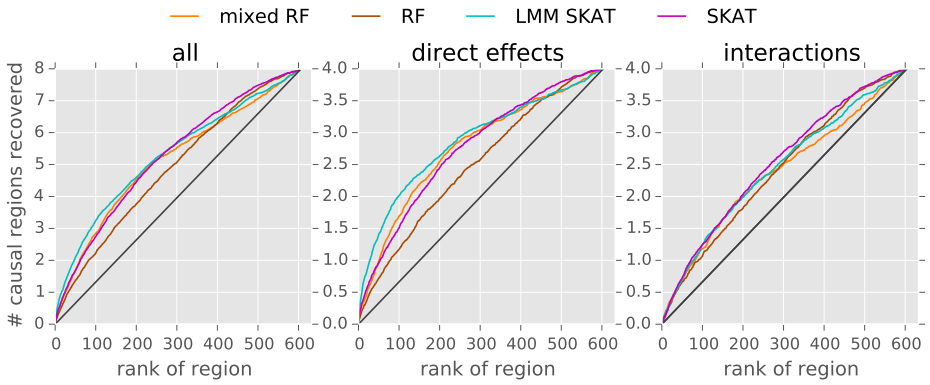


Figure 7.2.: Comparison of alternative methods for detecting rare variants on semi-empirical data using genotypes from *Arabidopsis thaliana*. For each of 500 randomly initialized experiments a simulated phenotype is influenced by a genomic region containing simple additive effects, a second region contributing epistatic effects as well as a genetic background signal (see text for details). Methods compute importance scores for individual regions considered in a sliding window. Plots show the number of causal regions recovered as a function of the regions' rank when averaged over the random restarts of the simulation.

linear model considered in Chapter 3.

Recursive kernels [17] combine the advantages of this Bayesian motivation while being able to account for non-linear structure in the data. They can be learned through our mixed RF framework and might therefore be an interesting alternative.

7.3. Author contributions and acknowledgements

Oliver Stegle conceived the idea. I implemented the underlying algorithm and contributed further ideas to its optimization. Oliver Stegle and I designed experiments on simulated data and I performed the analysis. My sincere thanks go to Paolo Casale who provided and supported me with the initial source code of our simulation study.

8

Theoretical Analysis of the mixed random forest

Theoretical analyses of random forests are challenging and most established insights build on empirical studies, e.g. [6, 41, 58, 59]. The main caveat for in depth studies lies in the dependence of the learned tree structure on the data-generating distribution $p(\mathbf{X}, \mathbf{y})$. To overcome this concern, Biau ([2]) uses a modification of the random forest algorithm dividing the training data into one set used for finding the optimal splits whereas the remaining samples are partitioned. It can be shown that the resulting learning algorithm is consistent and its rate of convergence solely depends on the number of informative features. e

However, given the non-i.i.d. sample structure in the focus of this work, the strategy proposed in [2] does not provide similar guarantees.

In the following, we give a runtime analysis and show that in the limit of no population structure (i.e. $\delta \to \infty$) our approach will behave like the standard random forest.

8.1. Runtime

Here, we consider the runtime for building a mixed random forest (i.e. the fitting stage). During prediction, the mixed random forest behaves like the standard random forest and requires no further treatment.

The main computational burden lies in the evaluation of the splitting function. In our case, this corresponds to testing $\mathcal{O}(M)$ predictors in a linear mixed model with $\mathcal{O}(N)$ samples. Making use of the same computational tricks introduced in [34] the runtime of a single split is bound by

$$c_t(d(t), N, M) \in \mathcal{O}(d(t)MN) \tag{8.1}$$

where $d(t)$ denotes the depth of node t, i.e. the distance from its root t_0 in T.

The runtime for building a single tree T is determined by its structure which in turn depends on the data generating distribution $p(\mathbf{X}, \mathbf{y})$ (see above). In general, we cannot estimate the runtime in the *average case* since this would require knowledge of the expected structure of T under $p(\mathbf{X}, \mathbf{y})$. Before proceeding with the *worst case* scenario, we consider the following alternative:

Balanced case. We suppose that our trees have a balanced structure, i.e. their depth is bounded by $\mathcal{O}(\log(N))$. In this case, the runtime for building as single estimator is dominated by the $\mathcal{O}(N)$ splits leading to its $\mathcal{O}(N)$ terminal nodes, i.e.

$$c_T^{\text{bal}}(M, N) \in \mathcal{O}(d(t_T)N^2M + N^3) = \mathcal{O}(\log(N)N^2M + N^3). \tag{8.2}$$

Here the additional summand N^3 accounts for the singular value decomposition of $\mathbf{\Sigma}$ prior to growing T which is needed for applying computational tricks of [34] during splitting (see Chapter 3.2).

Worst case. In the *worst case*, a tree T is a chain of $N-1$ inner nodes. Since the expected depth of any node t_C along this chain is $\mathcal{O}(N)$, we find that we require

$$c_T^{\text{worst}}(N, M) \in \mathcal{O}(\langle d(t_C) \rangle N^2M + N^3) = \mathcal{O}(N^3M) \tag{8.3}$$

steps to build T.

Note, that the optimal tree depth for prediction tasks we consider in this work is always below $\log(N)$ (see Figure 6.3) and therefore runtime for building a single tree stays within bounds of the *balanced case*.

Further Considerations. Overall, the runtime of our mixed RF scales linearly with the number of genetic features, whereas - owing to the singular value decomposition of $\mathbf{\Sigma}$ - we require a qubic number of computations in the number of samples N. Thus our method is advantageous in scenarios where $M \gg N$. Better scalability w.r.t. large N can be obtained using a low rank approximation of the relationship matrix ($\mathbf{\Sigma}$) which allows to apply additional computational tricks introduced by Lippert and others [34].

In order to avoid the *worst case* scenario in feature selection tasks, adjusting the model complexity (tree depths) may be considered here as well.

Measured runtime. Here, we restrict our runtime evaluation to the mouse phenotype prediction, which is computationally the most demanding task considered in this work (see Chapter 6.3). Our methods run on data with 1940 individuals for each of which we have 12,545 genetic features (SNPs). For a given phenotype the current mixed RF implementation takes about 4400 seconds for a full five-fold (randomized) cross-validation on a single Intel© Xeon© L5420 core (Figure E.5 in Appendix E). Thus, a typical runtime of 15–30 minutes is to be expected if one intends to train a single model on a dataset of comparable size. Note, that the relatively large difference in mean runtime between LASSO and LMM LASSO is a result of the wider range needed to fit the shrinkage parameter in the case of LMM LASSO.

In the case of our mixed RF implementation, the memory requirement never exceeded 2 gigabyte of RAM.

8.2. Limiting Cases

Mixed random forests with optimal depth of 0. When fitted for prediction tasks, the lower limit for the optimal forest depth is 0 corresponding to an ensemble of means, each fitted to a bootstrap sample. Provided little variance between these means, this model closely resembles the Best Linear Unbiased Predictor (BLUP), a widely used approach for genomic prediction which does omit a feature selection step (see Chapters 3.1 and 6.3).

Mixed random forests in the limit of $\delta \to \infty$. In this limit our splitting objective (Equation 5.1) turns into a linear model with i.i.d. Gaussian noise which (in turn) is an equivalent objective used for the standard random forest (i.e. Equation (4.3)). So we have,

Proposition 2. *In the limit of $\delta \to \infty$ the mixed random forest is consistent with the random forest.*

A sketch of this proof is given in Appendix C.

9

Conclusions

In this thesis we presented a new method for QTL mapping and phenotype prediction which can be regarded as an extension to both, the popular linear mixed models as well as random forests. By combining the strengths of random effect modelling with tree-based models our approach is capable of producing robust predictions and phenotype associations over a wide range of potential use cases, and particularly in scenarios where both, polygenic- and non-linear effects like epistasis co-occur.

QTL mapping

To show our method's capabilities in mapping quantitative traits, we compared the proposed mixed random forest to four state of the art methods, each of which has previously been shown to be superior in their respective modelling domain (epistasis, multiple additive- and/or polygenic effects) [41, 50, 53, 65]. Results on simulated QTL data show that our mixed RF is not only the most robust across these different domains but also outperforms all competitors when multiple epistatic- and polygenic effects are present (Chapter 6.1). On real QTL data from heterogeneous stock mice, we show that mixed RF recovers more effector-target gene relationships than any of the other methods (Chapter 6.2).

Instead of mapping single nucleotide polymorphisms, we also considered whole genomic regions in Chapter 7. The idea exploited here is that sensitivity for rare genomic variants can be increased if other rare and/or common genetic variants in local linkage are considered as well. We compared our mixed random forest to the vanilla random forest and two marginal linear models, SKAT [42] and its extension to account for polygenic and genetic background LMM SKAT [63]. On simulated data we found that LMM SKAT was the only method that outperformed our mixed random forest, which only emphasises the versatility of our approach.

Phenotype prediction and learning of trait complexity

As in RF, LASSO and LMM LASSO, the proposed mixed RF is capable of modelling more than a single genetic feature at a time which renders it useful for prediction tasks. On held-out mouse phenotypes, we find that mixed RF is the best predictor for the majority of phenotypes considered. Moreover, it is the only method that is consistently among the top predictors across the entire range of trait complexity (see Chapter 6.3).

An appealing feature of mixed RF is that model complexity can be easily adjusted to the data: since only a subset of samples is used to build each tree, the remaining, so called "out of bag samples" are left over to fit the optimal depth of the trees. This mechanism to control the model complexity is implemented into our method and does not require further input by the user. Importantly, the learned tree depth provides insight into the relevance of the fixed genetic effects in relation to the polygenic background - or in other words - this approach can be used to quantify the complexity of the trait. In principle, tree depth can be adjusted for the vanilla RF as well. However, its interpretability is limited because relative contributions of direct genetic effects and polygenic background to the overall trait complexity cannot be disentangled.

Although it may seem trivial that an optimal model should account for the true number of (genetic) factors contributing to trait variation, there is a scarcity of examples actually performing such model adjustment. Methods such as bagging create an additional computational overhead that in the past may often not have been deemed necessary. Our results question this view - at least when it comes to explaining trait variation.

Limitations

Mixed RF also has its limitations most of which are common among random bagging approaches. First of all, computational demand is generally larger when compared to alternative (linear) methods (see Chapter 8).

While our mixed RF adds interpretability to the model by dissecting trait variance into direct genetic and polygenic contributions, interpretation of the RF component remains a challenge. For example, while RF and mixed RF account for epistatic interactions, it is non-trivial to determine which of the markers considered significant are in epistasis with each other (if any). We have recently proposed a method for extracting epistatic interactions from random forests [49] that is readily applicable to mixed RF.

10

Future work

Our future work is divided into two main directions: further optimization of the presented mixed random forest algorithm and exploiting new use cases in which different sources of confounding than polygenic effects are captured by our random effect model. Here, we present a brief overview of potential areas of future work.

Low rank population structure

Our model faces the same limitation as linear models in that runtime increases cubically with the number of samples. In analogy to LMM [34], computational tricks combined with low rank approximations to the population structure covariance can be used to scale the mixed RF to even larger cohort sizes. In future work, we may update and refine the provided implementation accordingly.

Variable importance on a selected subset of features

In this work we use random effect modelling to account for population structure and genetic background. In future work, we plan to consider large scale data sets (as, for instance, obtained from human cancer cell lines) where features are additionally categorized. Such additional annotation usually specifies the type of mutation/variant that is indicated by the particular feature.

Our idea is to consider a subset/category of features that is of particular interest to the study (like deletions and insertions in developing cancer) using the random forest component of our model, whereas remaining variables - which may also explain a significant fraction of (phenotypic) variation - are captured by the random effect term. This way, our method can help to greatly reduce runtime by computing variable importances for features that are in focus of a particular study.

Batch effect correction

Batch effects are another source of confounding which can be dealt with using a random effect term (e.g. [26, 36]). Therefore our method can be readily applied to scenarios where strong batch effects are present (e.g. analysis of gene expression data).

Non-linear random effect models

As introduced in Chapter 7, random effect models like SKAT and LMM SKAT are state of the art when it comes to mapping of whole genomic regions to phenotypes. However, they are built on the notion that genomic features influence phenotypes in a linear fashion. In future work, we can utilize our mixed random forest to construct a random effect term along the lines of [17] while also accounting for (genetic) background variation. As well as LMM SKAT, the resulting model will contain one random effect term modelling genetic features considered within a specified genomic region, while additionally allowing for non-linear effects. The second random effect models features that are not tested for association.

Part III.

Appendix

Appendix A

Notation used throughout this work

Symbol	Meaning
N	the number of samples
M	number of features (variables)
$a \in \mathbb{R}$	a scalar value
$\mathbf{a} \in \mathbb{R}^N$	a column vector with n rows
$\mathbf{A} \in \mathbb{R}^{N \times M}$	a matrix with N rows and M columns
$[\mathbf{a}]_i$ or a_i	the ith entry of vector $\mathbf{a}$
$[\mathbf{A}]_{ij}$ or a_{ij}	entry in the ith row and jth column of matrix $\mathbf{A}$
$[\mathbf{A}]_{i:}$	the ith row of matrix $\mathbf{A}$
$\mathbf{A}^{\mathrm{T}}$	is the transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	denotes the inverse of matrix $\mathbf{A}$
$\lvert\mathbf{A}\rvert$	the determinant of matrix $\mathbf{A}$
$\mathbf{I}_d$	the identity matrix of dimension d. If no subscript is given the dimension is implied by the context
$\mathbf{1}$	column vector where all entries are one with dimensionality implied by the context
$\mathbf{0}$	depending on the context either a matrix or a column vector where all entries a zero
$[\mathbf{A}, \mathbf{B}]$	horizontal concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$
$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}$	vertical concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathcal{N}\left(y \,\middle|\, \mu, \sigma^2\right)$	is the univariate normal distribution with mean μ and variance σ^2
$\mathcal{N}\left(\mathbf{y} \,\middle|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$	denotes the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

Appendix B

Mathematical Background

Here we summarized some rules, derivations and theorems that are used throughout this thesis. They most of them can also be found for instance in [51].

B.1. Realized Relationship matrix

The **R**ealized **R**elationship **m**atrix [23] is constructed by taking the outer product genetic features considered, i.e.

$$\mathbf{\Sigma}_{\mathrm{RRM}} = \tilde{\mathbf{X}}^{\mathbf{T}}\tilde{\mathbf{X}}. \tag{B.1}$$

Here $\tilde{\mathbf{X}}$ is the standardized version of $\mathbf{X}$ such that each genetic feature (i.e. matrix column) has mean zero and a standard deviation of one. This adjustment gives more weight to variants having low frequencies in the study population which is a commonly made assumption.

B.2. Gaussian Identities

In the following we are given a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ in N dimensions

$$\mathcal{N}\left(\mathbf{x} \,|\, \boldsymbol{\mu}, \mathbf{\Sigma}\right) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{B.2}$$

where $Z = (2\pi)^{N/2}\left|\mathbf{\Sigma}\right|^{1/2}$ is the normalizing constant. Let $\mathbf{x}$ and $\mathbf{y}$ be jointly Gaussian distributed

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^{\mathrm{T}} & \mathbf{B}\end{bmatrix}\right) \tag{B.3}$$

then the marginal distribution of $\mathbf{x}$ and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ are both Gaussian:

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \mathbf{A}) \text{ and} \tag{B.4}$$

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}\left(\boldsymbol{\mu}_x + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \boldsymbol{\mu}_y), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^{\mathrm{T}}\right). \tag{B.5}$$

B.3. Matrix reformulations

Let an invertible $N \times N$ matrix $\mathbf{A}$ and its inverse $\mathbf{A}^{-1}$ be partitioned to

$$\mathbf{A} = \begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix} \text{ and } \quad \mathbf{A}^{-1} = \begin{bmatrix} \tilde{\mathbf{P}} & \tilde{\mathbf{Q}} \\ \tilde{\mathbf{R}} & \tilde{\mathbf{S}} \end{bmatrix}, \tag{B.6}$$

then we have for the submatrices of the inverse $\mathbf{A}^{-1}$

$$\begin{aligned} \tilde{\mathbf{P}} &= \mathbf{P}^{-1} + \mathbf{P}^{-1}\mathbf{Q}\mathbf{M}\mathbf{R}\mathbf{P}^{-1}, \\ \tilde{\mathbf{Q}} &= -\mathbf{P}^{-1}\mathbf{Q}\mathbf{M}, \\ \tilde{\mathbf{R}} &= -\mathbf{M}\mathbf{R} - \mathbf{P}^{-1} \text{ and} \\ \tilde{\mathbf{S}} &= \mathbf{M}, \end{aligned} \tag{B.7}$$

where we defined

$$\mathbf{M} = (\mathbf{S} - \mathbf{R}\mathbf{P}^{-1}\mathbf{Q})^{-1}. \tag{B.8}$$

Appendix C

Proofs

We start with an equivalent representation of the optimization in Equation (4.3) where we changed basis

$$\mathrm{LL}(\mathbf{y}(t) \,|\, \boldsymbol{\beta}, \sigma_v^2, \mathbf{x}_j) = \log \mathcal{N}\left(\begin{bmatrix} \mathbf{y}(t_1) \\ \mathbf{y}(t_2) \end{bmatrix} \middle| \, \beta_L \begin{bmatrix} \mathbf{1}(t_1) \\ \mathbf{0}(t_2) \end{bmatrix} + \beta_2 \begin{bmatrix} \mathbf{0}(t_2) \\ \mathbf{1}(t_2) \end{bmatrix}, \sigma_v^2 \mathbf{I} \right). \tag{C.1}$$

Taking the derivative w.r.t. $\boldsymbol{\beta}$ and setting it to zero we obtain

$$\hat{\boldsymbol{\beta}} = \left(\frac{1}{N(t_1)} \sum_{i \in t_1} y_i, \frac{1}{N(t_2)} \sum_{i \in t_2} y_i \right)^{\mathrm{T}} = (\bar{y}(t_1), \bar{y}(t_2))^{\mathrm{T}}$$

and doing the analogue for σ_v^2 gives us the following:

$$\hat{\sigma}_v^2 = \frac{1}{N(t)} \left(\sum_{i \in t_1} (y_i - \bar{y}(t_1))^2 + \sum_{i \in t_2} (y_i - \bar{y}(t_2))^2 \right).$$

Plugging the results for $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}_v^2$ back into (C.1) we have

$$\begin{aligned} \mathrm{LL}(\mathbf{y}(t) | \hat{\boldsymbol{\beta}}, \hat{\sigma}_v^2, \mathbf{x}_j) &= \frac{N}{2} \log(\hat{\sigma}_v^{-2}) - \frac{N}{2} - N \\ &= -\frac{N}{2} \log\left(\mathrm{R}(t_1) + \mathrm{R}(t_2)\right) + \text{const.} \,. \end{aligned} \tag{C.2}$$

Noting that $\mathrm{R}(t_0)$ is constant and that the log is a strictly monotonic increasing function we have

$$\begin{aligned} \underset{j \in \{1,\dots M\}}{\operatorname{argmax}} \, \mathrm{LL}(\hat{\boldsymbol{\beta}}, \hat{\sigma}_v^2 | \mathbf{x}_j) &= \underset{j \in \{1,\dots M\}}{\operatorname{argmax}} \, -\log\left(\mathrm{R}(t_1) + \mathrm{R}(t_2)\right) \\ &= \underset{j \in \{1,\dots M\}}{\operatorname{argmax}} \, -(\mathrm{R}(t_1) + \mathrm{R}(t_2)) \\ &= \underset{j \in \{1,\dots M\}}{\operatorname{argmax}} \, \Delta\mathrm{R}'(\mathbf{x}_j, t), \end{aligned} \tag{C.3}$$

provided $\mathrm{R}(t_1) + \mathrm{R}(t_2) > 0$. In case $\mathrm{R}(t_1) + \mathrm{R}(t_2) = 0$, mixed random forest directly returns the corresponding index $\hat{j}$ avoiding an evaluation of $\log(0)$. □

Before getting to prove Proposition 2 we require some general properties that hold for least squares optimization in linear models. First, we note that

$$\mathrm{LL}(\hat{\boldsymbol{\beta}}_1, \hat{\sigma}^2 | \mathcal{B}_1) = \mathrm{LL}(\hat{\boldsymbol{\beta}}_2, \hat{\sigma}^2 | \mathcal{B}_2), \tag{C.4}$$

provided $\mathcal{B}_1$ and $\mathcal{B}_2$ are each bases of a common euclidean space V. Secondly we need:

Lemma 1. *Let W and $U_j \in \{U_1, \cdots, U_M\}$ be euclidean spaces such that $W \perp U_j$, $\forall j \in \{1, \cdots, M\}$. Let further $\mathcal{B}$ and $\mathcal{C}_j$ denote bases of W and U_j. Then*

$$\underset{j \in \{1,\cdots,M\}}{\operatorname{argmax}} \; LL(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2 | \, \langle \mathcal{B}, \mathcal{C}_j \rangle) = \underset{j \in \{1,\cdots,M\}}{\operatorname{argmax}} \; LL(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2 | \mathcal{C}_j).$$

Sketch proof of Proposition 2

For any given node t in T and predictor $\mathbf{x}_j$ we find a decomposition of the optimization space given by $\langle C, \mathbf{x}_j \rangle$ in to a fixed part W and a variable part U_j such that $\mathbf{x}_j \in U_j$ and $W \perp U_j$. Construction of W follows by subtracting the last column $\mathbf{x}_p$ of $\mathbf{C}$ from all previous columns. We denote the resulting matrix as $\tilde{C}$ and set $W = \left\langle \tilde{C}_{-\mathbf{x}_p} \right\rangle$ and $U_j = \langle \mathbf{x}_p, \mathbf{x}_j \rangle$. Now $W \perp U_j$ follows simply from the property that all entries in W are zero where the basis vector entries of U_j are one. Furthermore, $\left\langle \tilde{W}, U_j \right\rangle = \langle C, \mathbf{x}_j \rangle$. We thereby found a decomposition such that Lemma 1 allows us to solely consider the subspaces $\{U_1 \dots U_j\}$ for least squares optimization in a linear model. Finally, we identify $\{U_1 \dots U_j\}$ to be the spaces considered by the random forest splitting optimization in equation (4.3). □

Appendix D

Tutorial on how to use mixed random forests

This tutorial shows how to use mixed random forests (mixed RF) for feature selection and prediction. The reader may benefit from having some background on Gaussian process prediction and the general use of the random forest as provided by the python scikit-learn package. However, knowledge of neither is required to execute the steps of this tutorial. All source files for the following examples can be found in the *examples* directory of the mixed RF module.

D.1. Installation

Mac OS, Linux and other Unix based systems. In order to install, the mixed RF requires:

- an installed C++ compiler
- python (2.7) with the following modules installed (do *not* use python 3 or beyond!):
  - numpy (1.7.1)
  - scipy (0.13.0)
  - cython (0.19.2)
  - matplotlib (1.3.1)

Numbers in brackets indicate the versions for which this tutorial was used to generate the following results. Older and newer versions of these packages may work as well.

Installing the mixed random forest package

- extract the files provided with "mixed_rf.zip" and change to the generated "mixed_rf" directory

- make sure that *python 2.7* is loaded when calling "python" from the terminal by running
  ```
  python --version
  ```

- build the mixed_rf-package
  ```
  python setup.py build
  ```

- install the package running
  ```
  python setup.py install --user
  ```

  or
  ```
  python setup.py install
  ```

  for a global installation (requires administrator rights)

D.2. Examples

For each example we require the loaded mixed RF module, some helper functions, the scipy library and matplotlib for plotting.

```
from mixed_forest.MixedForest import Forest as LMF
import mixedForestUtils as utils
import scipy as SP
import pylab as PL
```

D.2.1. Example 1: Recovering a single fixed effect

Please see the source file *examples/tutorial.py* in the module's directory.

At first we need to create some data.

```
SP.random.seed(43)
n_sample = 100
X = SP.empty((n_sample,2))
X[:,0] = SP.arange(0,1,1.0/n_sample)
X[:,1] = SP.random.rand(n_sample)
noise = SP.random.randn(n_sample,1)*.05
y_fixed = (X[:,0:1] > .5)*.5
y_fn  = y_fixed +  noise
```

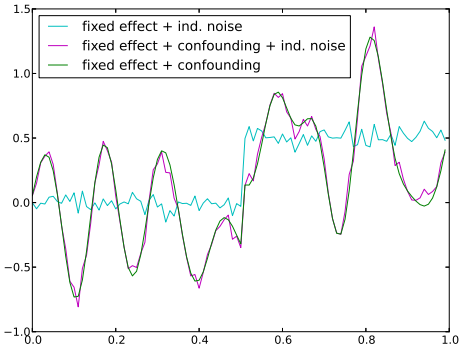


Figure D.1.: Components of simulated data.

Here we consider a simulated data set of size 100. For each sample we simulate features in 2D where the first dimension is sampled from a grid on $[0, 1]$. The second dimension contains random samples from the uniform distribution on the interval $]0, 1[$. We combine features for all samples into the matrix $\mathbf{X}$. The fixed effect $\mathbf{y}_{\text{fixed}}$ shall only be affected by the first feature dimension. In addition, we add some Gaussian noise to obtain the simulated observation $\mathbf{y}_{\text{fn}}$ The second part of the simulation adds structured noise to the observations $\mathbf{y}_{\text{fn}}$. The sample become connected (i.e. correlated) through the simulated covariance.

```
kernel = utils.getQuadraticKernel(X[:,0], d=0.0025)
y_conf = .5*SP.random.multivariate_normal(\
SP.zeros(n_sample), kernel)
y_conf = y_conf.reshape(-1,1)
y_tot = y_fn + y_conf
```

We can now visualize our simulated data (Figure D.1).

```
PL.plot(X[:,0], y_fn, 'c')
PL.plot(X[:,0], y_conf+y_fn, 'm')
PL.plot(X[:,0], y_conf + y_fixed, 'g')
PL.show()
PL.close()
```

Next, we divide our data into training- and test sample.

```
training_sample = SP.zeros(n_sample, dtype='bool')
training_sample[SP.random.permutation(n_sample)\
```

```
  [:SP.int_(.66*n_sample)]] = True
test_sample = ~training_sample
X_train = X[training_sample]
y_train = y_fn[training_sample]
```

We proceed by fitting the standard random forest to the training sample. We can do so using our mixed RF module with the identity as the covariance matrix (i.e. setting kernel='iid').

```
random_forest = LMF(kernel='iid')
random_forest.fit(X[training_sample],\
  y_tot[training_sample])
response_rf = random_forest.predict(X[test_sample])
```

For fitting the mixed RF we need to pick the rows and columns of the covariance according to the training sample indexes. For prediction we need to use the cross covariance between training and test sample.

```
kernel_train = kernel[SP.ix_(training_sample,\
  training_sample)]
kernel_test =  kernel[SP.ix_(test_sample,\
  training_sample)]
lm_forest = LMF(kernel=kernel_train)
lm_forest.fit(X[training_sample],y_tot[training_sample])
response_lmf = lm_forest.predict(X[test_sample],\
  k=kernel_test)
```

Finally, we plot the results of our prediction.

```
PL.plot(X[:,0:1], y_fixed + y_conf, 'g')
PL.plot(X[test_sample,0:1], response_lmf, '.b-.')
PL.plot(X[test_sample,0:1], response_rf, '.k-.')
PL.ylabel('predicted y')
PL.xlabel('first dimension of X')
PL.show()
PL.close()
```

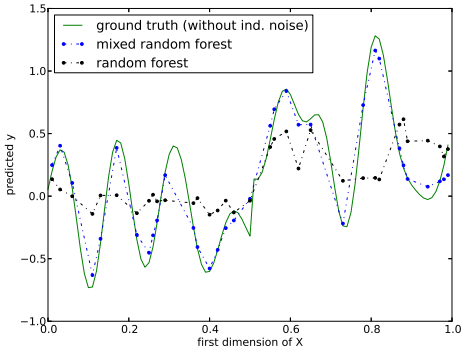


Figure D.2.: Prediction of mixed random forest and random forest on the test sample.

D.2.2. Example 2: Recovering interactions and evaluating feature importances

In the second toy example we predict held-out simulated data when the phenotype is the result of an interaction between two features and the random effect. Now our feature vector is a sorted array of integers ranging from 0 to $2^8 - 1 = 255$. We can easily convert this into an eight-dimensional feature vector using binary encoding of each integer, i.e. $0 \rightarrow (0,0,0,0,0,0,0,0)$, $1 \rightarrow (0,0,0,0,0,0,0,1)$, ... and $255 \rightarrow (1,1,1,1,1,1,1,1)$.

```
SP.random.seed(42)
n_samples=2**8
x = SP.arange(n_samples).reshape(-1,1)
X = utils.convertToBinaryPredictor(x)
```

Our simulated fixed effect shall be an interaction of the first and third feature dimension.

```
y_fixed = X[:,0:1] * X[:,2:3]
```

Finally, like in the previous example, we simulate confounding effects by adding a sample from a multivariate Gaussian with a squared quadratic covariance function.

```
kernel=utils.getQuadraticKernel(x, d=200)
y_conf = y_fixed.copy()
y_conf += SP.random.multivariate_normal(\
```

```
  SP.zeros(n_samples), kernel).reshape(-1,1)
y_conf += .1*SP.random.randn(n_samples,1)
```

In addition, we also add a small amount of independent Gaussian noise.

After splitting, we fit both, the mixed RF and the random forest on the training sample. The test sample is used obtain predictions.

```
(training, test) = utils.crossValidationScheme(\
  2, n_samples)
lm_forest = LMF(kernel=kernel[SP.ix_(\
  training, training)])
lm_forest.fit(X[training], y_conf[training])
response_tot = lm_forest.predict(X[test],\
  kernel[SP.ix_(test,training)])
# make random forest prediction for comparison
random_forest = LMF(kernel='iid')
random_forest.fit(X[training], y_conf[training])
response_iid = random_forest.predict(X[test])
```

So far everything is analog to our previous example. In addition, the mixed RF allows us to predict the fixed effect only. All we need to do is dropping the cross covariance from the parameters of the prediction function, i.e.

```
response_fixed = lm_forest.predict(X[test]).
```

The results of all predictions and ground truth can be observed in the upper panel of Figure D.3. To visualize the whole binary predictor matrix $\mathbf{X}$ in a single dimension we just need to plot the prediction against the original (decimal) feature vector $\mathbf{x}$.

```
PL.plot(x, y_fixed, 'g--')
PL.plot(x, y_conf, '.7')
PL.plot(x[test], response_tot, 'r-.')
PL.plot(x[test], response_fixed, 'c-.')
PL.plot(x[test], response_iid, 'b-.')
PL.title('prediction')
PL.xlabel('genotype (in decimal encoding)')
PL.ylabel('phenotype')
PL.show()
PL.close()
```

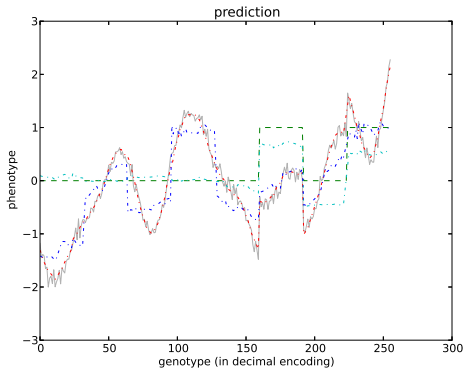


Figure D.3.: Results for a simulated data set with a single interaction

Finally, we evaluate the feature importances which are automatically computed while fitting the forests.

```
feature_scores_lmf = lm_forest.log_importance
feature_scores_rf = random_forest.log_importance
```

These scores are plotted in Figure D.4. We generate this plot by:

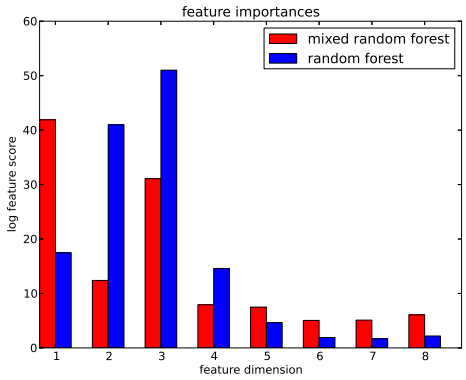


Figure D.4.: Variable importances as computed by random forest and mixed random forest

```
n_predictors = X.shape[1]
PL.bar(SP.arange(n_predictors),\
  feature_scores_lmf, .3, color='r')
PL.bar(SP.arange(n_predictors)+.3,\
```

```
    feature_scores_rf, .3, color='b')
PL.title('feature importances')
PL.xlabel('feature dimension')
PL.ylabel('log feature score')
PL.xticks(SP.arange(n_predictors)+.3,\
  SP.arange(n_predictors)+1)
PL.legend(['mixed random forest', 'random forest'])
PL.show().
```

Appendix E

Supplementary Figures and Tables

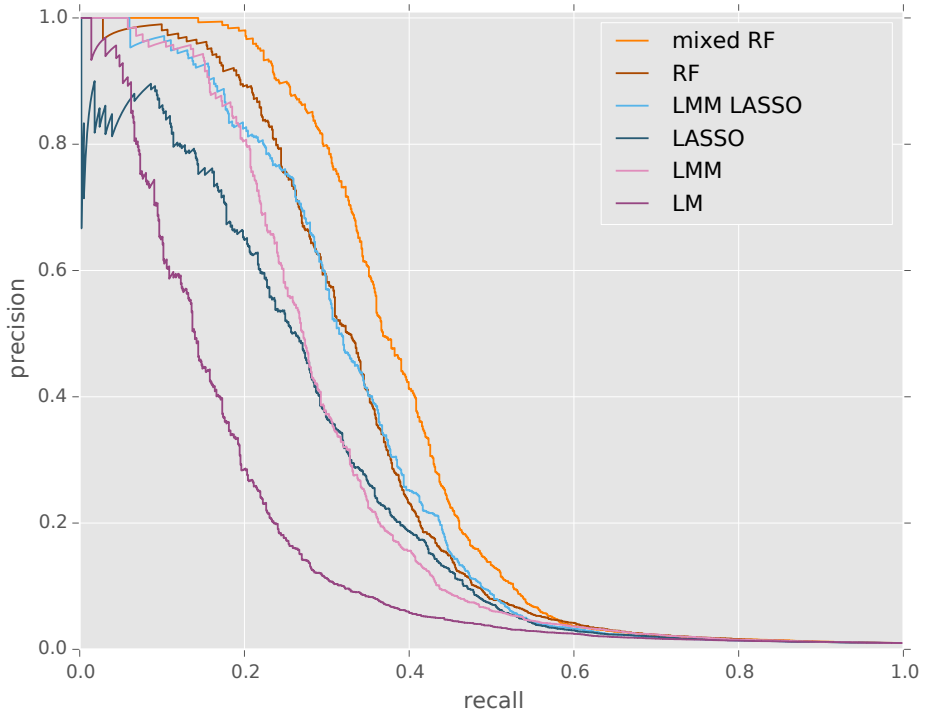


Figure E.1.: Precision-recall curves for alternative methods in our baseline simulation setting indicated by the asterix in Figure 6.1 where variation of simulated traits is caused by three linear additive-, three pairwise epistatic genetic effects, population structure and independent noise. Relative contributions of fixed genetic-, population effect and independent Gaussian noise to the total trait variance are split into 0.375:0.5:0.125.

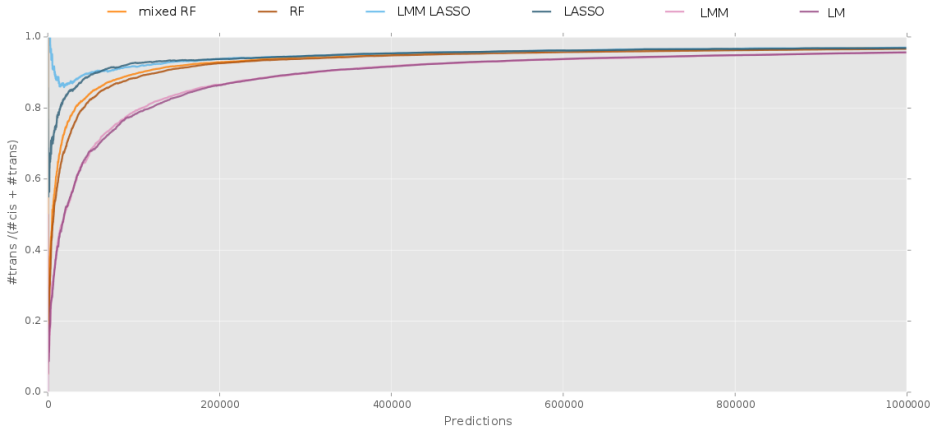


Figure E.2.: Fraction of discovered pathway-member genes in *trans* in dependence of the feature importance rank. Overall, the fraction of *cis* annotated SNPs was relatively low, however, as expected the highest ranking predictors are strongly enriched for *cis* associations.

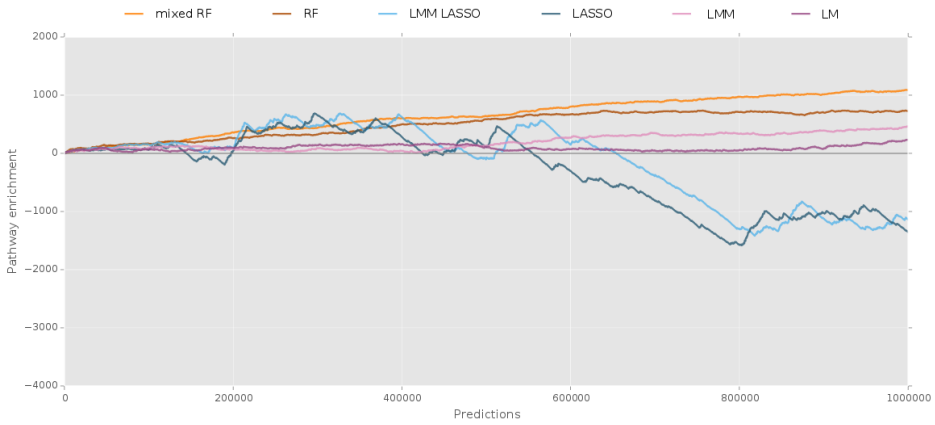


Figure E.3.: Concordance of detected expression QTLs using alternative methods. The experimental setup is identical to the analysis shown in Figure 6.2. However, instead of applying alternative association methods to individual chromosomes, they were applied to genome-wide markers. In line with the hypothesis of proximal contamination ([35]), the performance of all mixed-model based approaches decreased.

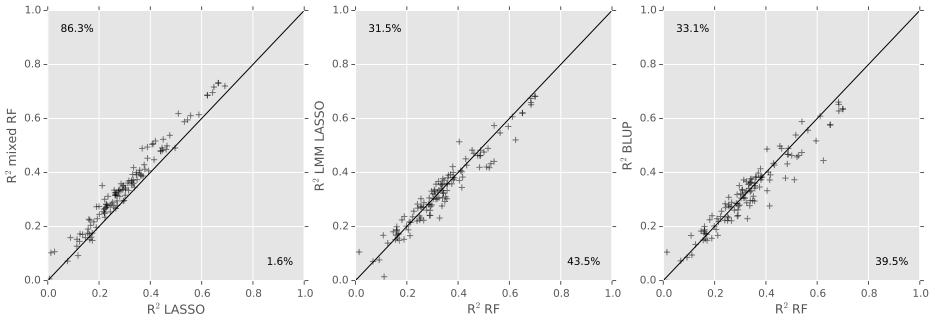


Figure E.4.: Additional correlation plots of the relative out-of-sample prediction accuracy when using alternative methods (see Figure 6.3(a)). Shown is the out-of-sample correlation coefficient, for different pair-wise comparisons of prediction methods.

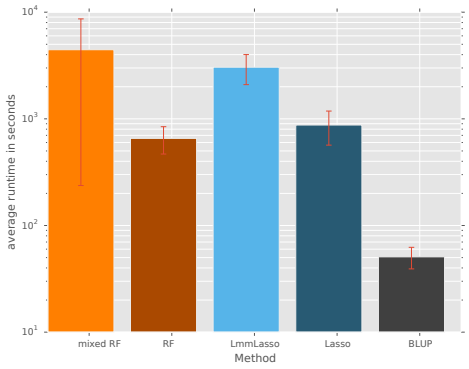


Figure E.5.: Empirical runtime for the mixed RF and alternative methods when applied for prediction of mouse phenotypes. Each single prediction experiment considers all genome-wide 12,545 markers for a total of 1,940 individuals, employing five-fold cross validation for out-of-sample prediction. The runtimes as shown correspond to the average time complexity across all 124 phenotypes. Errobars correspond to the empirical variance across phenotypes.

Table E.1.: Complete list of results for mouse phenotype prediction as shown in Figure 6.3. For each phenotype the best predicting method (out of sample correlation coefficient) is marked in bold font.

	mfex	rfsci	LmmLasso	Lasso	BLUP
AdrenalMeanWeight	**0.62 ±4.4e-03**	0.62 ±3.8e-03	0.52 ±5.5e-03	0.51 ±5.3e-03	0.44 ±5.6e-03
Context	**0.23 ±2.9e-03**	0.21 ±3.7e-03	0.22 ±5.0e-03	0.16 ±7.7e-03	0.23 ±3.1e-03
Anx	0.33 ±3.5e-03	0.31 ±4.9e-03	0.33 ±3.5e-03	0.29 ±2.3e-03	**0.34 ±4.2e-03**
Freeze	**0.28 ±6.1e-03**	0.25 ±4.9e-03	0.28 ±7.2e-03	0.24 ±7.7e-03	0.28 ±6.1e-03
Explore	**0.37 ±5.6e-03**	0.32 ±5.7e-03	0.37 ±5.5e-03	0.32 ±6.1e-03	0.37 ±5.8e-03
Biochem.Albumin	0.24 ±2.9e-03	**0.25 ±2.7e-03**	0.23 ±3.5e-03	0.20 ±2.8e-03	0.23 ±3.3e-03
Biochem.ALP	**0.59 ±3.3e-03**	0.56 ±2.4e-03	0.55 ±4.9e-03	0.53 ±3.8e-03	0.56 ±3.7e-03
Biochem.ALT	0.15 ±1.3e-02	**0.19 ±8.9e-03**	0.15 ±1.2e-02	0.11 ±1.0e-02	0.16 ±1.2e-02
Biochem.AST	0.16 ±8.0e-03	0.16 ±3.9e-03	**0.19 ±7.3e-03**	0.17 ±8.2e-03	0.18 ±7.2e-03
Biochem.Calcium	0.25 ±8.7e-03	**0.27 ±8.3e-03**	0.22 ±8.0e-03	0.22 ±8.3e-03	0.22 ±8.9e-03
Biochem.Chloride	0.34 ±5.4e-03	**0.35 ±3.9e-03**	0.29 ±5.5e-03	0.29 ±3.3e-03	0.30 ±4.6e-03
Biochem.Creatinine	**0.17 ±7.1e-03**	0.16 ±8.0e-03	0.16 ±7.8e-03	0.13 ±8.8e-03	0.15 ±6.3e-03
Biochem.Glucose	**0.26 ±3.9e-03**	0.25 ±5.0e-03	0.24 ±3.6e-03	0.22 ±3.4e-03	0.24 ±3.8e-03
Biochem.HDL	**0.61 ±7.2e-03**	0.60 ±5.6e-03	0.57 ±5.0e-03	0.56 ±6.3e-03	0.52 ±5.4e-03
Biochem.LDL	**0.37 ±3.7e-03**	0.31 ±3.4e-03	0.36 ±4.2e-03	0.33 ±5.8e-03	0.37 ±4.1e-03
Biochem.Phosphorous	**0.22 ±2.8e-03**	0.22 ±1.6e-03	0.22 ±3.9e-03	0.18 ±3.3e-03	0.22 ±3.5e-03
Biochem.Potassium	0.10 ±1.2e-02	**0.11 ±1.4e-02**	0.01 ±1.2e-02	0.01 ±1.4e-02	0.09 ±1.4e-02
Biochem.Sodium	0.33 ±6.3e-03	**0.34 ±6.1e-03**	0.28 ±5.5e-03	0.26 ±7.2e-03	0.28 ±5.4e-03
Biochem.Tot.Cholesterol	**0.48 ±6.0e-03**	0.48 ±6.4e-03	0.46 ±5.0e-03	0.45 ±5.6e-03	0.38 ±4.6e-03
Biochem.Tot.Protein	**0.18 ±7.1e-03**	0.17 ±6.1e-03	0.15 ±6.9e-03	0.17 ±1.8e-03	0.15 ±7.1e-03
Biochem.Triglycerides	**0.40 ±5.3e-03**	0.40 ±5.6e-03	0.37 ±4.8e-03	0.36 ±1.6e-03	0.33 ±6.7e-03
Biochem.Urea	0.31 ±6.0e-03	0.32 ±7.9e-03	**0.33 ±7.3e-03**	0.24 ±8.2e-03	0.32 ±7.1e-03
BurrowedPelletWeight	**0.36 ±1.7e-03**	0.34 ±2.6e-03	0.35 ±1.1e-03	0.31 ±2.2e-03	0.36 ±1.9e-03
Imm.PctCD4andCD3	**0.39 ±4.8e-03**	0.38 ±5.6e-03	0.37 ±4.1e-03	0.36 ±4.4e-03	0.39 ±4.3e-03
CD4Count	0.35 ±4.3e-03	0.34 ±7.2e-03	0.32 ±2.9e-03	0.29 ±3.7e-03	**0.36 ±4.4e-03**
Imm.PctCD4inCD3	**0.69 ±5.2e-03**	0.65 ±3.9e-03	0.62 ±3.8e-03	0.62 ±3.8e-03	0.58 ±4.6e-03
Imm.PctCD3	0.48 ±7.2e-03	**0.49 ±6.2e-03**	0.46 ±6.8e-03	0.44 ±6.9e-03	0.47 ±8.1e-03
Imm.PctCD8inCD3	**0.73 ±3.4e-03**	0.70 ±1.9e-03	0.68 ±2.8e-03	0.67 ±3.7e-03	0.64 ±3.9e-03
Imm.PctCD8andCD3	**0.72 ±6.6e-03**	0.68 ±6.0e-03	0.66 ±6.7e-03	0.65 ±7.0e-03	0.65 ±5.4e-03
CD8Count	0.33 ±7.3e-03	**0.34 ±5.3e-03**	0.32 ±9.4e-03	0.26 ±9.3e-03	0.33 ±6.4e-03
Context.Mean.Freeze	**0.32 ±4.9e-03**	0.29 ±8.0e-03	0.31 ±4.7e-03	0.30 ±3.3e-03	0.32 ±3.7e-03
Cue.Mean.Freeze.Corrected.D.	**0.35 ±2.2e-03**	0.33 ±3.2e-03	0.23 ±5.3e-03	0.21 ±2.6e-03	0.23 ±5.5e-03
Cue.Freeze.Base12.Increase	**0.01 ±5.6e-03**	-0.02 ±7.2e-03	0.01 ±1.0e-02	0.01 ±6.3e-03	0.01 ±9.6e-03
Cue.Mean.Freeze.Post	0.16 ±7.1e-03	0.11 ±7.7e-03	**0.17 ±6.9e-03**	0.09 ±6.1e-03	0.17 ±6.8e-03
Cue.Mean.Activity.Corrected.D.	**0.02 ±8.9e-03**	0.00 ±1.0e-02	-0.00 ±6.7e-03	-0.03 ±8.5e-03	-0.00 ±7.1e-03
Cue.Mean.Activity.Post	-0.01 ±1.2e-02	-0.02 ±1.1e-02	-0.01 ±7.5e-03	**-0.05 ±1.0e-02**	-0.00 ±6.9e-03
Cue.Activity.Base12.Increase	-0.01 ±6.4e-03	**0.01 ±4.9e-03**	-0.01 ±6.1e-03	-0.01 ±8.8e-03	-0.01 ±5.7e-03
Cue.Raw.Activity.Before.Tone1	**0.26 ±5.0e-03**	0.24 ±2.8e-03	0.22 ±4.5e-03	0.22 ±3.5e-03	0.22 ±4.8e-03
EMO	0.28 ±4.7e-03	0.27 ±5.2e-03	**0.29 ±5.4e-03**	0.25 ±7.5e-03	0.29 ±5.4e-03
EPM.ClosedArmDistance	**0.27 ±1.7e-03**	0.24 ±2.3e-03	0.27 ±2.2e-03	0.26 ±4.3e-03	0.27 ±2.1e-03
EPM.ClosedArmEntries	**0.17 ±6.5e-03**	0.17 ±4.4e-03	0.17 ±5.6e-03	0.16 ±7.7e-03	0.17 ±5.8e-03
EPM.ClosedArmTime	0.29 ±2.5e-03	0.26 ±3.0e-03	0.30 ±2.7e-03	0.25 ±4.2e-03	**0.31 ±3.4e-03**
EPM.JunctionDistance	**0.45 ±1.9e-03**	0.43 ±5.0e-03	0.43 ±2.2e-03	0.39 ±1.3e-03	0.43 ±2.4e-03
EPM.JunctionEntries	0.18 ±5.7e-03	0.16 ±1.1e-02	**0.19 ±2.3e-03**	0.16 ±5.0e-03	0.19 ±2.7e-03
EPM.JunctionTime	**0.20 ±2.7e-03**	0.20 ±2.7e-03	0.20 ±3.2e-03	0.19 ±3.6e-03	0.20 ±3.2e-03
EPM.OpenArmDistance	**0.39 ±3.1e-03**	0.38 ±1.8e-03	0.38 ±6.7e-03	0.35 ±5.2e-03	0.38 ±6.8e-03
EPM.OpenArmEntries	0.36 ±8.7e-03	0.36 ±7.8e-03	**0.38 ±9.1e-03**	0.33 ±7.0e-03	0.38 ±8.9e-03
EPM.OpenArmLatency	0.23 ±5.0e-03	0.19 ±7.0e-03	**0.24 ±5.4e-03**	0.19 ±4.0e-03	0.24 ±5.4e-03
EPM.OpenArmLatencyC.	0.17 ±4.5e-03	0.15 ±8.6e-03	**0.18 ±3.9e-03**	0.14 ±4.4e-03	0.18 ±4.0e-03
EPM.OpenArmTime	**0.34 ±7.0e-03**	0.33 ±5.8e-03	0.34 ±8.9e-03	0.28 ±6.5e-03	0.34 ±8.9e-03
FirstEarHoleArea	**0.39 ±4.8e-03**	0.36 ±4.7e-03	0.39 ±5.9e-03	0.37 ±4.4e-03	0.38 ±4.8e-03
SecondEarHoleArea	0.39 ±7.7e-03	0.38 ±6.7e-03	**0.40 ±8.5e-03**	0.36 ±5.5e-03	0.37 ±8.5e-03
EarHoleMeanArea	**0.45 ±5.1e-03**	0.43 ±5.7e-03	0.45 ±4.3e-03	0.41 ±4.7e-03	0.43 ±4.6e-03
FN.Latency	0.22 ±4.7e-03	0.18 ±4.6e-03	0.22 ±5.0e-03	0.16 ±5.4e-03	**0.23 ±5.2e-03**
FN.LatencyCensored	**0.20 ±4.8e-03**	0.16 ±3.7e-03	0.20 ±4.9e-03	0.17 ±3.4e-03	0.20 ±5.1e-03
FN.PctWtLoss	0.27 ±5.4e-03	**0.29 ±6.2e-03**	0.24 ±5.8e-03	0.22 ±7.9e-03	0.24 ±5.3e-03
Glucose_0	**0.51 ±7.2e-03**	0.51 ±7.2e-03	0.42 ±6.0e-03	0.41 ±4.5e-03	0.37 ±6.4e-03
Glucose_15	**0.32 ±2.7e-03**	0.31 ±6.2e-03	0.30 ±2.0e-03	0.27 ±1.2e-03	0.31 ±3.2e-03

Table E.1.: Complete list of results for mouse phenotype prediction as shown in Figure 6.3. For each phenotype the best predicting method (out of sample correlation coefficient) is marked in bold font.

	mfex	rfsci	LmmLasso	Lasso	BLUP
Glucose_30	0.27 ±3.1e-03	**0.28 ±3.7e-03**	0.26 ±3.2e-03	0.27 ±1.5e-03	0.26 ±3.1e-03
Glucose_75	**0.30 ±5.5e-03**	0.29 ±6.7e-03	0.26 ±5.0e-03	0.22 ±3.7e-03	0.26 ±4.6e-03
Glucose.Slope	0.09 ±3.3e-03	0.09 ±3.0e-03	0.08 ±6.6e-03	**0.12 ±2.5e-03**	0.09 ±5.1e-03
Haem.ALYabs	0.07 ±6.3e-03	0.07 ±6.7e-03	0.07 ±6.1e-03	**0.08 ±6.2e-03**	0.07 ±6.2e-03
Haem.BASabs	**0.11 ±7.8e-03**	0.01 ±7.1e-03	0.11 ±7.9e-03	0.03 ±5.7e-03	0.11 ±7.9e-03
Haem.HCT	0.13 ±9.6e-03	0.13 ±9.2e-03	**0.14 ±9.4e-03**	0.11 ±3.5e-03	0.13 ±8.3e-03
Haem.HGB	0.16 ±1.1e-02	0.16 ±7.6e-03	0.17 ±9.8e-03	0.15 ±7.7e-03	**0.18 ±8.2e-03**
Haem.LICabs	0.15 ±1.3e-02	**0.21 ±1.7e-02**	0.17 ±1.0e-02	0.16 ±8.4e-03	0.17 ±9.5e-03
Haem.MCH	0.49 ±6.6e-03	0.40 ±7.7e-03	**0.51 ±6.7e-03**	0.50 ±5.7e-03	0.49 ±6.8e-03
Haem.MCHC	0.41 ±3.2e-03	**0.42 ±2.1e-03**	0.38 ±5.5e-03	0.36 ±4.8e-03	0.39 ±3.6e-03
Haem.MCV	**0.50 ±4.9e-03**	0.45 ±3.9e-03	0.48 ±5.8e-03	0.47 ±5.0e-03	0.50 ±5.0e-03
Haem.MONabs	0.14 ±8.6e-03	**0.16 ±1.0e-02**	0.15 ±7.8e-03	0.12 ±5.2e-03	0.15 ±7.5e-03
Haem.MPV	0.32 ±1.1e-02	**0.33 ±1.1e-02**	0.33 ±6.5e-03	0.27 ±5.8e-03	0.29 ±5.8e-03
Haem.NEUabs	0.28 ±5.7e-03	**0.29 ±8.6e-03**	0.28 ±6.3e-03	0.23 ±6.5e-03	0.28 ±6.3e-03
Haem.PCT	**0.27 ±4.8e-03**	0.25 ±4.6e-03	0.22 ±4.1e-03	0.19 ±3.0e-03	0.22 ±4.8e-03
Haem.PLT	0.27 ±8.8e-03	**0.29 ±8.7e-03**	0.24 ±9.6e-03	0.20 ±7.6e-03	0.24 ±1.1e-02
Haem.RBC	0.15 ±5.3e-03	**0.17 ±4.8e-03**	0.17 ±4.6e-03	0.17 ±3.1e-03	0.17 ±4.9e-03
Haem.RDW	**0.49 ±1.8e-03**	0.49 ±2.9e-03	0.48 ±2.2e-03	0.45 ±1.9e-03	0.49 ±1.9e-03
Haem.WBC	0.31 ±5.2e-03	**0.33 ±6.3e-03**	0.31 ±4.2e-03	0.26 ±4.2e-03	0.33 ±2.1e-03
Imm.Bcell.size	0.49 ±5.5e-03	**0.52 ±3.0e-03**	0.42 ±6.3e-03	0.39 ±6.2e-03	0.46 ±3.5e-03
Imm.CD4.size	0.52 ±5.6e-03	**0.54 ±3.5e-03**	0.44 ±6.5e-03	0.42 ±6.4e-03	0.47 ±4.2e-03
Imm.CD8.size	0.50 ±5.0e-03	**0.53 ±3.6e-03**	0.43 ±6.0e-03	0.41 ±6.3e-03	0.46 ±3.7e-03
Imm.B220Median	0.35 ±8.7e-03	**0.38 ±1.1e-02**	0.36 ±9.0e-03	0.34 ±7.8e-03	0.35 ±8.8e-03
Imm.CD4XGeoMean	**0.59 ±2.8e-03**	0.54 ±4.3e-03	0.57 ±2.4e-03	0.55 ±2.9e-03	0.59 ±2.1e-03
Imm.CD4YGeoMean	0.29 ±7.7e-03	**0.36 ±8.7e-03**	0.30 ±8.7e-03	0.29 ±8.0e-03	0.29 ±7.6e-03
Imm.CD4inCD3XGeoMean	**0.61 ±4.5e-03**	0.61 ±4.7e-03	0.61 ±4.6e-03	0.59 ±5.3e-03	0.61 ±4.7e-03
Imm.CD4inCD3YGeoMean	0.30 ±7.7e-03	**0.34 ±9.3e-03**	0.31 ±8.8e-03	0.30 ±7.9e-03	0.30 ±7.4e-03
Imm.CD8XGeoMean	0.28 ±8.6e-03	**0.29 ±6.6e-03**	0.28 ±8.7e-03	0.23 ±9.8e-03	0.29 ±8.8e-03
Imm.CD8YGeoMean	0.34 ±4.5e-03	**0.36 ±4.5e-03**	0.33 ±5.6e-03	0.30 ±5.3e-03	0.34 ±4.6e-03
Imm.CD8inCD3XGeoMean	0.28 ±8.6e-03	0.28 ±6.9e-03	0.28 ±8.6e-03	0.23 ±9.9e-03	**0.29 ±8.8e-03**
Imm.CD8inCD3YGeoMean	0.33 ±5.0e-03	**0.35 ±8.6e-03**	0.31 ±4.5e-03	0.28 ±2.4e-03	0.31 ±3.9e-03
Imm.PctB220	**0.49 ±2.3e-03**	0.46 ±5.0e-03	0.47 ±3.0e-03	0.46 ±2.9e-03	0.49 ±3.3e-03
Imm.PctCD3	0.48 ±7.2e-03	**0.49 ±6.2e-03**	0.46 ±6.8e-03	0.44 ±6.9e-03	0.47 ±8.1e-03
Imm.PctCD4	**0.41 ±9.5e-03**	0.41 ±7.4e-03	0.41 ±8.9e-03	0.39 ±6.8e-03	0.40 ±9.8e-03
Imm.PctCD4inCD3	**0.69 ±5.2e-03**	0.65 ±3.9e-03	0.62 ±3.8e-03	0.62 ±3.8e-03	0.58 ±4.6e-03
Imm.PctCD8	**0.72 ±3.4e-03**	0.68 ±5.0e-03	0.67 ±3.4e-03	0.69 ±2.9e-03	0.66 ±3.3e-03
Imm.PctCD8inCD3	**0.73 ±3.4e-03**	0.70 ±1.9e-03	0.68 ±2.8e-03	0.67 ±3.7e-03	0.64 ±3.9e-03
Imm.CD4CD8Ratio	**0.70 ±4.7e-03**	0.68 ±5.4e-03	0.65 ±4.6e-03	0.64 ±5.7e-03	0.63 ±5.6e-03
Insulin_0	0.33 ±4.7e-03	0.30 ±6.7e-03	0.33 ±5.0e-03	0.28 ±3.8e-03	**0.35 ±5.3e-03**
Insulin_15	0.35 ±6.3e-03	0.34 ±5.7e-03	**0.38 ±8.1e-03**	0.33 ±5.6e-03	0.38 ±8.5e-03
Insulin_30	0.37 ±3.3e-03	**0.39 ±2.7e-03**	0.38 ±3.6e-03	0.35 ±3.8e-03	0.38 ±3.7e-03
Insulin_75	0.29 ±3.7e-03	**0.32 ±3.0e-03**	0.30 ±7.4e-03	0.27 ±4.3e-03	0.31 ±6.7e-03
KI67	**0.54 ±5.8e-03**	0.52 ±1.0e-02	0.49 ±3.3e-03	0.48 ±3.4e-03	0.54 ±6.1e-03
OFT.CenterTime	0.27 ±4.7e-03	0.24 ±5.1e-03	**0.28 ±2.6e-03**	0.23 ±5.9e-03	0.28 ±2.9e-03
OFT.Latency	0.25 ±4.7e-03	0.23 ±4.5e-03	**0.26 ±5.1e-03**	0.24 ±5.7e-03	0.26 ±5.2e-03
OFT.Latency.Censored	0.23 ±3.1e-03	0.22 ±2.6e-03	**0.24 ±3.6e-03**	0.22 ±3.8e-03	0.24 ±3.1e-03
OFT.TotalActivity	0.37 ±5.3e-03	0.35 ±3.4e-03	0.37 ±4.8e-03	0.33 ±7.6e-03	**0.38 ±4.2e-03**
Obesity.BMI	**0.42 ±1.9e-03**	0.41 ±4.1e-03	0.34 ±2.7e-03	0.33 ±2.2e-03	0.28 ±2.3e-03
Obesity.BodyLength	**0.34 ±2.7e-03**	0.34 ±2.6e-03	0.33 ±3.3e-03	0.30 ±4.7e-03	0.33 ±3.0e-03
PAS.Ambulatory1	0.31 ±3.1e-03	0.30 ±5.1e-03	**0.32 ±3.6e-03**	0.28 ±7.0e-03	0.32 ±3.3e-03
PAS.Ambulatory6	0.19 ±9.0e-03	**0.21 ±7.5e-03**	0.19 ±9.4e-03	0.16 ±6.5e-03	0.19 ±9.1e-03
PAS.TotalAmbulatory	0.33 ±5.5e-03	0.31 ±2.7e-03	**0.34 ±6.2e-03**	0.32 ±8.8e-03	0.34 ±5.4e-03
PAS.TotalFine	**0.27 ±8.0e-03**	0.26 ±6.4e-03	0.27 ±7.5e-03	0.23 ±8.9e-03	0.27 ±8.0e-03
Pleth.EnhancedDiff	**0.36 ±5.5e-03**	0.34 ±7.1e-03	0.36 ±6.2e-03	0.33 ±5.9e-03	0.36 ±5.9e-03
Pleth.base.BreathFrequency	**0.32 ±5.2e-03**	0.30 ±6.6e-03	0.30 ±5.3e-03	0.26 ±4.9e-03	0.31 ±5.0e-03
Pleth.base.EnhancedPause	0.28 ±2.6e-03	**0.30 ±5.2e-03**	0.28 ±3.0e-03	0.27 ±4.8e-03	0.28 ±2.7e-03
Pleth.base.ExpiratoryTime	**0.32 ±5.5e-03**	0.31 ±4.4e-03	0.32 ±5.7e-03	0.27 ±3.1e-03	0.32 ±5.5e-03
Pleth.base.InspiratoryTime	0.25 ±9.4e-03	0.26 ±7.3e-03	**0.27 ±9.4e-03**	0.22 ±7.9e-03	0.27 ±9.4e-03

Table E.1.: Complete list of results for mouse phenotype prediction as shown in Figure 6.3. For each phenotype the best predicting method (out of sample correlation coefficient) is marked in bold font.

	mfex	rfsci	LmmLasso	Lasso	BLUP
Pleth.base.MinuteVolume	**0.39 ±3.3e-03**	0.36 ±4.3e-03	0.36 ±3.9e-03	0.33 ±3.6e-03	0.35 ±3.4e-03
Pleth.base.TidalVolume	**0.43 ±7.6e-03**	0.41 ±8.2e-03	0.40 ±7.2e-03	0.38 ±4.3e-03	0.37 ±7.0e-03
Pleth.meta.BreathFrequency	0.35 ±5.4e-03	0.33 ±6.7e-03	**0.36 ±4.6e-03**	0.30 ±9.1e-03	0.36 ±4.8e-03
Pleth.meta.EnhancedPause	0.41 ±4.0e-03	0.38 ±3.6e-03	**0.42 ±4.6e-03**	0.38 ±6.4e-03	0.41 ±4.3e-03
Pleth.meta.ExpiratoryTime	**0.36 ±3.3e-03**	0.36 ±5.6e-03	0.36 ±3.2e-03	0.29 ±4.7e-03	0.36 ±3.2e-03
Pleth.meta.InspiratoryTime	**0.40 ±2.9e-03**	0.37 ±4.4e-03	0.39 ±3.0e-03	0.35 ±5.0e-03	0.40 ±3.1e-03
Pleth.meta.MinuteVolume	**0.49 ±4.8e-03**	0.48 ±2.5e-03	0.42 ±6.7e-03	0.37 ±6.0e-03	0.43 ±6.7e-03
Pleth.meta.TidalVolume	**0.52 ±4.5e-03**	0.50 ±6.0e-03	0.48 ±4.9e-03	0.45 ±4.5e-03	0.46 ±5.2e-03

Bibliography

[1] Susanna Atwell, Yu S Huang, Bjarni J Vilhjálmsson, Glenda Willems, Matthew Horton, Yan Li, Dazhe Meng, Alexander Platt, Aaron M Tarone, Tina T Hu, et al. Genome-wide association study of 107 phenotypes in arabidopsis thaliana inbred lines. *Nature*, 465(7298):627–631, 2010.

[2] Gérard Biau. Analysis of a random forests model. *J. Mach. Learn. Res.*, 98888:1063–1095, June 2012.

[3] Peter J Bickel and Kjell A Doksum. Mathematical statistics, volume i, 2001.

[4] Joshua S Bloom, Ian M Ehrenreich, Wesley T Loo, Thúy-Lan Võ Lite, and Leonid Kruglyak. Finding the sources of missing heritability in a yeast cross. *Nature*, 494(7436):234–237, 2013.

[5] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[6] Anne-Laure Boulesteix, Andreas Bender, Justo Lorenzo Bermejo, and Carolin Strobl. Random forest gini importance favours snps with large minor allele frequency: impact, sources and recommendations. *Briefings in Bioinformatics*, 13(3):292–304, 2012.

[7] George EP Box. Non-normality and tests on variances. *Biometrika*, 40(3-4):318–335, 1953.

[8] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[9] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[10] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. 10.1023/A:1010933404324.

[11] Richard P Brent. *Algorithms for minimization without derivatives*. Courier Dover Publications, 2013.

[12] Karl W Broman and Terence P Speed. A model selection approach for the identification of quantitative trait loci in experimental crosses. *Journal of the Royal Statistical Society: Series B (Statistical Method-*

ology), 64(4):641–656, 2002.

[13] Ö. Carlborg and C.S. Haley. Epistasis: too often neglected in complex trait studies? *Nature Reviews Genetics*, 5(8):618–625, 2004.

[14] Örjan Carlborg, Susanne Kerje, Karin Schütz, Lina Jacobsson, Per Jensen, and Leif Andersson. A global search reveals epistatic interaction between qtl for early growth in the chicken. *Genome research*, 13(3):413–421, 2003.

[15] Augustin Cauchy. Méthode générale pour la résolution des systemes déquations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[16] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

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

[18] Evan E Eichler, Jonathan Flint, Greg Gibson, Augustine Kong, Suzanne M Leal, Jason H Moore, and Joseph H Nadeau. Missing heritability and strategies for finding the underlying causes of complex disease. *Nature Reviews Genetics*, 11(6):446–450, 2010.

[19] Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine learning*, 43(3):293–318, 2001.

[20] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[21] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38:367–378, 1999.

[22] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[23] Ben John Hayes, Peter M Visscher, and Michael E Goddard. Increased accuracy of artificial selection by using the realized relationship matrix. *Genetics Research*, 91(01):47–60, 2009.

[24] Ronald R Hocking. A biometrics invited paper. the analysis and selection of variables in linear regression. *Biometrics*, pages 1–49, 1976.

[25] Guo-Jen Huang, Sagiv Shifman, William Valdar, Martina Johannes-

son, Binnaz Yalcin, Martin S Taylor, Jennifer M Taylor, Richard Mott, and Jonathan Flint. High resolution mapping of expression qtls in heterogeneous stock mice in multiple tissues. *Genome research*, 19(6):1133–1140, 2009.

[26] Jong WJ Joo, Jae H Sul, Buhm Han, Chun Ye, and Eleazar Eskin. Effectively identifying regulatory hotspots while capturing expression heterogeneity in gene expression studies. *Genome biology*, 15(4):r61, 2014.

[27] G Joshi-Tope, Marc Gillespie, Imre Vastrik, Peter D'Eustachio, Esther Schmidt, Bernard de Bono, Bijay Jassal, GR Gopinath, GR Wu, Lisa Matthews, et al. Reactome: a knowledgebase of biological pathways. *Nucleic acids research*, 33(suppl 1):D428–D432, 2005.

[28] Hyun Min Kang, Jae Hoon Sul, Susan K Service, Noah A Zaitlen, Sit-Yee Kong, Nelson B Freimer, Chiara Sabatti, and Eleazar Eskin. Variance component model to account for sample structure in genome-wide association studies. *Nature genetics*, 42(4):348–54, April 2010.

[29] Hyun Min Kang, Chun Ye, and Eleazar Eskin. Accurate discovery of expression quantitative trait loci under confounding from spurious and genuine regulatory hotspots. *Genetics*, 180(4):1909–25, December 2008.

[30] Hyun Min Kang, Noah A. Zaitlen, Claire M. Wade, Andrew Kirby, David Heckerman, Mark J. Daly, and Eleazar Eskin. Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709–1723, March 2008.

[31] AB Korol, IA Preigel, and NI Bocharnikova. Linkage between quantitative and marker loci. v. joint analysis of various marker and quantitative traits. *Genetika*, 23(8):1421–1431, 1987.

[32] Bingshan Li and Suzanne M Leal. Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. *The American Journal of Human Genetics*, 83(3):311–321, 2008.

[33] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

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

[35] J. Listgarten, C. Lippert, C.M. Kadie, R.I. Davidson, E. Eskin, and

D. Heckerman. Improved linear mixed models for genome-wide association studies. *Nature Methods*, 9(6):525–526, 2012.

[36] Jennifer Listgarten, Carl Kadie, Eric E Schadt, and David Heckerman. Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences*, 107(38):16465–16470, 2010.

[37] Bo Eskerod Madsen and Sharon R Browning. A groupwise association test for rare mutations using a weighted sum statistic. *PLoS genetics*, 5(2):e1000384, 2009.

[38] Teri A. Manolio, Francis S. Collins, Nancy J. Cox, David B. Goldstein, Lucia A. Hindorff, David J. Hunter, Mark I. McCarthy, Erin M. Ramos, Lon R. Cardon, Aravinda Chakravarti, Judy H. Cho, Alan E. Guttmacher, Augustine Kong, Leonid Kruglyak, Elaine Mardis, Charles N. Rotimi, Montgomery Slatkin, David Valle, Alice S. Whittemore, Michael Boehnke, Andrew G. Clark, Evan E. Eichler, Greg Gibson, Jonathan L. Haines, Trudy F. Mackay, Steven A. McCarroll, and Peter M. Visscher. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, October 2009.

[39] Mark I. McCarthy, Goncalo R. Abecasis, Lon R. Cardon, David B. Goldstein, Julian Little, John P. A. Ioannidis, and Joel N. Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nat Rev Genet*, 9(5):356–369, May 2008.

[40] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[41] J.J. Michaelson, R. Alberts, K. Schughart, and A. Beyer. Data-driven assessment of eqtl mapping methods. *BMC genomics*, 11(1):502, 2010.

[42] Stephan Morgenthaler and William G Thilly. A strategy to discover genes that carry multi-allelic or mono-allelic risk for common diseases: a cohort allelic sums test (cast). *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 615(1):28–56, 2007.

[43] Alison A Motsinger-Reif, David M Reif, Theresa J Fanelli, and Marylyn D Ritchie. A comparison of analytical methods for genetic association studies. *Genetic epidemiology*, 32(8):767–778, 2008.

[44] Richard Mott and Jonathan Flint. Simultaneous detection and fine mapping of quantitative trait loci in mice using heterogeneous stocks. *Genetics*, 160(4):1609–1618, 2002.

[45] Ulrike Ober, Julien F Ayroles, Eric A Stone, Stephen Richards, Dianhui Zhu, Richard A Gibbs, Christian Stricker, Daniel Gianola, Martin Schlather, Trudy FC Mackay, et al. Using whole-genome sequence data to predict quantitative trait phenotypes in drosophila melanogaster. *PLoS genetics*, 8(5):e1002685, 2012.

[46] Karim Oualkacha, Zari Dastani, Rui Li, Pablo E Cingolani, Timothy D Spector, Christopher J Hammond, J Brent Richards, Antonio Ciampi, and Celia MT Greenwood. Adjusted sequence kernel association test for rare variants controlling for cryptic and family relatedness. *Genetic epidemiology*, 37(4):366–376, 2013.

[47] Nadia Payet and Sinisa Todorovic. (rf)ˆ2 – random forest random field. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1885–1893. Curran Associates, Inc., 2010.

[48] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[49] Paola Picotti, Mathieu Clément-Ziza, Henry Lam, David S Campbell, Alexander Schmidt, Eric W Deutsch, Hannes Röst, Zhi Sun, Oliver Rinner, Lukas Reiter, et al. A complete mass-spectrometric map of the yeast proteome applied to quantitative trait analysis. *Nature*, 494:266–270, 2013.

[50] Barbara Rakitsch, Christoph Lippert, Oliver Stegle, and Karsten Borgwardt. A lasso multi-marker mixed model for association mapping with population structure correction. *Bioinformatics*, 29(2):206–214, 2013.

[51] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, December 2006.

[52] G. K. Robinson. That blup is a good thing: The estimation of random effects. *Statistical Science*, 6(1):pp. 15–32, 1991.

[53] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[54] Nicholas J Schork, Sarah S Murray, Kelly A Frazer, and Eric J Topol. Common vs. rare allele hypotheses for complex diseases. *Current*

opinion in genetics & development, 19(3):212–219, 2009.

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

[56] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, January 2013.

[57] LeahC. Solberg, William Valdar, Dominique Gauguier, Graciela Nunez, Amy Taylor, Stephanie Burnett, Carmen Arboledas-Hita, Polinka Hernandez-Pliego, Stuart Davidson, Peter Burns, Shoumo Bhattacharya, Tertius Hough, Douglas Higgs, Paul Klenerman, WilliamO. Cookson, Youming Zhang, RobertM. Deacon, J.NicholasP. Rawlins, Richard Mott, and Jonathan Flint. A protocol for high-throughput phenotyping, suitable for quantitative trait analysis in mice. *Mammalian Genome*, 17(2):129–146, 2006.

[58] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(307):1471–2105, 2008.

[59] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(25):1471–2105, 2007.

[60] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[61] Andreĭ Nikolaevich Tikhonov. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer, 1995.

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

[63] Michael C Wu, Seunggeun Lee, Tianxi Cai, Yun Li, Michael Boehnke, and Xihong Lin. Rare-variant association testing for sequencing data with the sequence kernel association test. *The American Journal of Human Genetics*, 89(1):82–93, 2011.

[64] Futao Zhang, Eric Boerwinkle, and Momiao Xiong. Epistasis anal-

ysis for quantitative traits by functional regression model. *Genome research*, 24:989–998, 2014.

[65] Z. Zhang, E. Ersoz, C.Q. Lai, R.J. Todhunter, H.K. Tiwari, M.A. Gore, P.J. Bradbury, J. Yu, D.K. Arnett, J.M. Ordovas, et al. Mixed linear model approach adapted for genome-wide association studies. *Nature genetics*, 42(4):355–360, 2010.

[66] Xiang Zhou and Matthew Stephens. Genome-wide efficient mixed-model analysis for association studies. *Nature genetics*, 44(7):821–824, 2012.

[67] Or Zuk, Eliana Hechter, Shamil R Sunyaev, and Eric S Lander. The mystery of missing heritability: genetic interactions create phantom heritability. *Proceedings of the National Academy of Sciences*, 109(4):1193–1198, 2012.

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschlielich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Andreas Beyer betreut worden.

Köln, den 9. September 2015

(Johannes Stephan)