Applications and extensions of Random Forests in genetic and environmental studies

Dissertation

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

von

Jacob James Michaelson, MS
geboren am 22. Oktober 1980 in Bountiful, Utah, USA

Betreuer:	Dr. Andreas Beyer Technische Universität Dresden
Betreuender Hochschullehrer:	Prof. Dr. Michael Schroeder Technische Universität Dresden
Gutachter:	Prof. Dr. Joachim Selbig MPI für Molekulare Pflanzenphysiologie, Potsdam
Tag der Einreichung:	08. 10. 2010
Tag der Verteidigung:	20. 12. 2010

For Jasmine

She openeth her mouth with wisdom;
and in her tongue is the law of kindness.
— Proverbs 31:26

Acknowledgements

This dissertation represents not only the culmination of more than a tenth of my life, but also the collective efforts and sacrifices of many others on my behalf. To all of them I am grateful and indebted.

First I wish to thank my advisor, Andreas Beyer, for providing the resources and direction that made all of my work possible. His keen mind and intuition directed me away from pitfalls and toward success.

I thank my loving parents for their constant support and for teaching me to work and to be curious about the world. I thank my parents via matrimony for their unwavering support and encouragement, and for allowing me to temporarily take their daughter and grandson to the other side of the planet.

I would also like to thank my siblings and their spouses for their encouragement and generosity: Amy and Dave, Laura and John, Missy, John and Gina, Adam and Summer, Spencer and Jaidyn, and Ford.

My colleagues in Andreas' group played an indispensable role by giving suggestions and critiques that shaped my work: Anna, Angela, Sinan, and Salvatore in the Mediterranean Room, and especially my local colleagues in the Continental Room: Weronika, Marit, Michael, and Mathieu. I especially thank Marit for our extensive conversations about statistical issues and Mathieu for his valuable perspectives as a seasoned biologist. I would also like to thank Boris Vassilev, who is the Bulgarian voice in my head whenever I'm writing code.

Of course I cannot help but acknowledge our group's support staff that keeps our ship afloat from day to day. To Ralf for keeping my workstation and the servers humming along, and to Mandy for processing an insane amount of travel paperwork and preventing me from being deported...and for having the best laugh of anyone on the 2^{nd} floor.

Much of my work centers around Random Forests. I would like to give my deep thanks to Adele Cutler, the mother of Random Forests, who as my undergraduate mentor was the person that got me interested in statistics and computational biology in the first place.

As a computational biologist, success is impossible without good experimental collaborators. I am grateful to Saskia, Franzi, Stefan, Irina, and Martin at the UFZ in Leipzig, Dani and Kristin at the EAWAG in Zürich, Rudi and Klaus at the HZI in Braunschweig, and Rupert and Gerd at the CRTD here in Dresden, all of whom worked tirelessly to produce top-quality data.

I thank the good people of Germany, especially the Saxons, for allowing me the opportunity to study and live in this beautiful and culturally rich country. Germany will always be a part of our family's identity.

Finally, I thank my family. My sweet Jasmine has sacrificed nearness to her family and has put her career on hold to make this experience possible. She is my life. She keeps me clean and fed and upbeat. I thank my little Jethro for sending me off with a hug and kiss every morning and welcoming me back home at the end of every day.

Publications

The research done during my dissertation led to the following publications and presentations:

Publications

1. **Michaelson, J. J.** and Beyer, A. Transcriptional regulatory contexts and epistasis among schizophrenia risk genes. (in preparation)

2. **Michaelson, J. J.**, Trump, S., Rudzok, S., Gräbsch, C., Madureira, D., Dautel, F., Schirmer, K., von Bergen, M., Lehmann, I., and Beyer, A. Transcriptional signatures of regulatory and toxic responses to chemical exposure. (submitted)

3. Loguercio, S., Overall, R., **Michaelson, J.J.**, Wiltshire, T., Pletcher, M.T., Miller, B.H., Walker, J., Kempermann, G., Su. A., and Beyer, A. Integrative analysis of low- and high-resolution eQTL. *PLoS ONE* 2010. 5(11): e13920.

4. Dautel, F., Kalkhof, S., Trump, S., **Michaelson, J.J.**, Beyer, A., Lehmann, I., and von Bergen, M. DIGE-based protein expression analysis of BaP-exposed hepatoma cells reveals a complex stress response at toxic and subacute concentrations. *J. Proteome Res.* 2010.

5. **Michaelson, J.J.**, Alberts, R., Schughart, K., and Beyer, A. Data-driven assessment of eQTL mapping methods. *BMC Genomics* 2010. 11:502.

6. **Michaelson, J.J.**, Loguercio, S. and Beyer, A. Detection and interpretation of expression quantitative trait loci (eQTL). *Methods* 2009. 48, 265-276.

Presentations

1. **Michaelson, J.J.** and Beyer, A. Transcriptional regulation in schizophrenia. Systems Biology: Networks 2010, Hinxton, UK.

2. **Michaelson, J.J.** and Beyer, A. Molecular mechanisms in schizophrenia uncovered with systems genetics. Systems Biology of Human Disease 2010, Boston, USA.

3. **Michaelson, J.J.** and Beyer, A. Identifying genetic interactions involved in adult neurogenesis. CRTD Bioinformatics Symposium 2009, Dresden, Germany.

4. **Michaelson, J. J.**, Trump, S., Madureira, D., Dautel, F., von Bergen, M., Schirmer, K., Lehmann, I., and Beyer, A. The *Ahr* transcriptional cascade. Helmholtz Alliance on Systems Biology – Status Meeting 2009, Heidelberg, Germany.

5. **Michaelson, J.J.**, Ackermann, M., and Beyer, A. Uncovering interactions with Random Forests. useR! 2009, Rennes, France.

6. **Michaelson, J.J.** and Beyer, A. Exploring the regulatory architecture of neurotransmitter receptors with Random Forests. INCF 2009, Pilsen, Czech Republic.

7. **Michaelson, J.J.**, Alberts, R., Schughart, K., and Beyer, A. Exploring the genetics of gene expression with Random Forests. ISMB 2009, Stockholm, Sweden.

8. **Michaelson, J.J.** and Beyer, A. Random Forests for eQTL analysis: a performance comparison. useR! 2008, Dortmund, Germany.

Summary

Transcriptional regulation refers to the molecular systems that control the concentration of mRNA species within the cell. Variation in these controlling systems is not only responsible for many diseases, but also contributes to the vast phenotypic diversity in the biological world. There are powerful experimental approaches to probe these regulatory systems, and the focus of my doctoral research has been to develop and apply effective computational methods that exploit these rich data sets more completely. First, I present a method for mapping genetic regulators of gene expression (expression quantitative trait loci, or eQTL) using Random Forests. This approach allows for flexible modeling and feature selection, and results in eQTL that are more biologically supportable than those mapped with competing methods. Next, I present a method that finds interactions between genes that in turn regulate the expression of other genes. This is accomplished by finding recurring decision motifs in the forest structure that represent dependencies between genetic loci. Third, I present a method to use distributional differences in eQTL data to establish the regulatory roles of genes relative to other disease-associated genes. Using this method, we found that genes that are master regulators of other disease genes are more likely to be consistently associated with the disease in genetic association studies. Finally, I present a novel application of Random Forests to determine the mode of regulation of toxin-perturbed genes, using time-resolved gene expression. The results demonstrate a novel approach to supervised weighted clustering of gene expression data.

Contents

Chapter 1

Introduction

1.1 Motivation

Traits that impact our everyday lives – like sickness, obesity, mental illness, or addiction – are an outward manifestation of the hidden systems of interactions taking place within and between our personal biology and the environment we live in. The more we understand the hidden systems that underlie these traits, the better equipped we are to design interventions to treat them.

One such hidden system that has an appreciable impact on physical traits like disease is transcriptional regulation. Transcriptional regulation refers to the molecular systems in the cell that dictate which pieces of information (in the form of mRNA) are emitted from the nucleus of a cell, the timing of their emission, and the intensity of the emission (concentration of mRNA). In short, transcriptional regulation determines the usable information content of the cell. A cell's information content determines whether it is capable of performing its assigned task. If information content is disregulated and the cell cannot perform its designated task, disease often ensues.

Because of its far-reaching role in connecting molecular biology to macroscopic biology, transcriptional regulation has been a booming field over the past decade. There are many tools to study it, but perhaps none more ubiquitous than the DNA microarray. With microarrays, one can get a snapshot of the expression of all genes in a sample of cells. Microarrays have been used to look at gene expression in experimental studies (e.g. how does compound XYZ change gene expression?) and observational studies (e.g. variation of gene expression across a population).

One application of microarrays is of particular importance to my work: eQTL studies. eQTL (expression quantitative trait locus) studies are usually performed in genetically well-defined populations of model organisms, where snapshots of gene expression are taken in a specific tissue (brain, for example), across the genetically distinct individuals of the population. Since all the animals were raised in the same conditions, whatever variation in gene expression is observed must be due to the genetic differences between the individuals. From a computational perspective, the task is to model mRNA expression (the response vector) in terms of the genetic (DNA) variation (the matrix of predictors). The genetic loci that contribute to the model of mRNA expression are called eQTL, and they represent the genomic location of probable regulators of the expressed gene in question. Consequently, the latent information in eQTL studies is very valuable, but requires the appropriate computational methodology to be fully exploited.

In my research, I have developed and applied methods to uncover transcriptional regulatory relationships in genetic and environmental studies. Here I have divided my work into four parts, the first three

dealing with methods and applications for eQTL studies, and the last dealing with prediction of the mode of transcriptional induction in an environment-centric study.

1.1.1 Description of Random Forests

Much of the work in this dissertation applies and extends Random Forests (RF) (Breiman, 2001), a machine learning method that has seen a wide variety of applications in the decade since its introduction. To provide context for my doctoral work, an overview of Random Forests is presented here.

Random Forests are ensembles, or collections, of decision trees. The individual trees are either regression or classification trees, depending on whether the modeled response is continuous or categorical, respectively. The decision trees within the forest differ from one another in two important ways. First, each tree is fit to a different bootstrap sample of the original data. Because of the bootstrap sampling (i.e. sampling with replacement), some observations will be left out of the sample, while others are replicated more than once in the sample. The left out samples (called out-of-bag data) form an implicit test set and are used to calculate an unbiased estimate of the classification or regression error of the forest. Second, each split in each tree is selected not as the overall best split, but rather the best of a randomly selected subset of predictor variables. Because of these two elements of stochasticity – bootstrap sampling and selection of the optimal split from a random subset of variables – each tree represents a slightly different solution to the same problem.

The combined prediction of all trees in the forest is a more reliable prediction than any of the individual trees, since it accounts for (or rather utilizes) variations in the sampling distribution of the data. In classification forests, a new observation is run down all the trees in the forest, and each tree casts a vote for the class it predicts for the new observation. The forest prediction for the new observation is the class with the majority of votes. For regression forests, the prediction for a new observation is the average of the individual tree predictions.

Random Forests have several practical advantages over other competing machine learning methods. It is generally difficult to overfit with RF, it makes no distributional assumptions about the data, it can handle correlations among the predictor variables, and can handle mixtures of continuous and categorical predictor variables. In addition to its utility in making predictions, it provides measures of variable importance for the predictors, and provides a proximity measure that indicates the (weighted) similarity of the observations. Both of these features of RF have led to applications beyond mere prediction, and they are described in the following sections.

Variable importance

In the context of RF, variable importance refers to any of three measures of a predictor's contribution to the success of the forest. Perhaps the most widely used is the permutation importance, which assesses the impact of randomly permuting the values of the variable in question. This importance measure is the average degradation of the forest's predictive accuracy upon permutation of the variable. Thus, a positive value represents an important variable, whereas a negative value or value close to zero indicates an unimportant variable. The second variable importance measure indicates the average increase in node "purity" as the result of a split on a variable. This is the reduction of residual sum of squares (RSS) in the regression context, and the Gini index in the classification context. Finally, the number of times a variable is used to split in the forest can be used as an indicator of variable importance. This is called selection

frequency. Clearly, this measure is not as sophisticated as the others, but may still be useful in some situations.

These importance measures are often used when the goal of analysis is not prediction, but rather to determine which features (e.g. genes) are relevant to the process being modeled. This is particularly useful when there are many irrelevant predictor variables, which is often the case in high-throughput data.

Proximity

The RF proximity is a measure of how frequently two observations end up in the same terminal node, throughout the decision trees of the forest. This leads to an $N \times N$ matrix, whose values are between 0 and 1 and represent the frequency of co-occurrence in terminal nodes of the i^{th} and j^{th} observations (the diagonal of the matrix is 1). A high proximity value indicates that observations frequently take the same decision path to their ultimate classifications. This measure provides implicit weighting of features, since only features with discriminatory power are used repeatedly in the RF. Since the proximity matrix is square and contains values that indicate the similarity of two subjects, it can be transformed to a distance matrix and be used in a variety of clustering algorithms. This can be especially useful since RF can be used to integrate information from a mixture of continuous and categorical data, which cannot be done with traditional distance measures such as the Euclidean and Manhattan distances. In addition, since RF can be run in either a supervised or unsupervised mode, the resulting proximity measure can be either a supervised or unsupervised measure.

Applications

Because of its flexibility and many features, RF has been widely applied since its introduction, especially in the biological sciences. It has been used in association studies and GWAS (Lunetta et al., 2004; Motsinger-Reif et al., 2008; Kim et al., 2009b; Goldstein et al., 2010), in QTL studies (Bureau et al., 2003; Lee et al., 2008), and for finding gene-gene interactions (Bureau et al., 2005; Wang et al., 2010c). It has also been used in pathway analysis (Pang et al., 2006; Pang and Zhao, 2008; Chang et al., 2008; Xiao and Segal, 2009; Pang et al., 2010) and modeling medical diagnostic data (Han, 2006). The proximity measure has also been used to cluster tumor samples in an unsupervised setting (Shi and Horvath, 2006) and to reconstruct regulatory networks in yeast (Xiao and Segal, 2009) in a supervised setting.

1.2 Definition of open problems

1.2.1 Open problem 1: Mapping expression quantitative trait loci (eQTL)

Context

Since the dawn of modern genetics, geneticists have striven to link observable traits to causal genes, in the hopes of learning enough to enhance desirable traits and prevent or treat detrimental ones. Some traits are categorical, for example, the presence or absence of some developmental disorder. Other traits are quantitative: height, blood pressure, cholesterol levels, etc. In the case of quantitative traits, geneticists have used a strategy called quantitative trait locus (QTL) mapping to expose the underlying genetic architecture that controls the trait. At its simplest, this works by breeding and thoroughly characterizing a small population of animals, then looking for correlations between variation in the trait of interest and allelic

variation across the same population at genotyped loci. Genetic loci that correlate strongly with a trait are thought to be the causal source of the trait, since from the trait's perspective the genome is "read-only"; in other words, the trait does not cause the genetic variation, but the genetic variation can influence the trait, thus establishing a causal relationship.

In the last decade, molecular biology underwent a rebirth due to the introduction of high throughput molecular technologies such as the DNA microarray. These technologies gave researchers access to thousands of new molecular traits, including gene (mRNA) expression measurements for all genes in an organism's genome. Traditional genetics approaches, such as QTL mapping, were then applied to these new genome-wide expression traits, giving rise to the expression quantitative trait locus, or eQTL. eQTL are genetic loci that influence gene expression. In this sense, mapping eQTL for all genes probed on a microarray can give a global view of transcriptional regulation. Increased understanding of transcriptional regulation will lead to a better understanding of the molecular mechanisms that underlie organismal traits of interest.

In order to correctly identify the loci (and subsequently the genes) that contribute to variation in gene expression, an appropriate statistical method must be used. Traditional QTL mapping methods are in effect univariate tests that test the trait-locus association one-by-one, ignoring all other loci in the genome. Such an approach will fail to capture the combinatorial effects of multiple loci responsible for complex traits such as gene expression. Methods should be used that can account for multiple additive and conditional effects. These methods should be thoroughly tested to demonstrate their effectiveness not only on simulated data, but on real data as well. Such proven methods will lead to improved accuracy of the conclusions reached from the mapped eQTL data.

Open problem

How can the performance of eQTL mapping methods be tested using measured data? Can modifications of these methods produce improved results?

1.2.2 Open problem 2: Finding epistasis in systems genetics data

Context

Epistasis is a genetic-phenotypic phenomenon where a gene's contribution to a trait does not occur in isolation; rather, it is dependent on the genetic background of the organism (Carlborg and Haley, 2004). This is in contrast to the prevailing public perception of genes conferring traits unilaterally, e.g. media reports of the "cheating gene", the "fat gene", or the "gay gene". Epistasis is usually interpreted as an interaction between genes, meaning that the effect of one gene can be changed depending on the state of another gene. It is a relationship that suggests a more intimate molecular connection between genes than additivity (independent contributions to a trait) does. Indeed, researchers have used it as a tool to reconstruct molecular pathways in model organisms (Phillips, 2008; Tong et al., 2004; Schuldiner et al., 2005; Hannum et al., 2009; Costanzo et al., 2010). In addition, epistasis has received increased attention with the advent of high-throughput human genetics, and offers a framework for interpreting immense variation, such as with personal genetics (Moore and Williams, 2009), as well as for understanding complex diseases (Shao et al., 2008; Carlborg and Haley, 2004). As we come to better understand epistasis and its meaning, we will be better able to diagnose and treat disease on a personal level.

Open problem

How can epistasis be efficiently discovered among millions of locus pairs and tens of thousands of traits? How can competing methods be benchmarked using real data?

1.2.3 Open problem 3: Finding transcriptional regulatory contexts for phenotype-linked genes

Context

Because of their multigenic nature, complex diseases frequently have hundreds or even thousands of genes associated with them in the literature. Often a precise molecular etiology of these complex diseases is lost in the long list of risk genes. Further, directly testing hypothetical disease pathways is complicated by the infeasibility of the requisite experiments in humans. Because of this, eQTL studies in mice and rats have become an attractive means for investigating transcriptional regulation in tissues and conditions that are difficult to acquire in humans. The regulatory programs found in these model organisms can shed light on the potential roles of their orthologs in human disease (Chen et al., 2008; La Merrill et al., 2010).

Open problem

Which disease-associated genes are most likely to be causal, and which are likely to be symptomatic? How can systems genetics data be used to give clues about the etiology of a disease or the drivers of a phenotype?

1.2.4 Open problem 4: Classifying direct and indirect transcriptional targets using time-resolved gene expression data

Context

The aryl hydrocarbon receptor (*Ahr*) is a ligand-activated transcription factor that has generated interest because of its role in mediating the cellular response to toxins in the environment. Exposure of mammalian cells to the environmental contaminant B[a]P (benzo-[a]-pyrene) initiates a complex transcriptional response via *Ahr*. Part of this response is not due to direct regulation by *Ahr*, but rather by the cellular stress induced by the conversion of B[a]P to more toxic metabolites, such as anti-benzo(a)pyrene-trans-7, 8-dihydroxy-9,10-epoxid (BPDE). Distinguishing the response that is directly mediated by *Ahr* from the secondary response is a fundamental step in reconstructing the *Ahr* pathway.

Open problem

Given an extensive transcriptional response upon induction of a transcription factor, how can direct targets be distinguished from indirect effects? How can the responding genes be clustered in functional groups in a way that accounts for individual transcript differences in synthesis and degradation?

1.3 Thesis outline

Each of the open problems described here will be presented in an individual chapter. Chapter 2 deals with open problem 1 and explores both legacy and modern methods for mapping expression quantitative trait loci (eQTL). Chapter 3 deals with open problem 2 and builds on the findings in chapter 2 as it presents a new and effective method for detecting epistasis (genetic interactions). Chapter 4 addresses open problem 3 and can be seen as the practical culmination of the work in the previous two chapters, specifically by presenting a novel method for using eQTL data to infer the regulatory context (e.g. transcriptional regulator or target) of disease risk genes. Finally, chapter 5 deals with open problem 4 and presents an application of Random Forests for predicting direct vs. indirect targets of an inducible transcription factor. It also demonstrates how the Random Forests proximity measure can be used as an effective weighted distance measure for time course expression data.

The appendix complements the theory and applications presented in the body of the dissertation by including tutorials designed to familiarize readers with the use of the methods and approaches presented here. Readers are then equipped to be able to apply and build on the methods for their own purposes.

Chapter 2

Mapping expression quantitative trait loci (eQTL)

The following publications and presentations relate to the work presented in this chapter:

1. Loguercio, S., Overall, R., **Michaelson, J.J.**, Wiltshire, T., Pletcher, M.T., Miller, B.H., Walker, J., Kempermann, G., Su. A., and Beyer, A. Integrative analysis of low- and high-resolution eQTL. *PLoS ONE* 2010. 5(11): e13920.

2. **Michaelson, J.J.**, Alberts, R., Schughart, K., and Beyer, A. Data-driven assessment of eQTL mapping methods. *BMC Genomics* 2010. 11:502.

3. **Michaelson, J.J.**, Loguercio, S. and Beyer, A. Detection and interpretation of expression quantitative trait loci (eQTL). *Methods* 2009. 48, 265-276.

4. **Michaelson, J.J.**, Alberts, R., Schughart, K., and Beyer, A. Exploring the genetics of gene expression with Random Forests. ISMB 2009, Stockholm, Sweden.

5. **Michaelson, J.J.** and Beyer, A. Random Forests for eQTL analysis: a performance comparison. useR! 2008, Dortmund, Germany.

2.1 Introduction

For decades scientists have used a variety of analytical techniques to relate allelic inheritance patterns in the genome to variation in continuous physical traits of interest. The goal of such analyses is often to locate quantitative trait loci (QTL), or genomic locations that exert an influence on the manifested trait. Understanding the genomic location of these genetic control points may provide insight into the genetic and molecular framework responsible for enabling the trait.

In the past decade, the advent of the DNA microarray and other high-throughput molecular technologies has updated the paradigm of the QTL. A QTL where mRNA expression is the complex trait of interest is generally referred to as an expression QTL or eQTL (Rockman and Kruglyak, 2006). By using DNA microarrays eQTL can be measured for basically all genes in the genome, rendering eQTL data information rich and potentially very powerful. eQTL have been studied in yeast, mouse, rat, human, and plants (Brem et al., 2005; Kempermann et al., 2006; Petretto et al., 2006; Veyrieras et al., 2008; Druka et al., 2008) and

eQTL have proven to be useful for elucidating the molecular mechanisms of human diseases (Sieberts and Schadt, 2007; Chen et al., 2008; Schadt and Lum, 2006; Michaelson et al., 2009).

Although complex traits are by definition controlled by the coordination of multiple genes, the prevailing techniques for mapping them have been deeply rooted in univariate thinking – testing for genetic association to a trait one locus at a time, ignoring combinatorial effects and interactions. In contrast, Broman and Speed (Broman and Speed, 2002) defined the QTL problem as one of multivariate variable selection, where ideally all loci and their combinations are allowed to enter and exit the model as the data dictate. Viewing eQTL mapping as a variable selection problem opens the door to using a host of machine learning algorithms which have rarely, if at all, been applied to QTL and eQTL studies (Chun and Keles, 2009; Huang et al., 2009; Lee et al., 2008; Bureau et al., 2005). Such a fresh look at the QTL problem may help to uncover latent and meaningful information in otherwise underexploited data.

A systematic comparison of eQTL mapping approaches is necessary to inform the research community which methods work best and in which contexts. Toward that goal, the purpose of this work is twofold. First, we establish a framework for comparing available eQTL mapping methods based on the tendency of each method to map eQTL that are systematically supported by external biological data. This is important because methods papers proposing new (e)QTL mapping techniques often draw their conclusions either solely or largely on the basis of simulated data (Broman and Speed, 2002; Chun and Keles, 2009; Lee et al., 2008; Benjamini and Yekutieli, 2005; Bureau et al., 2003; Jiang and Zeng, 1995; Zeng, 1994; Haley and Knott, 1992; Lander and Botstein, 1989). This is perhaps understandable in the case of earlier work with QTL, where only a limited number of phenotypes were available and external knowledge about their context and probable genetic regulators was not available in a systematic form, making biology-based benchmarking difficult. However, this is not the case in the era of eQTL. Although some genes remain uncharacterized, there are rich sources of data for many genes that give insight about their role and context within the cell. Such knowledge is often contained in databases like the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000), which makes using it as a basis for a benchmark easier. Our battery of knowledge-driven benchmarks consists of 1) assessing the proportion of *cis*-eQTL recovered by each method, 2) testing each method's high-scoring eQTL for enrichment of loci related to the target by KEGG pathway information, and 3) agreement of each method's high-scoring eQTL with systematic loss-of-function studies. In this framework we tested three variable importance measures from Random Forests (RF) (Breiman, 2001) as well as sparse partial least squares (SPLS) (Chun and Keles, 2009), the lasso (Tibshirani, 1996), the elastic net (Zou and Hastie, 2005), Haley-Knott regression (HK) (Haley and Knott, 1992), and composite interval mapping (CIM) (Zeng, 1994). We also performed simulations to complement the findings of the knowledge-driven benchmarking framework. We show that multi-locus methods in general (Random Forests, SPLS, lasso, elastic net) are better at recovering biologically meaningful loci than traditional QTL mapping methods such as HK and CIM.

Second, we demonstrate that based on both simulations and the knowledge-driven benchmarks, RF shows superior performance as an eQTL mapping method. RF has previously been applied to genome-wide association studies (GWAS) and QTL studies (Bureau et al., 2003; Lunetta et al., 2004; Bureau et al., 2005; Motsinger-Reif et al., 2008; Lee et al., 2008). The contribution of our work, however, lies in the discovery that the most naive measure of variable importance in RF, the variable selection frequency (RFSF), actually performs much better than the more popular permutation importance (RFPI) in this context. Since RFSF has been ignored in all previous works using RF in the QTL or GWAS context, its use here represents a novel eQTL mapping method with demonstrated superior performance.

Definition of open problem

How can the performance of eQTL mapping methods be tested using measured data? Can modifications of these methods produce improved results?

2.2 Materials and Methods

2.2.1 eQTL mapping

We used expression data from four eQTL studies in four different tissues in recombinant inbred BXD mouse strains: regulatory T-cell, lung, hematopoietic stem cells (Bystrykh et al., 2005), and hippocampus (Overall et al., 2009). We used only probe sets that mapped unambiguously to Ensembl gene IDs with KEGG annotations (Kanehisa and Goto, 2000). This resulted in a set of 6,121 probe sets for studies using the Affymetrix Mouse 430 2.0 array (lung, regulatory T-cell, and hippocampus) and 3,051 probe sets for the hematopoietic stem cell study, which used the Affymetrix U74Av2 array. Genotype data for the BXD recombinant inbred strains of mice used in these studies consisted of 3,794 markers and was downloaded from the GeneNetwork database (Wang et al., 2003). In addition to the mouse data, we used the yeast eQTL study previously published in (Brem and Kruglyak, 2005). After filtering out probes with missing or otherwise ambiguous data, we were left with 4,501 gene expression measurements and 2,914 markers.

Random Forests

We used the reference implementation of Random Forests (Liaw and Wiener, 2002) in R for all mapping discussed in this work. We grew forests with 5,000 trees, the `mtry` parameter was set to the default (one third of the total number of markers) and the node size was also the default of 5, unless otherwise noted. We then extracted unscaled permutation importance measures (RFPI), residual sums of squares importance measures (RFRSS), and selection frequencies (RFSF) from the forests for use as the scores for each marker.

We estimated and accounted for bias in RFSF as follows. Using the actual genotype data as predictors, we fit 500 10-tree forests to independent draws from Gaussian noise. This resulted in 5,000 trees, equal in size to the forests used in this work. We collected the selection frequencies for each marker and subtracted the mean selection frequency to yield a vector of correction factors — one value for each marker. Subtracting this vector of correction factors from the observed selection frequencies (from the observed data) gives bias-corrected selection frequencies (Fig. 2.8). In the context of results in this work, all references to RFSF imply the bias-corrected RFSF, as described here.

Sparse partial least squares

Chun and Keles (Chun and Keles, 2009) recently introduced a method of eQTL mapping using sparse partial least squares, which included an R package and a thorough tutorial available online. We used the `spls` R package to map eQTL, performing cross-validation on every target to determine the optimal parameters for each fit. `eta`, the thresholding value, was allowed to vary between 0.3 and 0.7, to prevent both overfitting and a model that was too sparse to score multiple loci. The number of hidden components was allowed to vary from 1 to 5. A final fit was performed with the optimal parameters, and the absolute value of the coefficients was used as the score for each marker.

The lasso

The lasso (Tibshirani, 1996), a regression shrinkage method, has previously been applied to QTL mapping (Foster, 2007), but to our knowledge has never been tested against competing mapping methods in the context of an eQTL study. For this work, we used the lasso as implemented in the `elasticnet` package for R. The lasso is a special case of the elastic net with lambda equal to (or very near) 0. For each target gene examined, we took the absolute value of the lasso coefficients for a fit performed with the `s` parameter determined by 10-fold cross-validation, with an imposed minimum of 0.5. These coefficients were used as the score for each marker.

The elastic net

The use of the elastic net (Zou and Hastie, 2005) was the same as above for the lasso, except that lambda was set to 1. We found this value of lambda to be optimal after testing a sample of target genes over a range of lambda values (0.5,1,10,100).

Haley-Knott regression

We used the implementation of Haley-Knott regression (Haley and Knott, 1992) available in the `qtl` package for R. LOD scores were calculated at the marker locations.

Composite interval mapping

To perform composite interval mapping (Zeng, 1994) we used the implementation in the `qtl` package for R, with the `method` argument set to "EM", and all other arguments set to their default. LOD scores were calculated at the marker locations.

2.2.2 Simulations

To simulate eQTL with known underlying models, we used the full BXD genotype matrix, available from the GeneNetwork (Wang et al., 2003). This matrix consists of 89 strains and 3,794 markers. Using this genotype data, we randomly selected one, two, or three markers (depending on the model to be simulated), and then simulated a trait by using a linear combination of the markers directly, or of logical operations on the markers (in the case of epistasis). All traits started with a baseline value of 9, before adding in the genetic effects. Genetic effects were added as follows: in the single locus model, a single marker was selected at random, and its vector of genotypes (where 1=BB and 0=DD) was multiplied by a coefficient, in this case 1. For the two-locus epistatic model, two marker vectors were selected at random, with each being multiplied by 0.25 and then summed. The epistatic component was added by applying the AND logical operation to the genotype vectors (where a 1 is a TRUE and a 0 is a FALSE) and then multiplying the result by a coefficient, in this case 1, and then adding to the additive component. Three locus additive and epistatic traits were constructed in a similar fashion. Gaussian noise with mean 0 was then added to the traits, over 8 levels of increasing standard deviation, which corresponded to 2.5, 5, 7.5, 10, 12.5, 15, 17.5, and 20% of the trait mean. The resulting distributions are comparable to the distributions of expression values that are observed in real data.

Each model type (i.e. single locus, two locus epistatic, etc.) was simulated independently 50 times, and each mapping method was applied to the same data. For each simulation and for each mapping method, the maximum (i.e. worst) rank among the set of causal markers was recorded in each noise level. The median of these values (over the 50 simulations) was used to reflect the performance of a given mapping method over the increasing levels of noise. Lower values represent the ability of a method to assign high scores to *all* causal loci.

2.2.3 *cis*-eQTL counts

Performance based on the proportion of recovered *cis*-eQTL was assessed by counting the number of expression traits where a marker within 500 kb (for mouse) or 50 kb (for yeast) of the midpoint of the target gene's genomic location had a score in the 99^{th} percentile of the scores for the respective target gene. These cutoffs, though arbitrary, reflect the difference in complexity between the yeast and mouse genomes – the conclusions drawn from the benchmark are not heavily influenced by this choice. This number was then divided by the number of total expression traits examined for the respective data set.

2.2.4 KEGG enrichment

Each expression trait we tested mapped to at least one KEGG pathway, and each gene found in the KEGG pathway was mapped to the nearest marker. If no marker fell within 5 Mb of a gene, the gene was omitted. For each expression trait, the markers having scores in the 99^{th} percentile were selected for the enrichment test. The hypergeometric test was used to test this set for the enrichment of markers mapping to genes participating in the same KEGG pathway as the target gene. If multiple pathways existed for any expression trait, all were tested and the minimum P value was used as the representative P value.

In the case of the yeast eQTL data, we additionally assessed enrichment of pathways in which transcription factors binding to the target gene participate. As a basis for mapping transcription factors to their targets, we used (Beyer et al., 2006). We did not attempt this test with the mouse data because of the lack of dense and reliable TF-target data for mouse.

Since in this test even randomly selected markers yield P values that deviate somewhat from the uniform distribution, we calculated an empirical null distribution of P values. To construct this distribution, we assigned scores to the markers, drawn randomly from a Gaussian distribution with mean 0 and standard deviation of 1. We then took the markers in the 99^{th} percentile and performed the proposed enrichment test. This was performed for an equivalent number of expression traits contained in the actual data sets. The actual enrichment P values were corrected against this empirical null distribution of enrichment P values.

We plotted the empirical cumulative distribution function (ECDF) of the corrected enrichment P values for each method. As a summary measure for each method's deviation from the uniform distribution, we used the D-statistic as given by the Kolmogorov-Smirnov test. The test was one-sided with the alternative hypothesis that the observed cumulative distribution function accumulated faster than the reference (i.e. uniform) distribution.

2.2.5 Mutant expression change enrichment

Systematic loss of function data in yeast (Hughes et al., 2000; Mnaimneh et al., 2004) was used to assess which eQTL mapping method tended to agree most with the regulatory relationships suggested by experimentally deactivating upstream regulators. We mapped each repressed gene to its nearest marker. Then, for each expression trait from the yeast eQTL study, we looked at markers in the 99^{th} percentile of scores for that target. For markers mapping to experimentally repressed regulator genes, we collected the maximum absolute $\log_2$ expression ratio (repressed expression divided by wild-type expression) for the appropriate target gene, aggregating them over the whole set of mapped expression traits. We then compared the distribution of the selected maximum absolute $\log_2$ ratios generated by each eQTL mapping method by the Kolmogorov-Smirnov (KS) test, collecting the associated *P* value and D statistic. As a reference distribution in the KS test, a null distribution was constructed by a similar aggregation of maximum absolute fold changes, only with the association between scores and markers randomized for each target gene. The test was one-sided with the alternative hypothesis that the observed cumulative distribution function accumulated slower than the reference distribution. Distributions with a tendency toward higher scores and deviating significantly from the reference distribution suggest an agreement between the eQTL and loss-of-function studies.

2.2.6 Variation of tree depth

To assess the impact of tree depth on each RF importance measure, we used the yeast eQTL data and recomputed eQTL maps for all expression traits, varying the `nodesize` argument to 5, 15, 29, 57, and 114. The `nodesize` argument dictates whether or not a node may be split — if the number of observations in the node under consideration is greater than `nodesize`, the node may be split. Otherwise the node is not split and is marked as a terminal node. The default value of `nodesize` is 5 — this is the value used in the main body of the study. By selecting a `nodesize` of 114 (the number of samples in the yeast study), we ensure that splitting stops after the first split. The other values are intermediate steps, each about half the size of the last. We then assessed the improvement in the enrichment of KEGG pathway members and proportion of cis-eQTL identified when growing the trees deeper, using the forest with `nodesize` 114 as the baseline.

2.3 Results

In order to evaluate the performance of the eQTL mapping methods in a comprehensive way, we used both simulated data and a variety of published and previously unpublished experimental data from mouse and yeast. The mouse data sets include gene expression data from four tissues of recombinant inbred (RI) BXD mouse strains: regulatory T-cell (H. Chen, RA, and KS, unpublished data), lung (RA, L. Lu, R. Williams, and KS, unpublished data), hematopoietic stem cells (Bystrykh et al., 2005), and hippocampus (Overall et al., 2009). The yeast data were taken from (Brem and Kruglyak, 2005). Further details are available in the methods section.

We note here that one of the goals of this comparison is to determine how susceptible each method is to the effects of linkage disequilibrium. In light of this goal we used all genotype data as-is, without prefiltering or fusing markers, or assigning surrogate eQTL post-hoc. This enables a straightforward comparison across all mapping methods.

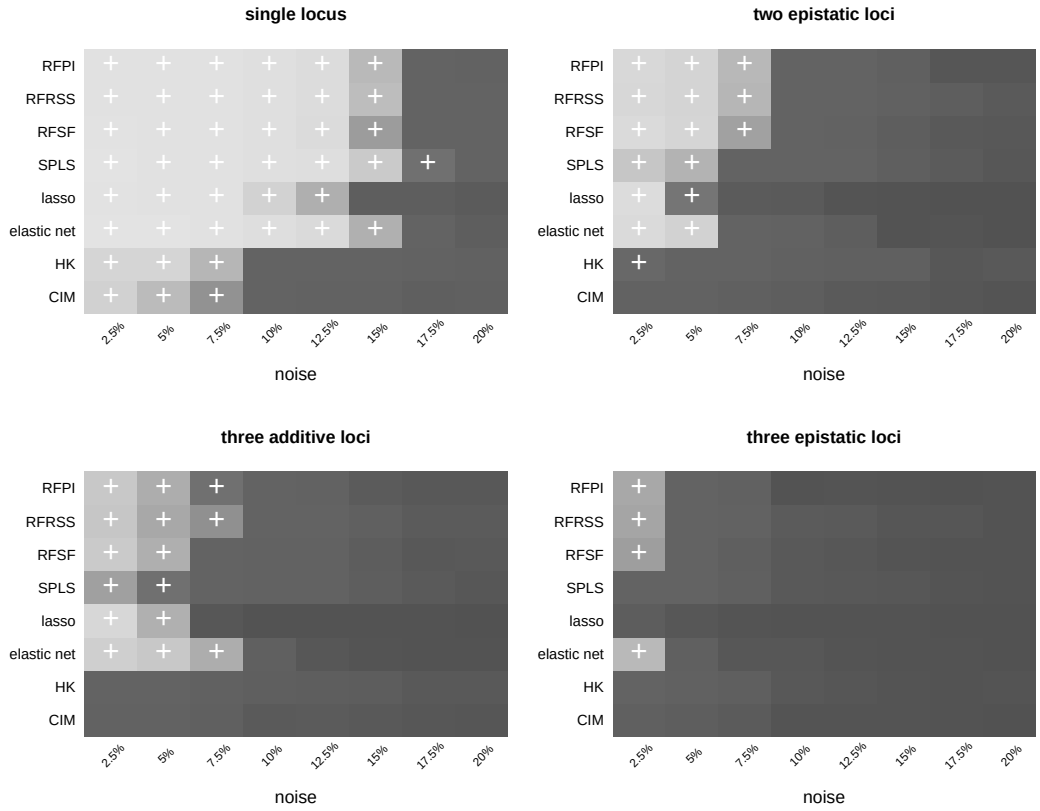


Figure 2.1 Results of the simulated eQTL models. Each method-noise level combination where all of the causal loci were contained in the 99^{th} percentile of scores is marked with a '+'. Ranking within the 99^{th} percentile (of the worst-ranking of the causal loci) is indicated by the shade of gray, with lighter shades indicating better ranking.

2.3.1 Simulations

We first set out to examine the performance of each method when the underlying model generating the data was known completely. We used the actual BXD genotypes and generated traits based on four models: single causal locus, two epistatic causal loci, three additive causal loci, and three epistatic loci. These configurations were sufficient to clearly distinguish the performance of the methods. Further details of the construction of the simulated data are given in the methods section. The goal of this investigation was to determine how well each method performed at placing *all* causal loci in the 99^{th} percentile of scores, over a range of increasing Gaussian noise in the trait. The results are given in Figure 2.1. In the single locus scenario, the performance gap between the newer multi-locus methods (RF, SPLS, the lasso, and the elastic net) and the legacy methods (HK and CIM) is quite apparent. In the single locus case, HK and CIM are unable to correctly identify causal loci in traits with more than 7.5% noise, and fail almost completely at pinpointing causal loci in the more complex two and three locus models. The elastic net and RF deliver comparable performance in the more complex models, with RF performing better in epistatic scenarios and the elastic net performing slightly better in the three-locus additive model. It should be noted that while SPLS, the lasso, and the elastic net do not explicitly search for interactions, they may still find loci participating in epistasis due to small but detectable marginal effects of the interaction.

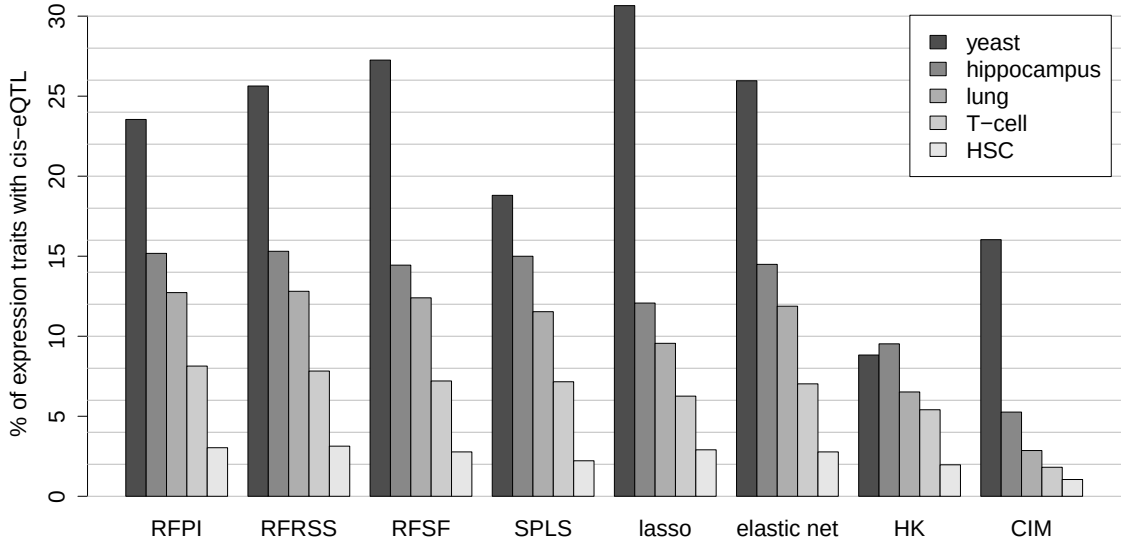


Figure 2.2 Percentage of expression traits with a recovered *cis*-eQTL. For each experimental data set, we calculated the percentage of transcripts which had a marker scoring in the 99^{th} percentile that co-localized with the genomic location of the target gene.

2.3.2 *cis*-eQTL counts

A "back of the envelope" approach for gauging the practical performance of a mapping method is the proportion of *cis*-eQTL found among all target transcripts in experimental data. Since promoter regions are often polymorphic, one would expect under optimal conditions to be able to recover an eQTL at the genomic location of many of the examined target transcripts. In this sense, the "external information" used in the benchmark is the knowledge of the genomic location of the gene — which, when compared to QTL in general, is information unique to eQTL. The results of this assessment are shown in Figure 2.2. Taken individually, no single method dominated the others. However, the legacy methods (HK and CIM) again showed poor performance when compared to their more modern counterparts. A relationship between study size and proportion of recovered *cis*-eQTL is also uncovered, with the larger studies (yeast, mouse hippocampus and mouse lung with 114, 67, and 44 observations, respectively) generally yielding higher proportions of *cis*-eQTL than smaller studies (mouse regulatory T-cell and mouse hematopoietic stem cell with 33 and 22 observations, respectively).

2.3.3 KEGG enrichment

We used the pathway information available in the KEGG database to establish relationships between target genes and potential regulators. KEGG was chosen because of its position as a standard in pathway information and because it is generally a better reflection of the molecular relationships between genes (compared to GO for instance). However, in principle other sources of pathway information could be used. One would not expect to recover an entire pathway in every eQTL map, but on a large scale there should be some overlap between the eQTL and the relationships contained in KEGG. We assert that methods that show higher agreement with the information in KEGG are more desirable for eQTL mapping. We

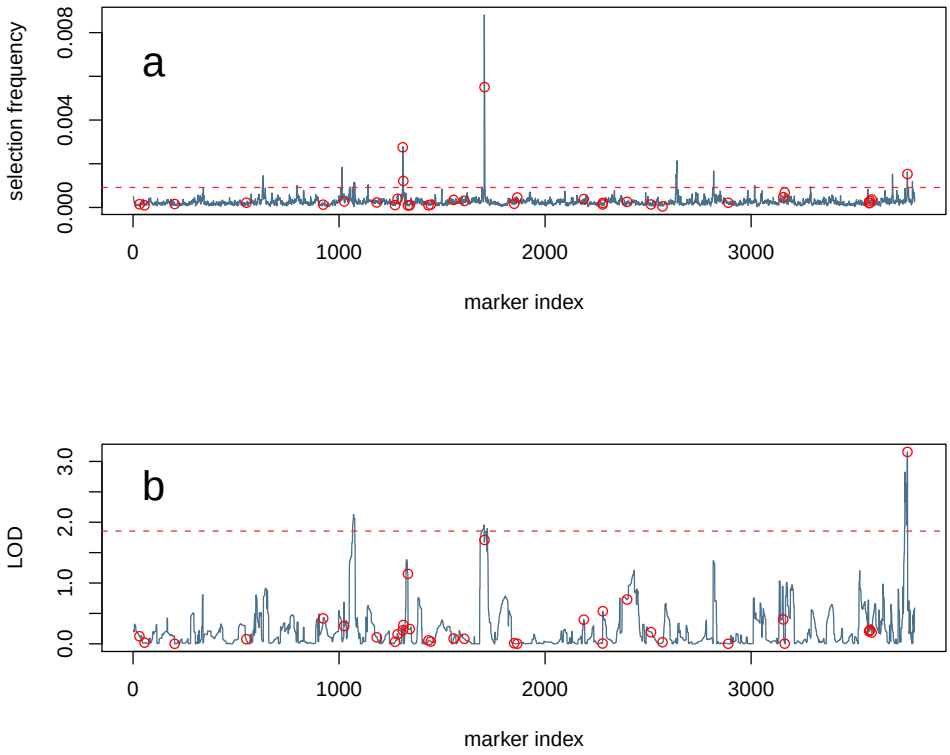


Figure 2.3 Comparison of eQTL profiles. An example eQTL profile for microarray probe set `1426838_at` (Pold3) from the hippocampus data set, using RFSF (A) as the importance measure. Loci near genes participating in the same pathway (DNA replication) as the target gene (Pold3 - a DNA polymerase) are marked with circles. The 99^{th} percentile of the values in this profile is marked with a dashed line. (B) The same target probe set, using HK as the eQTL mapping method. The traditional mapping methods based on the LOD score tend to have very broad, blunt peaks, sometimes spanning most of a chromosome. Random Forests, on the other hand, produces very sharp, narrow peaks.

formalize this by assessing the enrichment of high-scoring eQTL for loci near genes known to participate in the same pathways as the gene whose expression trait is being mapped. A graphical depiction of this idea is given in Figure 2.3 and further details on the enrichment test are given in the methods section.

We tested pathway enrichment in yeast and mouse eQTL separately. For yeast, we included an additional enrichment test, which connected target genes not to pathways in which they participate, but to pathways in which the target's known transcription factors participate. We used the distributional properties of the enrichment P values to compare the eQTL mapping methods, with results for the yeast data shown in Figure 2.4. It should be noted that HK did not deviate significantly from the uniform distribution in either the pathway member or the TF-centric enrichment tests (P = 0.72 and P = 0.07, respectively, by the Kolmogorov-Smirnov test). In contrast, RFSF showed superior performance on the yeast data ($P = 1.56 \times 10^{-133}$ and $P < 10^{-324}$ for the pathway member and TF-centric KEGG enrichment tests, respectively).

The mouse data showed more modest enrichment across all tissues and with all methods, suggesting perhaps that larger studies are needed to better recover the complex regulatory systems present in higher eukaryotes (Fig. 2.5). All methods yielded significant deviation from the uniform distribution in each tissue

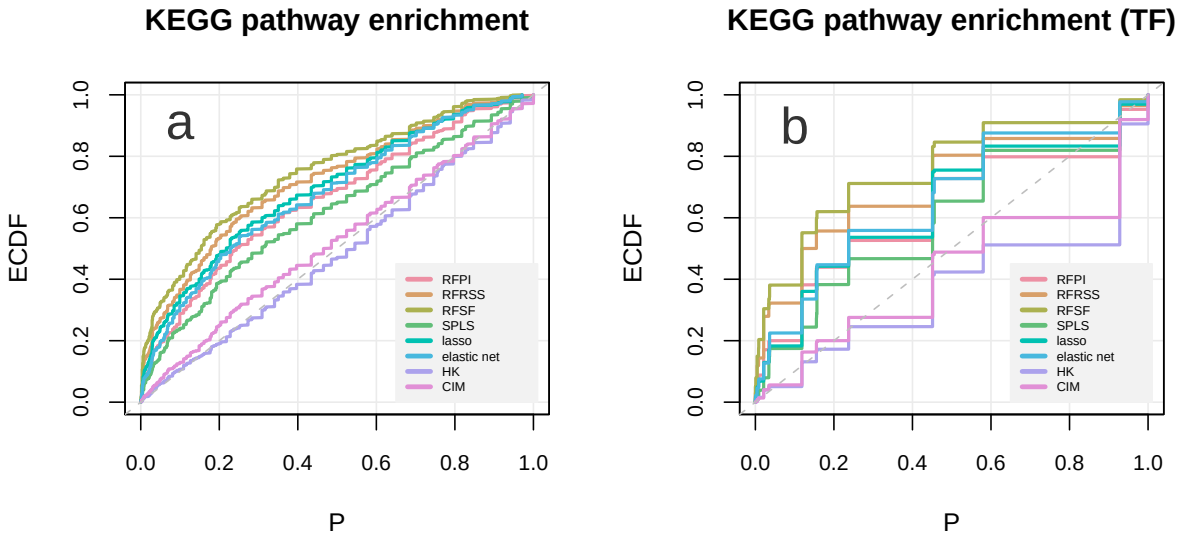


Figure 2.4 Empirical cumulative distribution functions (ECDF) of enrichment *P* values. The *P* values show the degree of enrichment among high-scoring yeast eQTL for genes that map to the same KEGG pathway as the target gene (A) and genes that map to the same pathway as the known transcription factors for the target gene (B). In both scenarios RFSF achieved the best performance in recovering loci enriched for pathway-related genes.

($P < 0.05$ by the KS test). Again, RFSF yielded the greatest degree of enrichment in all tissues.

SPLS, the lasso, and the elastic net produce sparse models, which means that not all loci are assigned a coefficient as a score. This had the effect that for a small minority of expression traits, the 99^{th} percentile of scores contained a small number loci with scores of 0. We examined whether this effect put these sparse methods at a disadvantage for the enrichment tests. We found no systematic relationship between enrichment *P* value and the number of 0 scores in the 99^{th} percentile.

2.3.4 Mutant expression change enrichment

Finally, we combined data from two systematic loss of function studies (Hughes et al., 2000; Mnaimneh et al., 2004) to see which method produced eQTL that agreed most with the mutant data.

In this test, we collected the maximum absolute expression change observed for each target gene when genes co-localized with eQTL in the 99^{th} percentile are mutated. These values were aggregated over all target genes, forming a distribution for each eQTL mapping method. We compared these distributions to a null distribution (see methods for details) via the Kolmogorov-Smirnov test. We assert that the method that yields eQTL that are enriched for large changes in expression in the mutant study is the most desirable method.

All methods produced score distributions that deviated significantly from the null distribution, suggesting that there is indeed consistency between the yeast eQTL data and independent mutant data. Although all methods showed significant deviation from the null, the magnitude of enrichment varied widely (Fig. 2.6). RFSF showed the most significant enrichment, followed closely by HK.

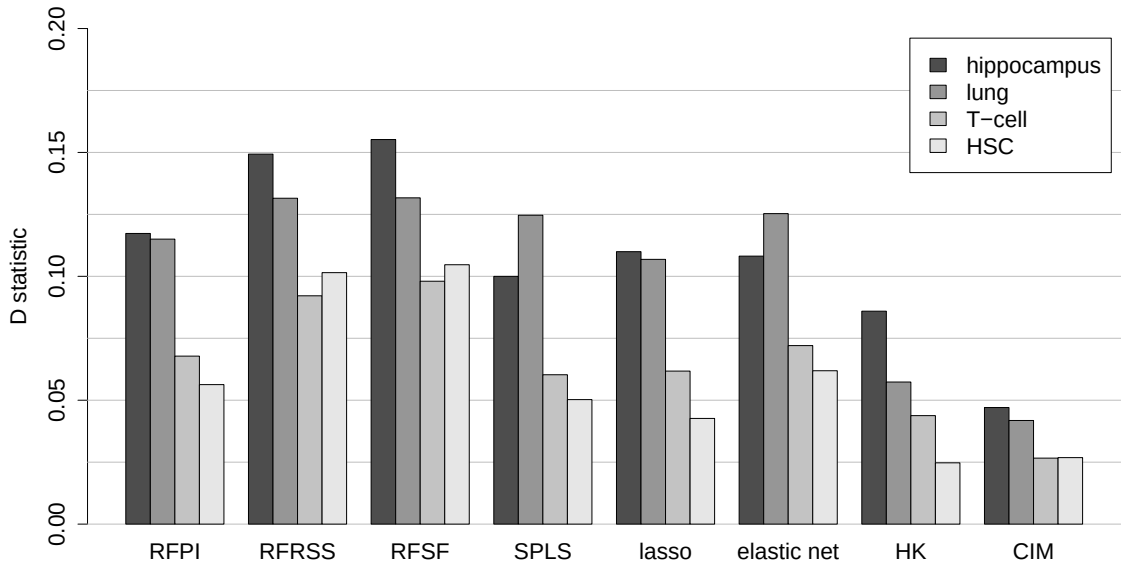


Figure 2.5 Enrichment of KEGG pathway members in top-scoring loci in mouse tissues hippocampus, lung, regulatory T-cell, and hematopoietic stem cell. The enrichment test procedure is the same as shown in Figure 2.4, but here the performance is summarized as the D statistic (maximum deviation from the uniform distribution) obtained from the Kolmogorov-Smirnov test.

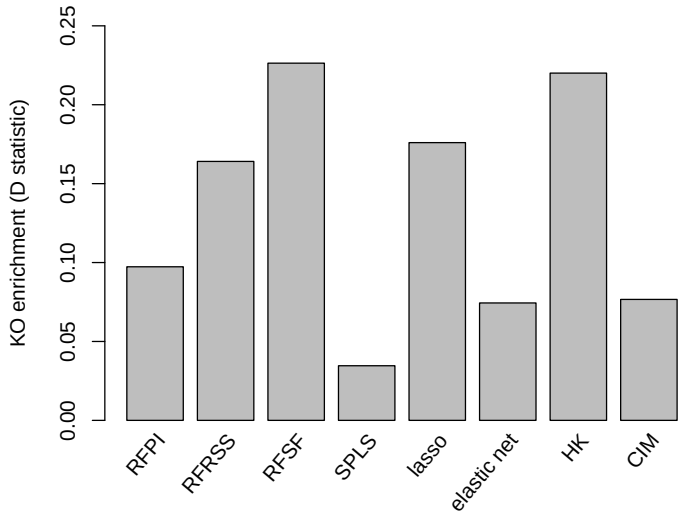


Figure 2.6 Enrichment of high-scoring eQTL for mutant expression changes. We used large-scale loss-of-function gene expression studies in yeast to determine whether high-scoring eQTL were near genes that, when mutated, perturbed the expression of the target gene. All methods showed significant enrichment for eQTL causing large expression changes when genes proximal to the eQTL are mutated, though the degree of enrichment varied widely. RFSF showed the most significant enrichment with $P = 1.03 \times 10^{-99}$.

2.4 Discussion

2.4.1 High-throughput data make functional benchmarking of eQTL mapping methods possible

Augmenting eQTL with independent information has been done previously to strengthen hypotheses suggested by the eQTL data (Wessel et al., 2007; Wu et al., 2008; Ghazalpour et al., 2006; Suthram et al., 2008). Although these applications demonstrate a certain degree of correspondence between eQTL data and external data sources, and imply that such correspondence is desirable in an eQTL mapping method, no benchmarks based on the systematic recovery of biological information have been proposed and applied to a wide variety of mapping methods and data sets.

Validating the performance of mapping methods is important not only for those whose analysis ends with an eQTL map, but also for more sophisticated algorithms such as Lirnet (Lee et al., 2009) and Geronemo (Lee et al., 2006) which build on top of basic mapping concepts. Our analysis, combined with previously cited works that integrate eQTL with other data, show that there is indeed agreement among eQTL and data from different sources. Maximizing this agreement should be a core objective of future mapping techniques. We hope that this approach to benchmarking, in addition to traditional simulated benchmarks, will help practitioners find the appropriate method now, and lead to the development of better mapping methods in the future.

2.4.2 Multi-locus eQTL mapping methods outperform legacy methods

With few exceptions, the legacy methods — HK and CIM — stood out as the poor performers, particularly in the simulations, *cis*-eQTL proportions, and enrichment for KEGG pathway relationships. In preliminary analyses, we found related univariate mapping methods such as EM interval mapping (Lander and Botstein, 1989) and ANOVA to have performance almost indistinguishable from HK (data not shown). This observation is important because even at the time of this writing there are still eQTL papers being published that use legacy mapping methods for their analysis (La Merrill et al., 2010; Chen et al., 2010; Wang et al., 2010b; Viñuela et al., 2010), ostensibly because the more modern methods are not as accessible. In light of our results, we expect that these studies have not exploited the full potential of the collected data. This represents a challenge for the computational community of working to promote not just the development, but also the adoption of these more advanced methods.

There is a fundamental difference in how the legacy linear methods (HK, CIM) and the multi-locus linear methods (SPLS, lasso, elastic net) score loci. The univariate mapping methods rely on a LOD score (or a P value in the case of one-way ANOVA) that expresses the significance of the estimated correlation between a single marker and the trait, resulting in thousands of individual modeling attempts per expression trait. The multi-locus methods, in contrast, assign coefficients to multiple loci in a single final model. These coefficients are then used as locus scores. The disparity in performance between the two classes of methods is likely a result of scoring by contribution to the model (multi-locus approach), rather than scoring by significance (univariate approach).

RF offers a third paradigm for scoring that is conceptually similar to the coefficient approach of the multi-locus linear methods, though distinct in implementation. Each of the three importance measures derived from RF measures a locus' average contribution in an ensemble of models. This differs from the coefficient approach in that it is a summary of multiple models, each including multiple loci, rather than

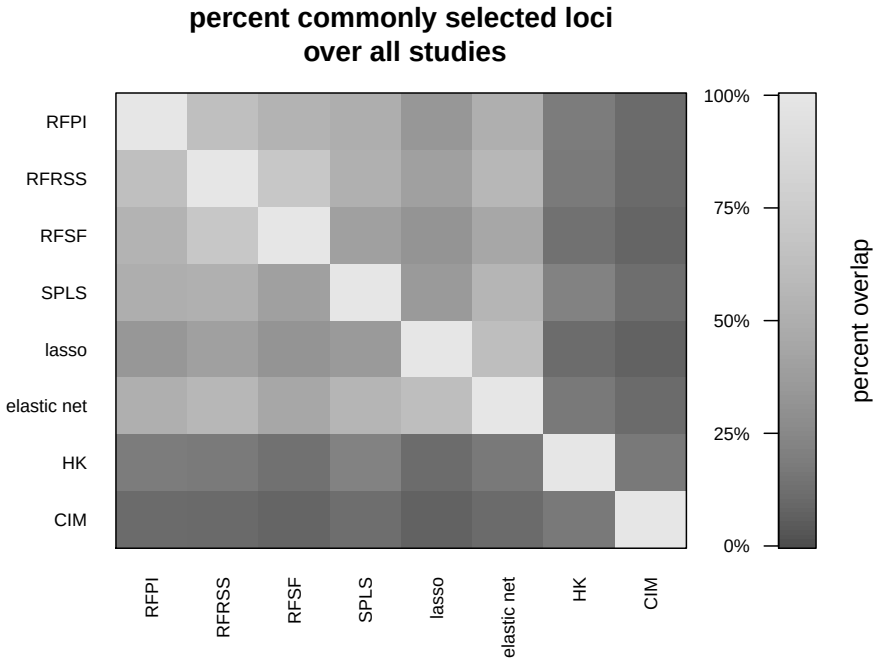


Figure 2.7 Agreement between methods expressed as the overlap of selected loci, over all experimental data sets. In general, the multi-locus approaches showed much more consistency with each other. The average percent overlap among RFPI, RFRSS, RFSF, SPLS, lasso, and elastic net was 49% (ranging from 31% to 67%), while HK and CIM had 17% of top loci in common (99^{th} percentile).

a summary of a single model including multiple loci. Additionally, the multi-locus linear methods do not implicitly allow for the inclusion of epistatic interactions in the locus scoring process, while RF does.

It should be noted that the benchmarking process described in this work did not focus on the methods' abilities for statistical inference, that is, determining whether a locus *significantly* explains an expression trait. Instead, our benchmarks focused on which methods prioritized the loci with the greatest degree of effectiveness over a large panel of data. If statistical inference is desired, appropriate permutation of the data can be performed to obtain a null distribution of scores for the chosen method, which can then be used to assess significance of the scores.

We evaluated all experimental data sets and compared the loci that each method scored in the 99^{th} percentile. In general, the multi-locus approaches showed agreement amongst themselves, with an average 49% overlap. Figure 2.7 highlights the lack of consistency between the legacy methods and the multi-locus methods, and amongst themselves.

2.4.3 Random Forests selection frequency maps the most biologically consistent eQTL

Random Forests (RF) (Breiman, 2001) is a classification and regression algorithm based on fitting an ensemble of trees. When mapping eQTL, RF fits decision trees by using markers as predictor variables, i.e., each node in a tree corresponds to a split of the population based on the genotype at the selected marker. By combining an ensemble of many diverse decision trees, RF guards against overfitting and also provides several measures of predictor variable importance. In this work, these measures of variable importance are used to map eQTL.

Although multi-locus methods in general outperformed the legacy methods HK and CIM, RFSF showed the most consistent performance overall. In the simulations and *cis*-eQTL proportion test it was among the best, and in the KEGG and mutant enrichment tests it outperformed the competitors. This finding is somewhat surprising because RFSF is virtually ignored as a variable importance measure in most applications of RF, including QTL and GWAS (Bureau et al., 2003; Lunetta et al., 2004; Bureau et al., 2005; Motsinger-Reif et al., 2008; Lee et al., 2008). Avoiding RFSF may have several explanations. For instance, it has been shown previously that RFSF can be biased. This bias manifests itself in the case of continuous or categorical predictors that vary widely in their scales or number of categories (Strobl et al., 2007). This is typically not an issue in the case of genotype data, where all predictors are categorical with the same number of categories. However, RFSF can also be biased when there is a significant degree of correlation between predictors, which is the case with genotype data. Under these conditions, RFSF preferentially selects variables (markers) with low correlation to other variables; markers in linkage disequilibrium are under-selected. In order to estimate and account for this bias, we add or subtract the deviation from the mean selection frequency observed under the null hypothesis (no association between trait and genotype data). See methods and Figure 2.8 for details.

We decided to investigate further the potential reasons why RFSF performed better than the more typically used RFPI or RFRSS. We hypothesized that perhaps RFSF picked up on smaller effects near the leaves of the trees, i.e. it is able to detect loci with very subtle effects on the trait. To demonstrate this, we use the largest data set (yeast) and grew several RFs with different characteristic tree depths. We then tested these forests with the *cis*-eQTL proportion test and the KEGG enrichment test (see methods for details). We found that increasing the depth of the trees had a modest effect on the performance of RFPI and RFRSS, with an increase in percentage of *cis*-eQTL from 22.4% to 23.5% and 24.1% to 25.6%, respectively, and an increase in D statistic (for the KEGG enrichment test) from 0.186 to 0.225 and 0.241 to 0.318, respectively. Conversely, RFSF benefited more from the deeper forests, with an increase in percentage of *cis*-eQTL from 24.3% to 27.4% and an increase in D statistic (for the KEGG enrichment test) from 0.241 to 0.361 (Fig. 2.9). In addition, we found that agreement with the linear methods (SPLS, lasso, elastic net, HK, and CIM) was at its highest when the tree growth was stopped early; similarity decreased with increasing tree depth. This effect was more pronounced for RFSF than for the other RF importance measures, which further suggests that the effects found near the leaves of the trees are connected to RFSF's superior performance (Fig. 2.10).

To further explore this idea, we performed simulations where the expression trait was a function of eight loci, two with strong effects, and six with small effects. As expected, the loci with the stronger effects were used in splits closer to the root node. The causal loci with weaker effects were used to split closer to the leaves. In these simulations, RFSF scored the weak causal loci in the 99^{th} percentile 18.3% of the time, while RFPI scored the same loci in the 99^{th} percentile only 10% of the time. These simulations also showed that RFPI is tightly coupled to a variable's proximity to the root node, while RFSF can give high scores even if the variable is not used close to the root node.

From these investigations we conclude that RFPI and RFRSS both essentially determine variable importance near the roots of the trees, and that biologically important splits further down the tree are not adequately reflected in the overall importance scores. RFSF on the other hand, recovers more biologically meaningful predictor variables (loci) when trees are grown deep, suggesting that even splits far down the tree can be reflected in this importance measure. Epistatic effects are an example of where this phenomenon is important — often genetic interactions are weak and only present in a subset of the

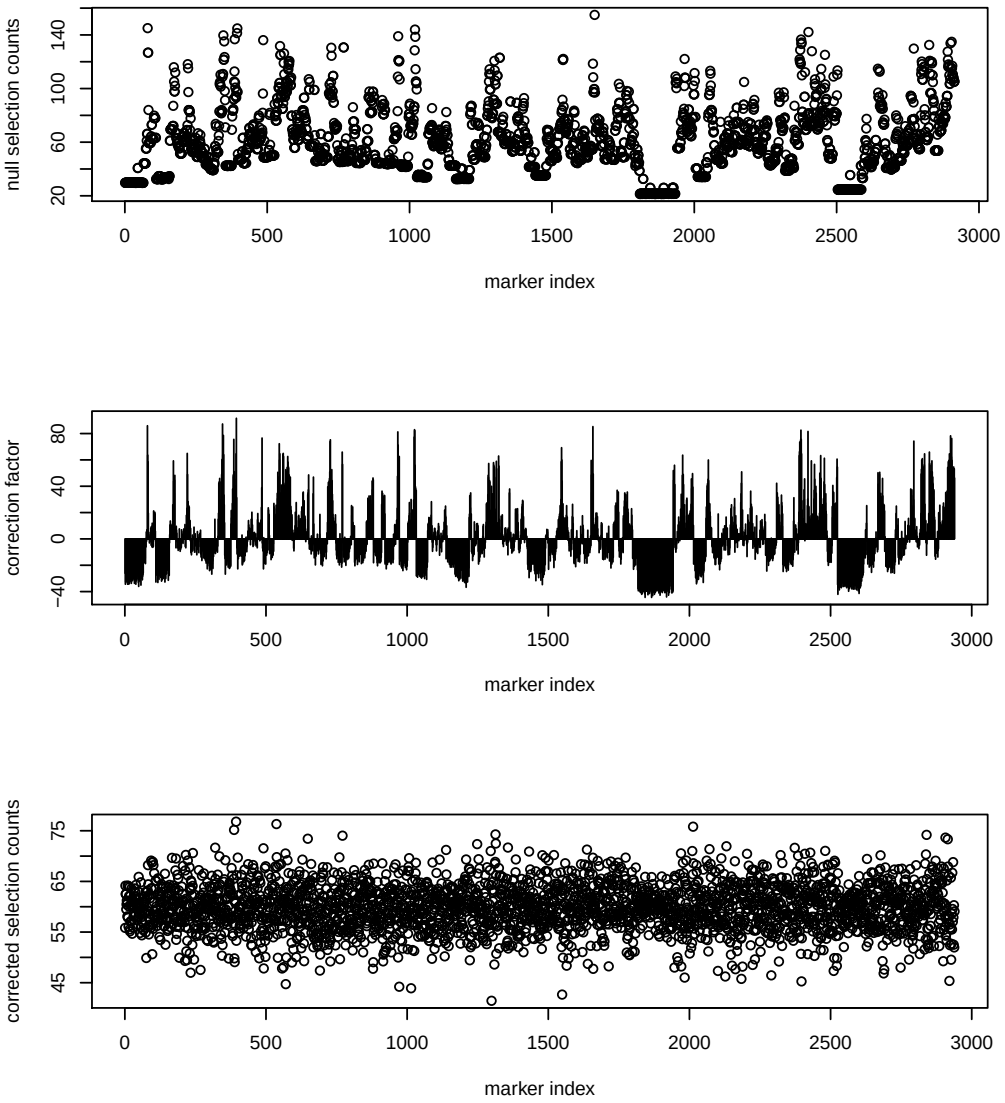


Figure 2.8 Bias estimation and correction in RFSF. Under the null hypothesis (no association between trait and genotypes), RFSF is biased towards variables with low correlation to others (top panel). The bias is estimated by fitting a forest to Gaussian noise, and a correction factor is derived by determining how much more or less frequently a marker is selected than the mean (middle panel). By subtracting the correction factor from the observed RFSF, the selection bias is removed (compare top panel to bottom panel).

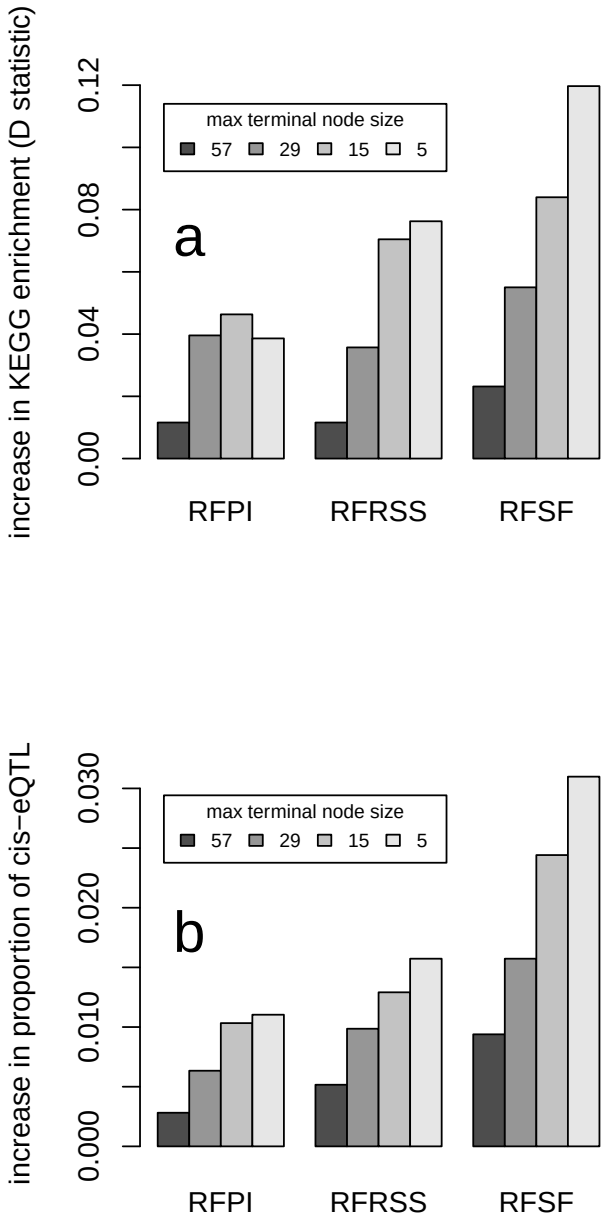


Figure 2.9 Effect of varying Random Forests tree depth on performance. The effect of varying Random Forests tree depth on performance as measured by the distributional deviation of the enrichment *P* values from the uniform distribution (A) and the percentage of expression traits with a *cis*-eQTL (B). Smaller node sizes correspond to deeper trees. The permutation importance and RSS importance improve modestly with deeper trees, whereas selection frequency shows more marked improvement with deeper trees. The improvement is measured with respect to forests that stop after the root split (`nodesize` 114).

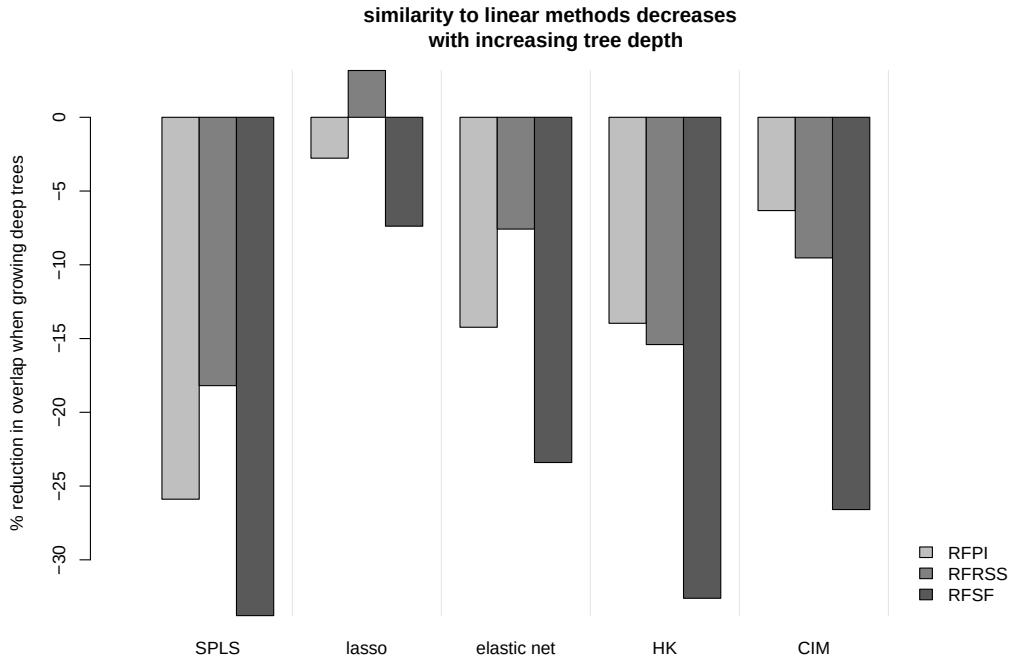


Figure 2.10 Overlap of RF and linear methods while increasing RF tree depth. In general, deeper trees caused the RF importance measures to diverge from the linear methods in terms of which loci were given the top scores. The effect is particularly pronounced for RF selection frequency (RFSF).

population. Such conditional effects are likely to manifest themselves deeper in the trees. RFSF is an attractive measure in these situations.

Because of its demonstrated performance advantages in finding biologically relevant loci, its ability to implicitly consider epistatic interactions, as well as its straightforward and readily available implementation, we recommend using Random Forests for eQTL mapping. We have prepared a short tutorial and example R code demonstrating mapping eQTL with the bias-corrected selection frequency at http://cellnet.biotec.tu-dresden.de/RFSF.

2.4.4 Marker density and analysis strategy

In this work we examined studies with genotype data in the range of thousands of markers. With the advent of next-generation sequencing and other ultra high-throughput methods, we expect to see more and more studies with hundreds of thousands, millions, or even tens of millions of SNPs. We wish to put the presented work in context by drawing a distinction between filtering methods, mapping methods, and explicit models (Fig. 2.11).

The state of computer hardware at the time of this writing makes the multi-locus methods presented here impractical for exhaustive evaluations of data sets with millions of SNPs and tens of thousands of expression traits. The current solution to this problem is to filter the SNPs to a more tractable number using univariate tests or expert knowledge (Rudd et al., 2005; Jegga et al., 2007; Chan et al., 2009). Considering the joint effects of markers at this point is generally a fruitless effort, given the astronomical number of potential combinations and the problem of dealing with false positives.

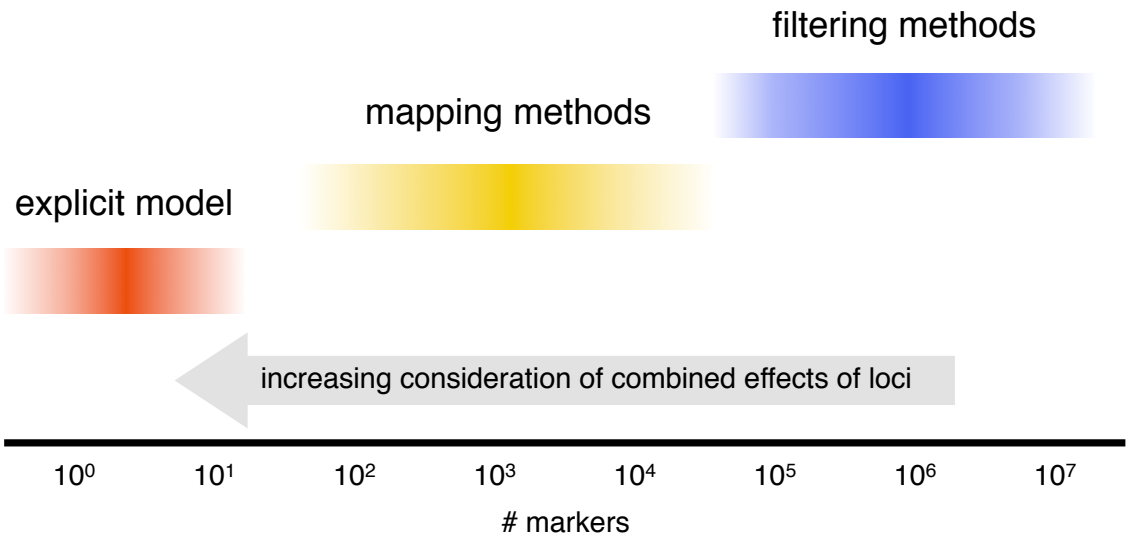


Figure 2.11 Relationship between SNP density and analysis strategy for eQTL data. The current state of computer hardware allows little if any consideration of joint effects of markers when millions of SNPs are considered for tens of thousands of expression traits. Simple univariate tests or expert knowledge are often employed to reduce the number of considered SNPs to a range where mapping methods may be used and increased attention may be given to the interplay between loci. In the optimal case, successful application of mapping methods in many populations will yield an explicit model of the expression trait in terms of a smaller number of genetic loci, optionally including environmental effects.

As the number of markers considered falls into the tens of thousands, the problem transitions from filtering to mapping. Mapping is a combination of modeling and feature selection, and the methods we explored in this work address the mapping problem. Here the interplay between loci becomes important for accurately identifying the causal regions that should be included in an explicit model of the trait.

Once causal loci have been identified reliably and the relationships between them have been characterized (additive vs. dominant, epistatic vs. additive, etc.), one can construct a linear model, usually consisting of a handful of terms, that accurately describes the trait as a function of the genetic state of the organism. Such an explicit model, though desirable, is rarely attained.

2.4.5 Implications for related mapping problems

Most of the conclusions from our work have implications beyond eQTL mapping. Ideally, the concept of a knowledge-driven benchmark could be used for any physiological trait, but our approach depends on a fairly detailed knowledge of the molecular mechanisms underlying the mapped trait. Neither our notion of measuring the enrichment of regulator-target gene groups in common pathways, nor our counting of *cis*-eQTL is immediately extendable to physiological traits. Still, taken together, the evidence from this study indicates that QTL mapping — whatever the trait — should be performed using a multi-locus method. Using univariate methods such as HK will lead to severe underexploitation of the data.

Some of the more specific conclusions from our work will need further validation in other organisms and populations. For example, the study populations used here all had roughly a 50/50 distribution of two possible alleles at each marker. Human populations are characterized by very uneven distributions of SNPs, where minor alleles can be extremely rare in a given population. Such a change in the char-

acteristics of the data could influence the ranking of the individual methods. However, such fluctuations in the individual rankings are still unlikely to affect the general conclusion that multi-locus methods produce more informative results than univariate methods, even in GWAS and linkage studies in outbred populations (Cordell, 2009; Phillips, 2008; Carlborg and Haley, 2004; Moore, 2003; Schadt et al., 2005).

Finally, in this work we observed the expected relationship between study size and power to detect biologically interesting loci. We explored this phenomenon explicitly by taking subsets of decreasing sample size from the hippocampus study, and then comparing two representative methods – here RFSF and HK – using the *cis*-eQTL and KEGG enrichment benchmarks. The results are depicted in Figure 2.12 and clearly show that while both methods show improvements with additional samples, it is RFSF, the multi-locus method, that shows consistently better performance, regardless of the sample size. This suggests that even in studies with small sample sizes, multi-locus approaches are preferable to single-locus methods.

2.5 Author contributions and acknowledgements

Andreas Beyer and I conceived the benchmarks and I performed the computational work described in this chapter. Our collaborators Rudi Alberts and Klaus Schughart provided the lung and T-cell data sets described here, as well as valuable input for a related manuscript. In addition, we thank Rupert Overall and Gerd Kempermann (both CRT, Dresden), as well as Rob Williams, University of Tennessee, Memphis, USA, for providing us eQTL data and for help with the data pre-processing. We thank the Center for High-Performance Computing, TU Dresden for providing computational resources.

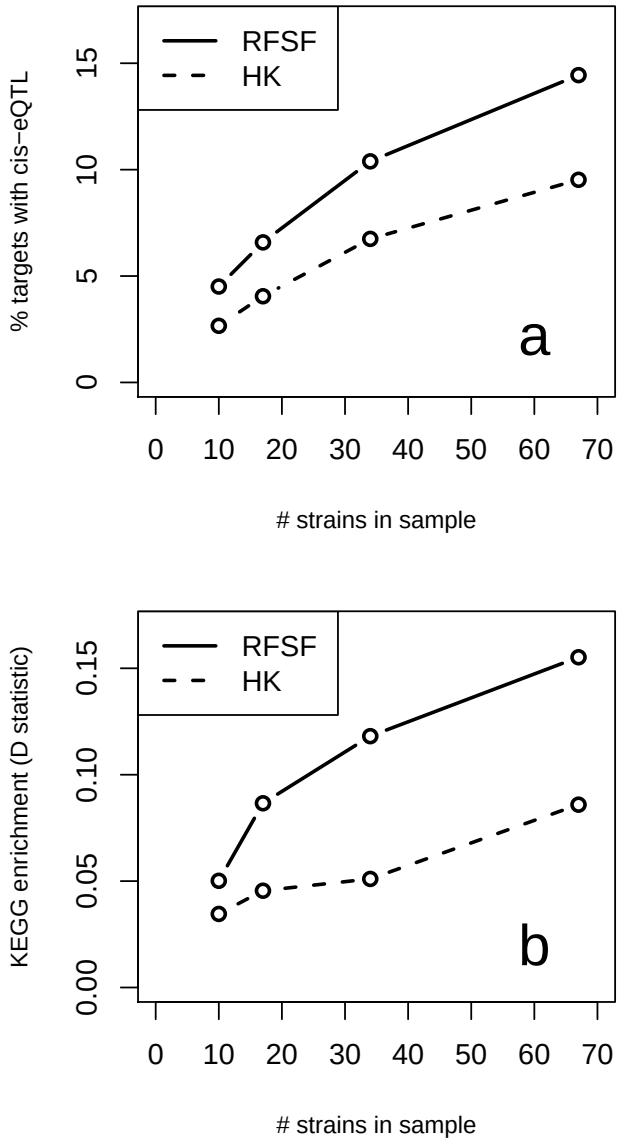


Figure 2.12 Relationship between sample size and ability to recover biologically relevant loci. Subsets of decreasing size (67, 34, 17, and 10 strains) were taken from the hippocampus eQTL study and eQTL were mapped using RFSF and HK. Performance was evaluated with the *cis*-eQTL and KEGG enrichment benchmarks. Both RFSF and HK improved performance when additional strains were added, though the performance RFSF was consistently better than HK in both benchmarks for all sample sizes.

Chapter 3

Epistasis controlling gene expression

The following publications and presentations relate to the work presented in this chapter:

1. **Michaelson, J. J.** and Beyer, A. Transcriptional regulatory contexts and epistasis among schizophrenia risk genes. (in preparation)
2. **Michaelson, J.J.** and Beyer, A. Transcriptional regulation in schizophrenia. Systems Biology: Networks 2010, Hinxton, UK.
3. **Michaelson, J.J.** and Beyer, A. Molecular mechanisms in schizophrenia uncovered with systems genetics. Systems Biology of Human Disease 2010, Boston, USA.
4. **Michaelson, J.J.** and Beyer, A. Identifying genetic interactions involved in adult neurogenesis. CRTD Bioinformatics Symposium 2009, Dresden, Germany.
5. **Michaelson, J.J.**, Ackermann, M., and Beyer, A. Uncovering interactions with Random Forests. useR! 2009, Rennes, France.
6. **Michaelson, J.J.** and Beyer, A. Exploring the regulatory architecture of neurotransmitter receptors with Random Forests. INCF 2009, Pilsen, Czech Republic.

3.1 Introduction

Epistasis is a genetic-phenotypic phenomenon where a gene's contribution to a trait does not occur in isolation; rather, it is dependent on the genetic background of the organism (Carlborg and Haley, 2004). This is in contrast to the prevailing public perception of genes conferring traits unilaterally, e.g. media reports of the "cheating gene", the "fat gene", or the "gay gene". Epistasis is usually interpreted as an interaction between genes, meaning that the effect of one gene can be changed depending on the state of another gene (Fig. 3.1). It is a relationship that suggests a more intimate molecular connection between genes than additivity (independent contributions to a trait) does. Indeed, researchers have used it as a tool to reconstruct molecular pathways in model organisms (Phillips, 2008; Tong et al., 2004; Schuldiner et al., 2005; Hannum et al., 2009; Costanzo et al., 2010). In addition, epistasis has received increased attention with the advent of high-throughput human genetics, and offers a framework for interpreting immense variation, such as with personal genetics (Moore and Williams, 2009), as well as for understanding complex

diseases (Shao et al., 2008; Carlborg and Haley, 2004). As we come to better understand epistasis and its meaning, we will be better able to diagnose and treat disease on a personal level.

Concurrent with the increased interest in epistasis on the experimental side, a wide variety of methods to detect epistasis has been developed by computational scientists (Nelson et al., 2001; Ritchie et al., 2001; McKinney et al., 2006; Dong et al., 2008; McKinney et al., 2009). A core challenge for any epistasis detection method is that the required runtime (and storage) for an exhaustive test for all pairs of genes/loci is proportional to the square of the number of genes/loci. This is trivial for problems involving thousands of loci, but is intractable for much larger problems, particularly when multiple traits are investigated, as is the case with eQTL. Most methods necessarily try to circumvent this challenge by using either greedy or stochastic search strategies without sacrificing the recovery of interesting interactions.

Another methodological challenge lies in reconciling the statistical and biological meanings and interpretation of epistasis (Cordell, 2002). Because of the relatively small amount of high-throughput data on epistasis, most methods are developed for and benchmarked with synthetic data. These toy models generally reflect the statistical definition of epistasis, though their generalizability to real data is debatable. The result is that while many methods perform well on paper, their performance in practice is hard to estimate.

We have previously found Random Forests (RF) (Breiman, 2001) to be quite effective in mapping quality eQTL (Michaelson et al., 2010). In this work, we propose an extension to RF to explicitly define epistatic interactions. RF has been used previously in similar contexts because of its ability to capture (with its importance measure) variables involved in interactions (García-Magariños et al., 2009; Kim et al., 2009b; McKinney et al., 2009). Our approach aims to go beyond merely identifying loci involved in interactions, but identifying the interactions themselves. This is accomplished by examining the forest structure for decision motifs that suggest dependence between splitting variables. A similar approach was taken using RF with binary traits, using contingency tables at nodes in the forest (Wang et al., 2010c). Our approach uses regression RFs (i.e. the trait is quantitative), and though similar in aim, differs appreciably in the implementation. The result is an approach that shows superior performance both in simulated data and in real data, compared to an exhaustive LOD-based method. Finally, we apply this method to find examples of epistasis regulating the expression of schizophrenia risk genes.

Definition of open problem

How can epistasis be efficiently discovered among millions of locus pairs and tens of thousands of traits? How can competing methods be benchmarked using real data?

3.2 Materials and Methods

3.2.1 RF split asymmetry

In the work presented here, epistatic relationships are found in the structure of a regression Random Forest (RF) by looking for a phenomenon we call *split asymmetry* (Fig. 3.2). Consider a sequence of two decision splits in a tree, involving two variables, first X_A and then X_B. This sequence may occur anywhere in the tree – near the root, leaves or somewhere in the middle – and its location may vary from tree to tree. After splitting on X_B, there will be some difference in means between the values in its left and right

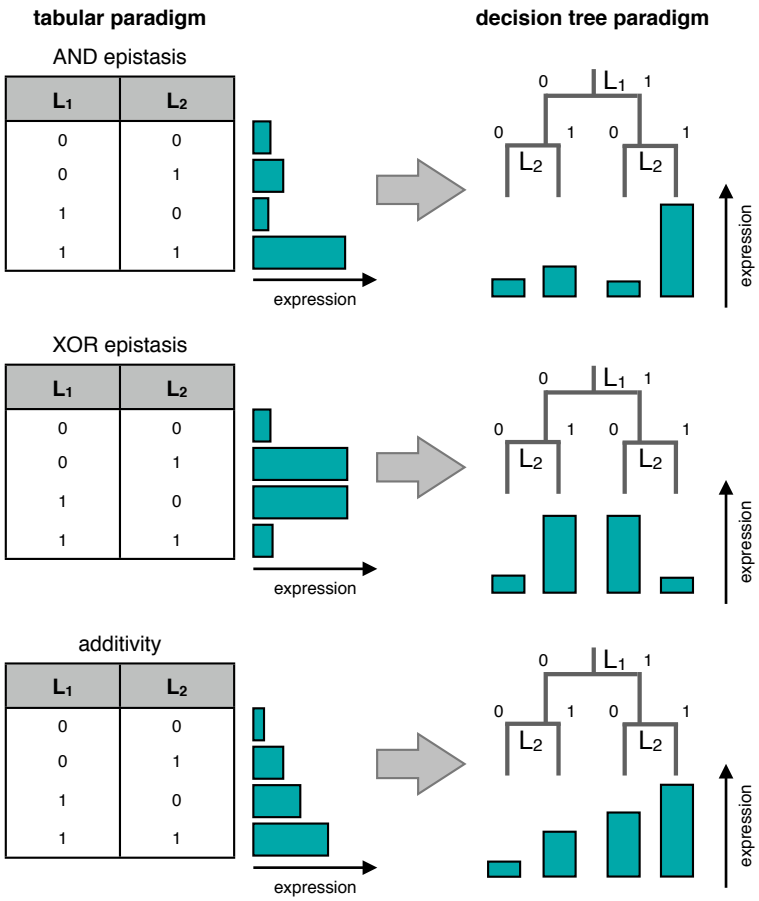


Figure 3.1 Conceptual comparison of epistasis and additivity. In this example we assume only two alleles: 0 and 1. If the effect of a locus on expression depends on the state of another locus, these loci interact, or are said to be in epistasis. Conversely, if the loci make contributions to the trait independently of one another, these loci have an additive effect. Both interactions and additivity can be viewed in a tabular or a decision tree paradigm, and both are shown here.

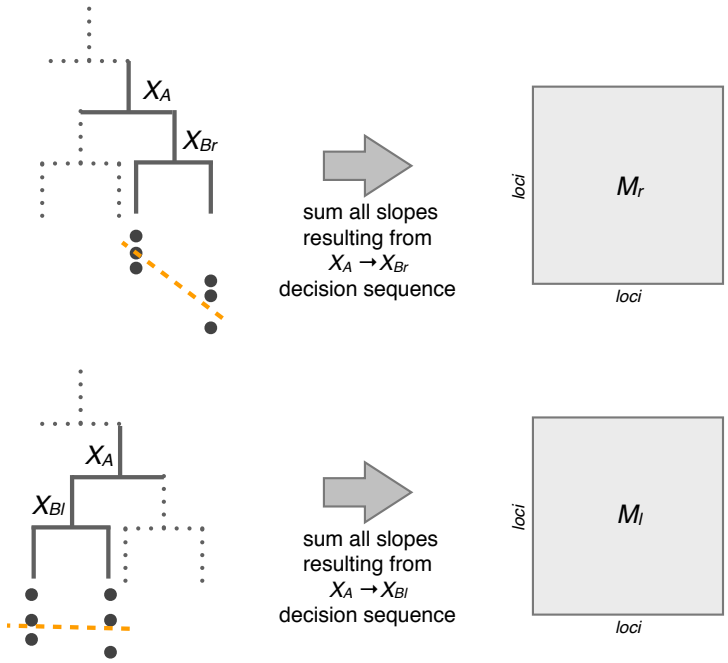


Figure 3.2 Searching Random Forest structure for splits showing asymmetry. In this representation, the decision sequences $X_A \rightarrow X_{Br}$ and $X_A \rightarrow X_{Bl}$ lead to different characteristic slopes, hence the split sequence $X_A \rightarrow X_B$ is *asymmetric*. The matrices M_r and M_l are used as described in equations 3.1 through 3.5, resulting in a score that indicates epistasis between loci.

daughter nodes. We can view this difference between means as a slope. If the mean of the right daughter is greater than that of the left daughter, the slope is positive, and in the opposite case the slope is negative. If there is no dependency between X_A and X_B when considering the response values, we would expect that the slope after splitting on X_B would be the same regardless of whether X_B splits on data in the left or right daughter node of the X_A split. On the other hand, if there is a dependency between X_A and X_B, we expect that the decision at X_A will influence the outcome of the split at X_B, thus resulting in different slopes for X_{Bl} (split on left daughter of X_A) vs. X_{Br} (split on right daughter of X_A). Given this context, we say that a split is *asymmetric* in a sequence of variables with dependencies, and a split is *symmetric* in a sequence of variables with no dependencies.

All such slopes involving all 2-variable decision sequences encountered in the forest are summed according to their "sidedness", leading to two square matrices: M_l for the sequence corresponding to $X_A \rightarrow X_{Bl}$, the "left" matrix, and M_r for the sequence corresponding to $X_A \rightarrow X_{Br}$, the "right" matrix (Fig. 3.2). In both matrices, the row indicates the first variable in the decision sequence, and the column indicates the second variable in the sequence.

In cases of extreme dependencies, X_{Br} (for example) might be used frequently, yet X_{Bl} might never be suitable as a splitting variable, and therefore might not occur at all in the forest (leading to an entry of 0 in M_l). It should also be noted that the *individual* slopes will be influenced by the stochastic characteristics of RF – in particular the bootstrap sample of data used to fit the tree in question. However, the aggregated slope for a variable pair will be more robust. In any case, the magnitude of the absolute difference of the aggregated slopes (a matrix D) is an indicator of the strength of the dependency between the involved splitting variables. Note: while $|M|$ traditionally denotes the determinant of M when M is a matrix, for

convenience we use it here to mean the matrix resulting from taking the absolute values of the entries in M.

$$D = |M_r - M_l| \tag{3.1}$$

We introduce a few modifications to D to further refine the score it represents. First, we'll subtract as a penalty the mean absolute slope, here the matrix S:

$$S = \frac{|M_r| + |M_l|}{2} \tag{3.2}$$

$$D' = D - S \tag{3.3}$$

We additionally constrain negative values to be 0, so that only pairs whose difference in slope exceeds the average magnitude are considered.

$$D''_{ij} = \begin{cases} 0 & \text{for } D'_{ij} \leq 0 \\ D'_{ij} & \text{for } D'_{ij} > 0 \end{cases} \tag{3.4}$$

Finally, we will take the minimum of the corresponding values D''_{ij} and D''_{ji}, since purely interacting variables (i.e. without an additive effect), will be "order-agnostic", meaning that the sequences $X_A \rightarrow X_B$ and $X_B \rightarrow X_A$ should both be asymmetric; we take the minimum of the two scenarios to be conservative. In practice, this has reduced the number of false positives encountered. This final epistasis score is stored in a (symmetric) matrix E:

$$E_{ij} = \min\{D''_{ij}, D''_{ji}\} \tag{3.5}$$

Significance of the values in E may be computed by obtaining a null distribution (either empirical or parametric). Functions implementing these operations can be found in Appendix A and a detailed tutorial of their use can be found in Appendix C. In addition, we modified the reference implementation of Random Forests such that the mean of the out-of-bag (OOB) data at each node in the forest is recorded, to facilitate the calculation of split asymmetry using test set data.

3.2.2 Simulations

Three epistatic scenarios were simulated: 2-way XOR, 2-way AND, and 3-way AND. Logical combinations of the genotypes from the yeast data from (Brem and Kruglyak, 2005) were constructed according to the type of epistasis to be simulated. Traits were simulated using the `simtrait` function (see Appendix A). For each type of epistatic model, 50 traits were simulated, with each trait using different randomly selected markers from the yeast genotype data.

Random Forests were constructed using 5,000 trees, and a node size of 3 was used. All other parameters were left at their default values.

Performance in recovering the causal loci was assessed with ROC curves (Fig. 3.3) for both RF split asymmetry and a 2-dimensional scan as implemented in the `qtl` package for R (Broman et al., 2008). The LOD score (for interaction) was used to score the exhaustive search, and $1 - P$ was used in the case of RF split asymmetry (where P is the P value of the RF split asymmetry score, obtained by using an

empirical null distribution).

3.2.3 Yeast data

Yeast eQTL data from (Brem and Kruglyak, 2005) was used and interaction scores were calculated using RF split asymmetry and the pairwise eQTL scan available in the `qtl` package for R. In both cases, the scores for a pair of markers were maximized (for LOD scores in the case of the exhaustive search) or minimized (for P values from RF split asymmetry) over the transcripts. Thus the final square matrix for each approach represented the best score over all transcripts for each pair of markers. For RF split asymmetry, a bias correction similar to (Michaelson et al., 2010) (i.e. bias correction on the matrix rather than a vector of eQTL scores) was applied to the matrix result of each transcript, before significance was calculated.

Two reference sets were used as standards to generate ROC curves for the two epistasis scoring approaches: (Costanzo et al., 2010) and (Schuldiner et al., 2005). For the purposes of our comparison, we made no distinction between so-called positive and negative genetic interactions (indeed, preliminary tests showed no appreciable difference in performance between these classes when handled individually). Since the eQTL data maps to locus (i.e. marker) resolution and the gold standard sets map directly to genes, we mapped genes to their closest marker. Thus, a pair of markers interact if any of their mapped genes interact with each other as demonstrated in the previous studies.

3.2.4 Mouse hippocampus eQTL data

RF split asymmetry was applied to eQTL data from a study of the murine hippocampus (Overall et al., 2009) to find epistatic interactions relevant to schizophrenia. Human schizophrenia-associated genes were selected based on a combination of expression and text association evidences (see Chapter 4 methods for details), and their murine orthologs were used here as both targets and (with their corresponding genetic loci) as potential interacting regulators. Scoring was performed as described for the yeast data, and interactions with an $FDR < 0.05$ were considered as significant.

3.3 Results

3.3.1 Simulations

We examined three models of epistatic interactions: 2-way XOR, 2-way AND, and 3-way AND. Each configuration was repeated 50 times, each time using different (randomly selected) causal markers from the yeast genotype data (Brem and Kruglyak, 2005). As a baseline method for comparison, we selected an exhaustive two-locus approach, implemented via the `scantwo` function in the `qtl` package for R (Broman et al., 2008). This resulted in a square matrix of LOD scores, indicating the strength of the evidence for interaction between the i^{th} and j^{th} markers.

The results (Fig. 3.3) indicate that the exhaustive approach outperforms RF split asymmetry in the XOR scenario, while RF split asymmetry fares better in both the 2-way and 3-way AND scenarios.

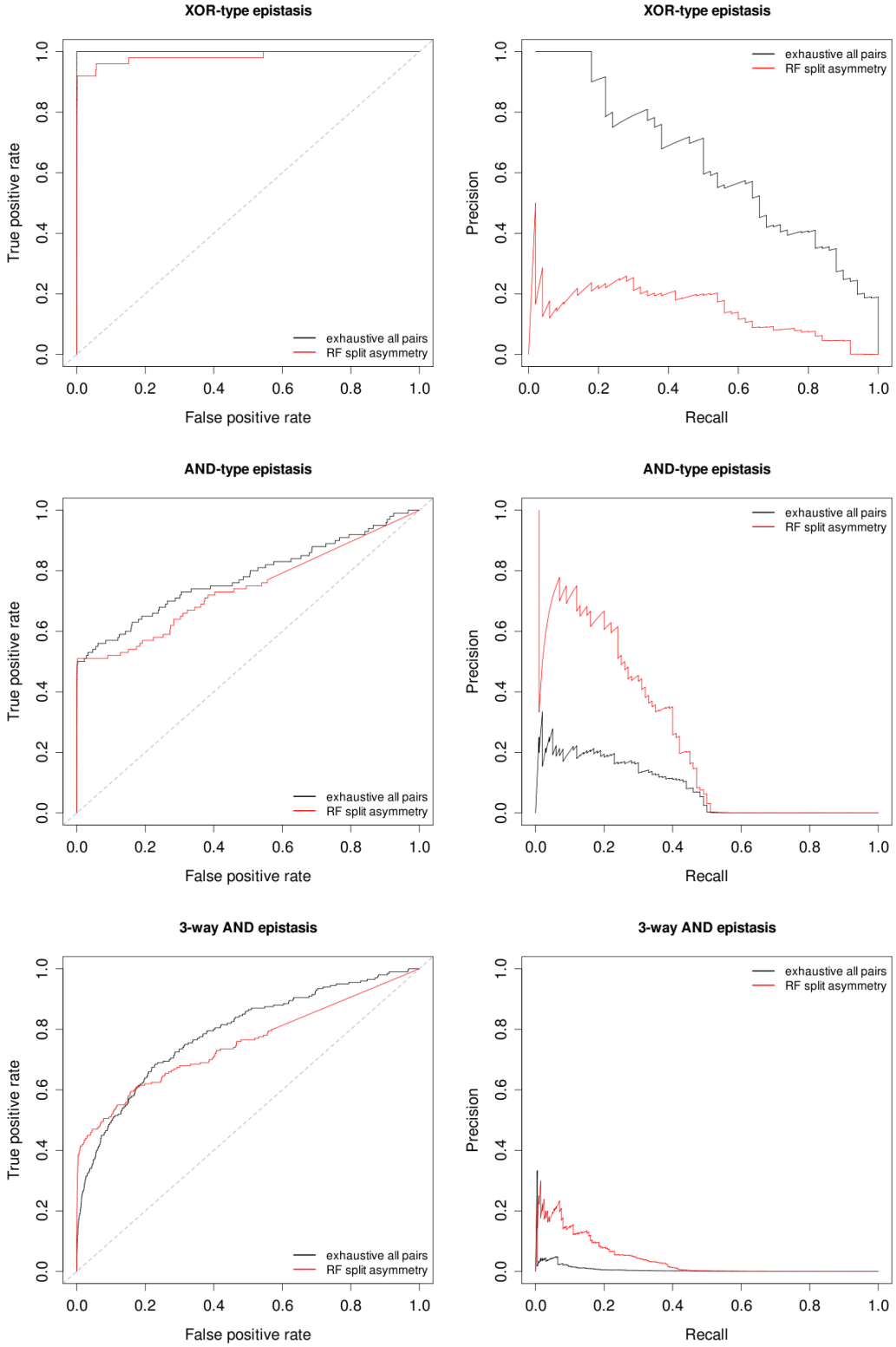


Figure 3.3 Performance comparison of RF split asymmetry vs. an exhaustive all-pairs approach. Three different interaction scenarios were investigated: 2-way XOR, 2-way AND, and 3-way AND. RF split asymmetry performed better in both AND scenarios, while the exhaustive approach performed better in the XOR scenario.

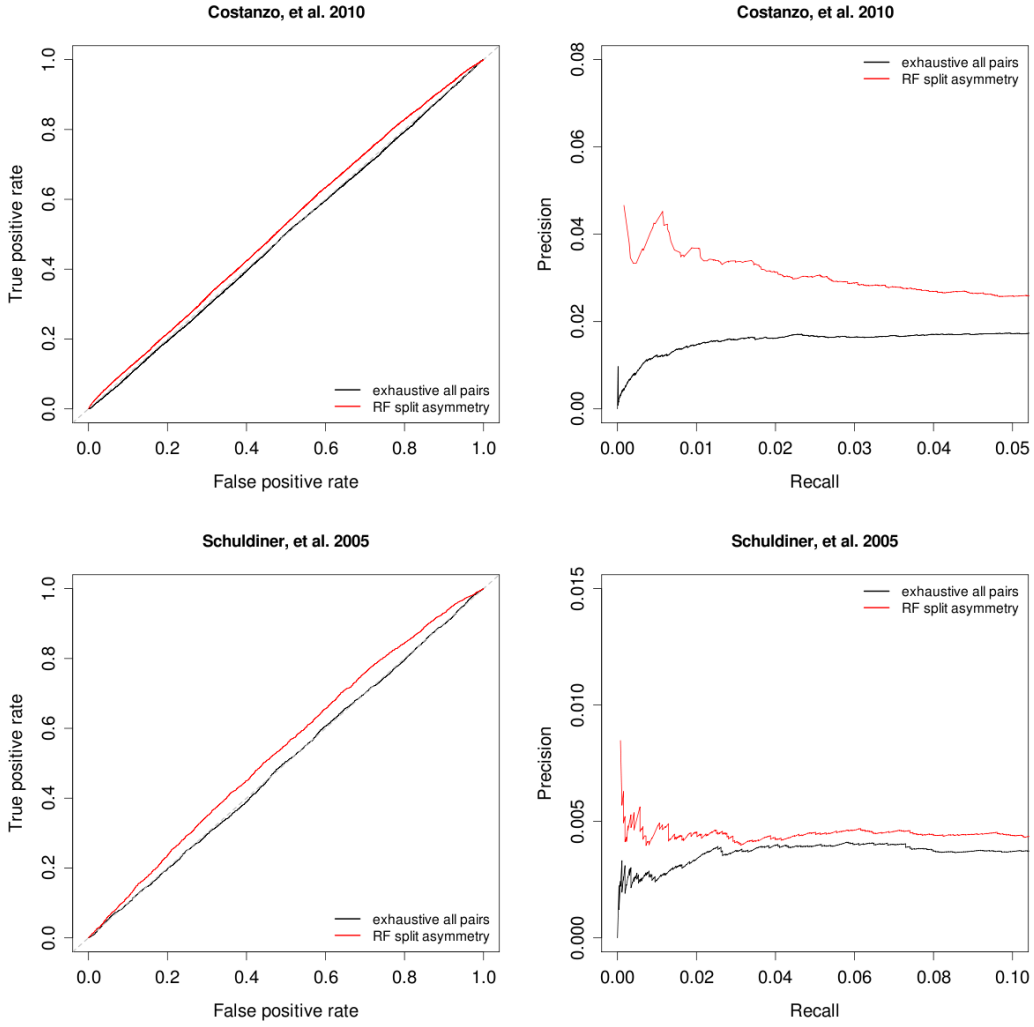


Figure 3.4 Performance comparison of RF split asymmetry on yeast eQTL data, vs. an exhaustive all-pairs approach. Here we investigated how well each method was able to recover interactions found in engineered networks (Schuldiner et al., 2005; Costanzo et al., 2010) using data from a natural diversity study (Brem and Kruglyak, 2005). In both methods, the recovery of interactions was very small, however, RF split asymmetry peformed better in both data sets.

3.3.2 Epistasis in yeast eQTL data

To make a more biologically meaningful comparison between methods, we used yeast eQTL data from (Brem and Kruglyak, 2005), together with interaction data from (Costanzo et al., 2010) and (Schuldiner et al., 2005) as the gold standard sets (Fig. 3.4), to see which method could better recover the known interactions using the eQTL data. Here RF split asymmetry shows better recovery of interacting locus pairs, though in general it shows only very modest agreement with the studies of "engineered" epistasis. The exhaustive approach leads to scoring that is comparable to random scoring.

3.3.3 Regulatory epistasis among schizophrenia risk genes

To assess whether RF split asymmetry could find interesting interactions in practice, we examined the transcriptional regulation of 334 murine orthologs of human genes with evidence suggesting a role in schizophrenia. We used eQTL data (expression and genotypes) from (Overall et al., 2009), and looked for

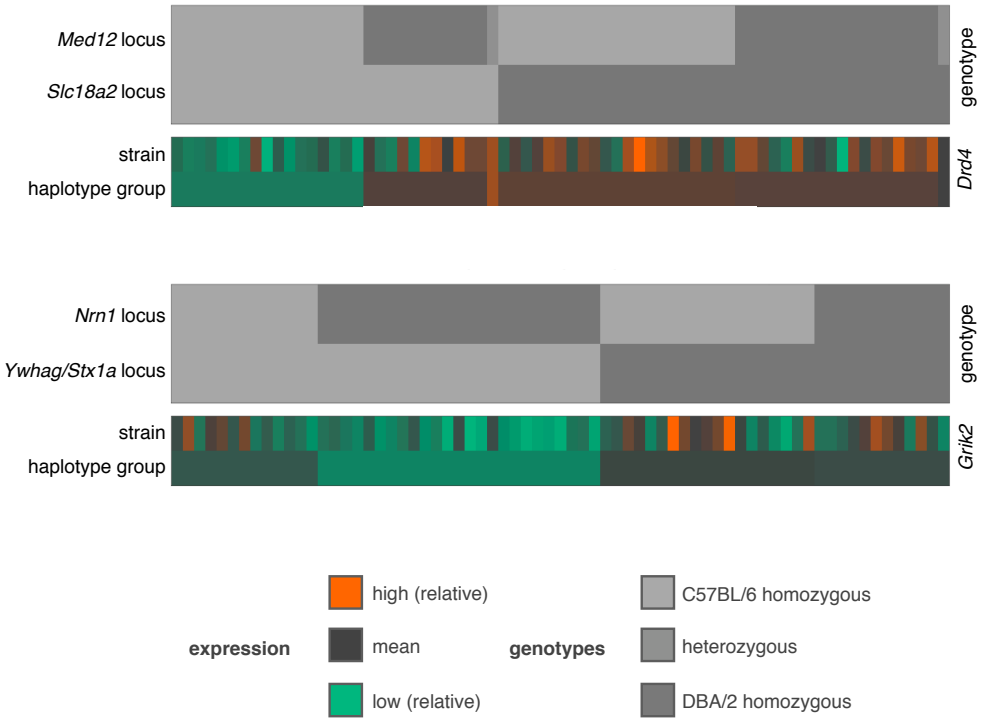


Figure 3.5 Two examples of epistasis between loci harboring schizophrenia risk orthologs that control the expression of mouse orthologs of schizophrenia risk genes. Expression of the dopamine receptor *Drd4* (top) is significantly regulated by an interaction between loci containing other genes with known connections to dopamine signaling: *Med12* and *Slc18a2*. Likewise we found that expression of the glutamate receptor *Grik2* (bottom) was significantly regulated by an interaction between the *Nrn1* locus and the locus containing *Ywhag* and *Stx1a*. Both glutamate and dopamine signaling pathways have been implicated in the etiology of schizophrenia, as well as its treatment.

interactions between loci containing one or more of these 334 genes. In this way we treated the 334 genes as both targets and potential epistatic regulators. We found significant ($FDR < 0.05$) epistatic regulation for 14 targets (Table 3.1). Of these, we found two that were of particular interest (Fig. 3.5).

3.4 Discussion

3.4.1 Performance in simulations and yeast data

In order to judge how well a new method performs, it is helpful to first look at simulated examples of obvious models. Simulations, though always an oversimplification of the actual biology at work, offer the most pure form of ground truth, since it is synthesized. We selected three types of interactions, 2-way XOR, 2-way AND and 3-way AND. The XOR and AND refer to how the 0/1 genotypes from the yeast data were combined (with a coefficient and gaussian noise) to form a quantitative trait. Examples of these types of interactions are depicted in Figure 3.1. XOR interactions present a problem for greedy and stochastic searches (such as RF) because there is no main effect in either of the interactors that could be used to guide the search in the right direction. Since RF recursively selects the best marker of a random subset, there needs to be at least a small main effect to warrant the selection of the first marker involved in an

target	interactor 1	interactor 2	P
Kcnn3	Hcrtr1	Opn5	$P < 10^{-9}$
	Htr6	Opn5	1.43×10^{-6}
Nts	Glul|Rgs16	Chl1	$P < 10^{-9}$
Rgs9	Tac1	Rgs9	7.15×10^{-7}
	Ace	Strn	1.43×10^{-6}
	Rgs9	Strn	2.67×10^{-6}
Slc1a1	Uqcrc1	Adra2a	7.80×10^{-7}
Atp5a1	Slc6a4	Adra2a	$P < 10^{-9}$
Drd4	Slc18a2	Med12	9.75×10^{-7}
	Slc18a2	Pcdh11x	1.50×10^{-6}
Nr4a2	Chrna4	Grm7	$P < 10^{-9}$
Adra2a	Sirpb1a|Sirpb1c|Gm5150|Gm9733	Wwc1	7.80×10^{-7}
	Slc32a1	A2bp1	7.15×10^{-7}
Zdhhc14	Tuba1b|Tuba1a	Med12	6.50×10^{-8}
Sirt5	Uqcrc1	D15Ertd621e	7.80×10^{-7}
Grik2	Stx1a|Ywhag	Nrn1	6.50×10^{-8}
Immp2l	Hivep2	Npas3	9.75×10^{-7}
	Ahi1|Fam54a	Npas3	9.75×10^{-7}
Csf2ra	Il10	Slc1a4	1.24×10^{-6}
	Rtn4	Olig2	1.43×10^{-6}
Ddo	Il10	Apba2	$P < 10^{-9}$

Table 3.1 Mouse orthologs of schizophrenia risk genes whose expression is regulated by epistasis between genetic loci harboring schizophrenia risk genes. All interactions had $FDR < 0.05$.

interaction. After the first marker in an interaction is selected, finding the second is more likely since it will be an optimal split (i.e. when conditioned on the first marker). If there is no main effect, it is less likely that RF will "stumble upon" the interaction. This is where an exhaustive approach performs well, as shown in Figure 3.3. However, although XOR interactions make frequent appearances in simulations (Dong et al., 2008; McKinney et al., 2009) , evidence of their prevalence in biological organisms remains scarce. On the other hand, AND-type interactions are the type typically encountered in the literature when biological epistasis is presented. RF split asymmetry outperformed the exhaustive search in the 2 and 3-way AND interaction simulations (Fig. 3.3). These results suggest that there are scenarios where RF split asymmetry is likely to miss certain types of interactions, but in the types of interactions encountered in biology, it outperforms the exhaustive approach, both in terms of CPU runtime and accurate scoring of interactions.

While simulations can give a first impression of how well a method performs, a comparison using real data is more helpful in showing which method is better in practice. Hannum and colleagues (Hannum et al., 2009) compared "natural" and "engineered" genetic interaction networks in yeast, where "natural" refers to interactions derived from eQTL data (Brem and Kruglyak, 2005) and "engineered" networks refer to interactions discovered in reverse-genetics studies such as (Schuldiner et al., 2005; Costanzo et al., 2010). We drew on this idea and used it as a benchmark to compare methods, that is, whichever method can better recover the engineered interactions using the natural data is a better method for detecting epistasis in real data. The results of our investigation using the yeast eQTL data show that RF split asymmetry outperforms the exhaustive approach (Fig. 3.4), consistent with the results of the AND-type simulations. However, neither method showed a large degree of overlap with the engineered interactions, which is not surprising considering the differences in experimental approaches and measured traits.

While these two benchmarks suggest that RF split asymmetry performs better than the exhaustive LOD-based approach, there are limitations to the method. As discussed previously, it has difficulty reliably finding interactions in which neither interactor has a main effect. Also, because of its stochastic nature, results are not exactly reproducible, and enough trees need to be used in RF construction so that results

are at least stable.

3.4.2 Epistatic transcriptional regulation among schizophrenia genes

Schizophrenia is an incredibly complex psychiatric disorder with a strong but ill-defined genetic component. Over 1,600 genetic association studies have been performed, involving almost 1,000 genes (Allen et al., 2008). Results are almost universally mixed – significant association in one population, but not reproducible in another. This lack of reproducibility across different populations fits the definition of epistasis given in (Carlborg and Haley, 2004), that the effect of a genotype on the phenotype depends on the genetic background of the organism. Indeed, epistasis is already recognized as a factor in schizophrenia (Braff et al., 2007), but it is typically explored as it affects the trait directly (e.g. in association studies) but not how it affects the transcriptional regulation of schizophrenia risk genes. Here we explore this facet of the disease by looking at eQTL data for mouse orthologs of human schizophrenia risk genes.

In our investigation of 334 genes with evidence for a role in schizophrenia, we found that 14 had significant regulation by epistasis between two or more schizophrenia loci (i.e. genetic loci containing a murine ortholog of a human schizophrenia gene). These are shown in Table 3.1. Particularly interesting are the examples of epistatic regulation of two neurotransmitter receptors: *Drd4* and *Grik2* (Fig. 3.5), because of their connections to dopamine and glutamate signaling (respectively), both pathways implicated in the etiology and treatment of schizophrenia.

The dopamine receptor *Drd4* showed significant regulation from an interaction between loci containing the risk orthologs *Med12* and *Slc18a2*, respectively. *Slc18a2* is a dopamine transporter and *Med12* is a subunit of the mediator transcriptional complex, shown to regulate the generation of dopaminergic neurons in vertebrates (Wang et al., 2006). These common connections to dopamine signaling (a critical part of the etiology of schizophrenia) among the target and its regulators further support the biological reality of this interaction.

The glutamate receptor *Grik2* was found to be significantly regulated by an interaction between a locus containing *Nrn1* and a locus containing both *Ywhag* and *Stx1a*. *Nrn1* has been shown to promote the maturation of glutamatergic synapses, and is itself regulated by glutamate neurotransmitter signaling (Naeve et al., 1997). The fact that we find it here as a regulator of a glutamate receptor suggests the existence of a regulatory loop. *Stx1a* has been shown to inhibit glutamate transport (Yu et al., 2006). As with *Drd4*, we found here evidence of functional relatedness of the target and regulators. These uncovered relationships constitute potential molecular mechanisms in the etiology of schizophrenia, and warrant further experimental investigation to better pinpoint the nature of their connections.

3.5 Author contributions and acknowledgements

I conceived of the concept of RF split asymmetry and implemented it. Andreas Beyer and I worked together to refine, test, and apply the method. I drafted the manuscript and Andreas Beyer made editorial contributions. I thank our colleagues Gerd Kempermann and Rupert Overall at the CRT, Dresden, for access to the hippocampus eQTL data. I would also like to thank Carolin Strobl (LMU, München) for constructive discussion and criticism of early implementations of the method.

Chapter 4

Trait-specific transcriptional regulatory hierarchies

The following publications and presentations relate to the work presented in this chapter:

1. **Michaelson, J. J.** and Beyer, A. Transcriptional regulatory contexts and epistasis among schizophrenia risk genes. (in preparation)
2. **Michaelson, J.J.** and Beyer, A. Transcriptional regulation in schizophrenia. Systems Biology: Networks 2010, Hinxton, UK.
3. **Michaelson, J.J.** and Beyer, A. Molecular mechanisms in schizophrenia uncovered with systems genetics. Systems Biology of Human Disease 2010, Boston, USA.

4.1 Introduction

Because of their multigenic nature, complex traits like disease frequently have hundreds or even thousands of genes associated with them in the literature. Often a precise molecular etiology of these complex diseases is lost in the long list of risk genes. Further, directly testing hypothetical disease pathways is complicated by the infeasibility of the requisite experiments in humans. Because of this, eQTL studies in mice and rats have become an attractive means for investigating transcriptional regulation in tissues and conditions that are difficult to acquire in humans. The regulatory programs found in these model organisms can shed light the potential roles of their orthologs in human disease (Chen et al., 2008; La Merrill et al., 2010).

Schizophrenia is psychiatric disorder that has a strong but ill-defined genetic component. It has been studied extensively, and has accumulated a long list of genes associated with it (Allen et al., 2008). Despite the extensive study of the disease, a definitive molecular etiology remains elusive. In our work, we used eQTL data from the mouse brain (Overall et al., 2009) to determine what regulatory roles orthologs of human schizophrenia risk genes play in relation to each other. In this way, it becomes possible to learn which risk genes are likely to be causal, and which are likely to be symptomatic of schizophrenia. This analysis was made possible by a method we developed based on the ideas behind the Kolmogorov-Smirnov test for distributional differences. We derived measures of regulatory "upstreamness" and "centrality" of schizophrenia risk genes, defining the orientation of genes within the underlying schizophrenia regulatory

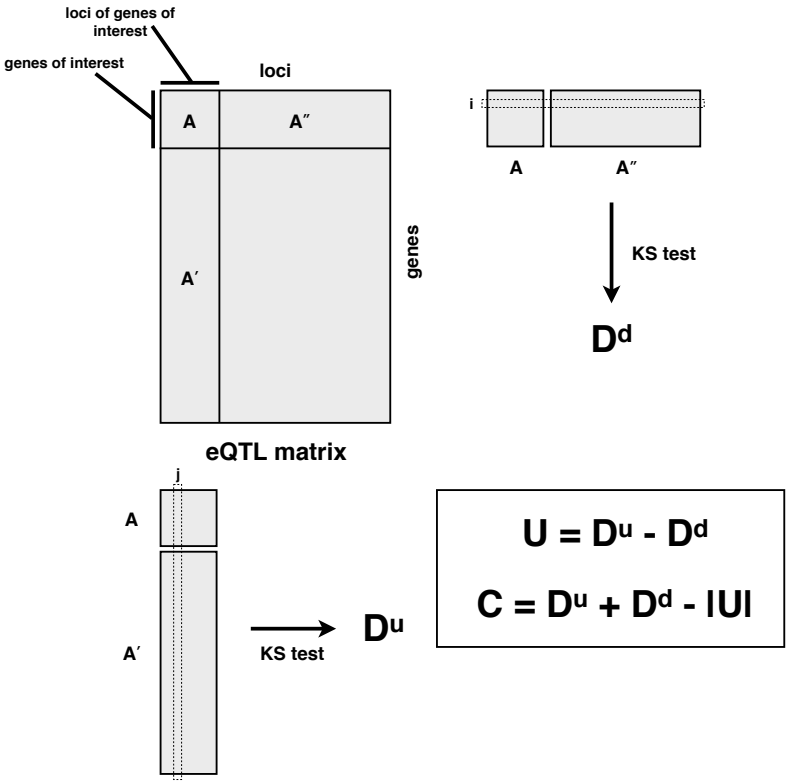


Figure 4.1 Using eQTL data and the KS test to derive regulatory upstreamness (U) and centrality (C).

network. We found that of those genes that showed a significant orientation, there was a significant correlation between their upstreamness and the reproducibility of their association to schizophrenia in different populations. This suggests that transcriptional regulators of schizophrenia genes have a more central role in the etiology of the disease, and as such deserve focused attention.

Definition of open problem

Which disease-associated genes are most likely to be causal, and which are likely to be symptomatic? How can systems genetics data be used to give clues about the etiology of a disease or the drivers of a phenotype?

4.2 Materials and Methods

4.2.1 Definition of schizophrenia-associated genes

Association of genes with schizophrenia in the literature was determined by using mappings from NCBI Entrez gene IDs to PubMed IDs, as obtained in the `gene2pubmed` file available at ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/. PubMed IDs of literature matching the search term "schizophrenia" were cross-referenced with the PubMed IDs in the `gene2pubmed` file, and matches were used to construct 2×2 contingency tables for each matching gene. The contingency tables evaluate the significance of the co-occurence of the gene with "schizophrenia" in the literature, given the overall number of times the gene appears in the literature at large. The significance of the contingency tables was assessed with Fisher's exact test.

Post-mortem brain gene expression data from (Narayan et al., 2008; Maycox et al., 2009) was combined and a Random Forests (RF) classifier (Breiman, 2001) was trained to distinguish schizophrenia patients from controls, based on the expression values of the genes measured. RF permutation importance values for each gene were obtained, and their significance was determined by constructing a null distribution of importance values by permuting the classification labels and fitting an RF.

The P values from the literature search and the expression studies were combined using Fisher's method for combining P values (i.e. the product of the P values follows a χ^2 distribution under the null hypothesis). This final P value represents a gene's association with schizophrenia. We selected all genes with $FDR < 0.01$ for further analysis. We mapped these genes to their mouse orthologs and were left with 334 genes.

4.2.2 eQTL mapping

eQTL data from (Overall et al., 2009) were mapped using Random Forests selection frequency, as described in Chapter 2.

4.2.3 Deriving upstreamness and centrality

Our novel approach for using the Kolmogorov-Smirnov (KS) test to derive regulatory upstreamness and centrality is covered thoroughly in Appendix E. Here we refer to Figure 4.1 as a conceptual representation of how upstreamness and centrality are derived.

Given a matrix of eQTL scores, where genes are rows and markers (loci) are columns, the matrix A is defined by the genes we are interested in (in this case, the schizophrenia-associated genes) and their corresponding loci. The rows in this matrix are target genes, and the columns are genomic loci (i.e. markers), thus, A_{ij} represents the effect of the j^{th} locus on the transcription of the i^{th} gene. The matrix A' is defined by the eQTL scores of all non-schizophrenia target genes at schizophrenia loci. Conversely, the matrix A'' contains eQTL scores of schizophrenia target genes at non-schizophrenia loci. We first define a statistic, D^u, that represents the tendency of a *genetic locus* to be an upstream regulator of genes in our group of interest:

$$D_j^u = D_{A'_j, A_j} = \sup_x \{F_{A'_j}(x) - F_{A_j}(x)\} \tag{4.1}$$

Where $F_{A'_j}(x)$ is the empirical cumulative distribution function of the values in the j^{th} column of A', and $F_{A_j}(x)$ is the empirical cumulative distribution function of the values in the j^{th} column of A. If D_j^u is large, it suggests that the locus corresponding to j (and by extension, the gene at that locus) is upstream of the genes defining A, but not the genes defining A' (genes not belonging to our group of interest). We note here that if multiple genes map to a locus (marker), each of the genes is assigned the corresponding value of D_j^u.

Next, we define a statistic D^d, representing the tendency of a *gene* to be downstream of genes in our group of interest:

$$D_i^d = D_{A''_i, A_i} = \sup_x \{F_{A''_i}(x) - F_{A_i}(x)\} \tag{4.2}$$

Where $F_{A''_i}(x)$ is the empirical cumulative distribution function of the values in the i^{th} row of A'', and

$F_{A_i}(x)$ is the empirical cumulative distribution function of the values in the i^{th} row of A. If D_i^d is large, it suggests that the gene corresponding to i tends to be downstream of the loci defining A, but not the loci defining A'' (genes not belonging to our group of interest).

From these two statistics, we define "upstreamness", which will be positive for regulators, negative for targets, and close to zero for less well-defined genes.

$$\text{upstreamness}_i = D_j^u - D_i^d \tag{4.3}$$

In this case, the subscript j corresponds to the locus containing the gene i.

If we have a gene that has substantial values for both D^u and D^d, upstreamness will be close to zero. Nevertheless, we would like to capture this as an interesting gene. We define centrality to be the sum of D^u and D^d, with the absolute value of upstreamness subtracted as a penalty.

$$\text{centrality}_i = D_j^u + D_i^d - |\text{upstreamness}_i| \tag{4.4}$$

Again, the subscript j corresponds to the locus containing the gene i.

In practice, we use the `ks.test` function in R to acquire D^u and D^d, which are simply the D-statistics from the corresponding KS test. Upstreamness and centrality are then quite straightforward to compute. All of this is wrapped in the function `ucScores`, which takes as arguments `eqtl` (the named eQTL matrix), `genes` (a named logical vector indicating which genes are in the group of interest), `markers` (a named logical vector indicating which loci correspond to the genes of interest), and `cis.map` (a character vector with gene names as names and marker names as entries in the vector, indicating the mapping from genes to markers). The `ucScores` function has a logical switch, `nulldist`, that, when true, calculates upstreamness and centrality scores under the null hypothesis (genes not functionally related, but rather sampled randomly from the data).

We plot the upstreamness and centrality values for the schizophrenia genes, and then overlay the density of values under the null hypothesis (estimated by 10 runs of `ucScore`, each time randomly selecting 334 genes), as shown in Figure 4.2. Values that lie far outside of this density are unlikely to occur by chance, and so represent genes with significant positions in the regulatory hierarchy of schizophrenia-associated genes.

4.2.4 Schizophrenia association studies

We used data from (Allen et al., 2008) to determine the reproducibility of the genes that showed significant ($P < 0.05$) positioning in the regulatory hierarchy. Reproducibility was calculated as the number of populations that showed a positive association of the gene with schizophrenia, divided by the total number of populations where the gene was tested for association. Eight genes with positioning $P < 0.05$ had been tested in association studies. We regressed the genes' upstreamness values on the reproducibility using a simple linear regression, and then extracted the R^2 and associated P value.

4.3 Results

4.3.1 Schizophrenia genes have significantly defined regulatory roles

We assessed regulatory roles in the transcriptional hierarchy of 334 schizophrenia genes by first computing their upstreamness and centrality, then determining the significance of their position by using an empirical 2-dimensional null distribution (Fig. 4.2). 14 of these genes showed a significant ($P < 0.05$) position in the regulatory hierarchy (Fig. 4.1). The positioning of each gene within this 2D map gives an indication as to whether it tends to be an upstream regulator relative to other genes in the map, whether it is central relative to the other genes, or whether it is a downstream transcriptional target of other genes in the map. If the gene's role does not involve many of the other genes, its position on the map will fall well within the null distribution.

MGI symbol	description	upstreamness	centrality	P
Numbl	numb-like	0.104	0.002	1.61×10^{-8}
Olig2	oligodendrocyte transcription factor 2	-0.024	0.184	2.04×10^{-4}
Gls	glutaminase	-0.114	0.003	9.71×10^{-4}
Spnb4	spectrin beta 4	0.075	0.059	1.16×10^{-3}
Cacna1c	calcium channel, voltage-dependent, L type, alpha 1C subunit	0.034	0.120	5.40×10^{-3}
Gabrb3	gamma-aminobutyric acid (GABA) A receptor, subunit beta 3	0.086	0.023	5.67×10^{-3}
Zdhhc8	zinc finger, DHHC domain containing 8	-0.112	0.011	6.54×10^{-3}
Rnh1	ribonuclease/angiogenin inhibitor 1	-0.108	0.048	1.17×10^{-2}
Cck	cholecystokinin	-0.102	0.004	1.66×10^{-2}
Hivep2	human immunodeficiency virus type I enhancer binding protein 2	0.050	0.092	1.80×10^{-2}
Apba2	amyloid beta (A4) precursor protein-binding, family A, member 2	0.028	0.122	1.87×10^{-2}
Arrb2	arrestin, beta 2	-0.101	0.039	1.98×10^{-2}
Rtn4r	reticulon 4 receptor	-0.093	0.011	4.43×10^{-2}
Gclc	glutamate-cysteine ligase, catalytic subunit	0.053	0.065	4.68×10^{-2}

Table 4.1 Upstreamness and centrality values for schizophrenia risk genes with a significant regulatory context.

4.3.2 Significant relationship between reproducibility and upstreamness

Of the 14 genes with a significant position on the regulatory map, eight of them had been previously investigated in genetic association studies for schizophrenia. We assessed reproducibility of the association for each gene by dividing the number of successful associations by the total number of populations investigated. We found a significant positive relationship between the upstreamness of significantly positioned genes and their reproducibility in association studies (Fig. 4.3).

4.4 Discussion

Perhaps the most popular approach to high-level analysis of eQTL data is the construction of an explicit network (Bing and Hoeschele, 2005; Keurentjes et al., 2007; Liu et al., 2008; Suthram et al., 2008; Ghazalpour et al., 2006). Network approaches work best when there is little noise in the data, so that the distinction between real and spurious network edges is clear when the requisite thresholding is performed. However, if there is an appreciable amount of noise in the data, or if the signal is weak, much information about the regulatory roles of genes may be lost after a threshold is applied. In this work, we propose a novel network-free approach for inferring important regulatory roles of genes within a specific functional context – here schizophrenia risk genes. The method utilizes the dense eQTL data without applying a threshold, relying instead on distributional enrichment of eQTL scores within the functional group

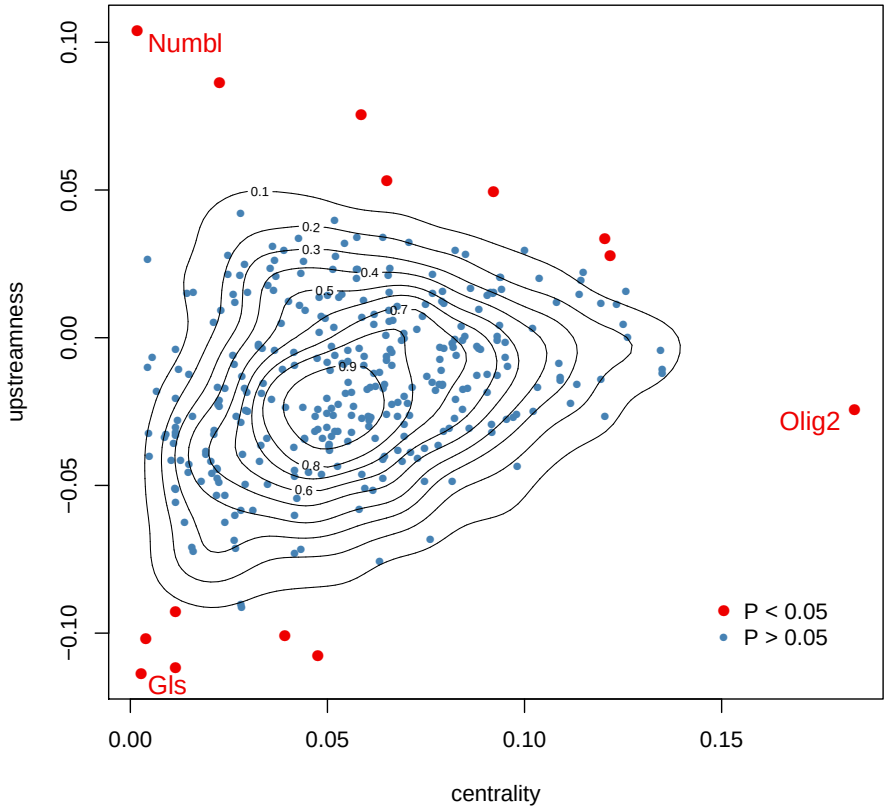


Figure 4.2 Transcriptional regulatory hierarchy for schizophrenia risk genes. Values of upstreamness and centrality expected under the null hypothesis are approximated with contour lines. Examples of highly significant regulatory positioning are shown: *Numbl* as a regulator (high upstreamness), *Olig2* as a router of regulatory information (high centrality), and *Gls* as a target (negative upstreamness).

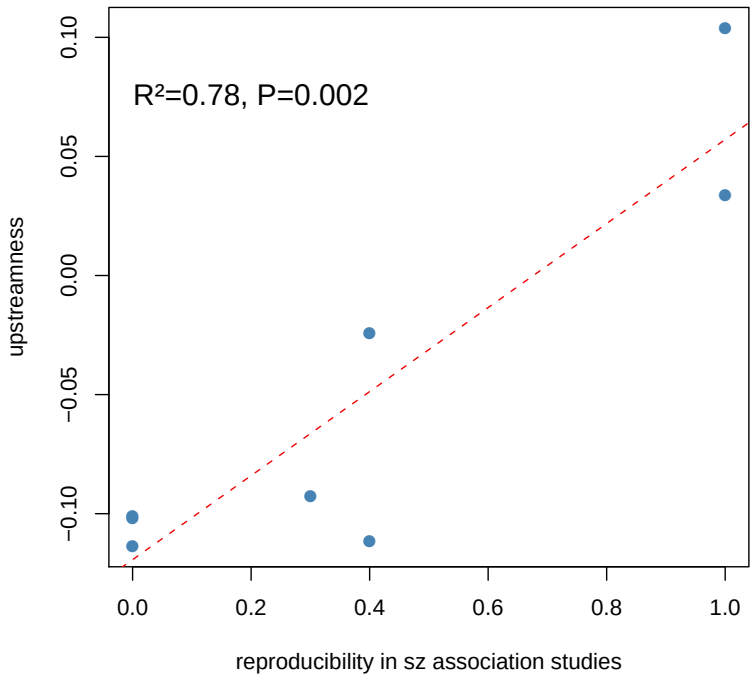


Figure 4.3 Eight of the schizophrenia genes with significant positioning within the regulatory hierarchy have been previously investigated for links to schizophrenia in association studies. We found a significant relationship between upstreamness (in mouse) and reproducibility of association (in human), suggesting that mutations in transcriptional regulators result in higher penetrance of schizophrenia.

to give information about the gene's position within a regulatory hierarchy. The method yields an intuitive 2-dimensional map (Fig. 4.2), where the coordinates of a gene on the map give information about the gene's probable placement within a regulatory hierarchy. While we have found the method to be more sensitive than network approaches in some settings (see for example the motivating simulation in Appendix E), it should be noted that one drawback of the method is that it gives no information about *specific* connections between regulators and their targets, and focuses instead on the probable role of each gene within a contextual group of genes. Therefore, the specific goals of an analysis need to be considered carefully to choose the appropriate method.

As an application of this method, we sought to pinpoint transcriptional regulatory roles of genes associated with schizophrenia – a well-studied psychiatric disorder whose molecular etiology is still unclear. Of the 334 schizophrenia genes investigated, 14 of them had a significant ($P < 0.05$) role in the regulatory hierarchy. Caution must be exercised in interpretation of specific genes, because this is around the number expected under the null hypothesis (i.e. with no correction for multiple testing). Nevertheless, the top 2 genes, *Numbl* and *Olig2* had $FDR < 0.05$, and the top 4 genes were significant at $FDR < 0.1$. We found that *Numbl* had the highest upstreamness of all schizophrenia genes, suggesting a crucial role in the transcriptional regulation of other schizophrenia genes. Polymorphisms in *Numbl* have been found to be significantly associated with schizophrenia in Danish and Brazilian cohorts (Passos Gregorio et al., 2006). Among other roles, *Numbl* is a regulator of Notch signaling, a pathway that influences cell fate and differentiation (Yoon and Gaiano, 2005), with particular importance in the maintenance of neural progenitors. Notch signaling also has a role in determining whether oligodendrocytes – cells responsible for myelination of neurons – continue to proliferate or exit the cell cycle. Recent research has drawn specific connections between notch signaling, regulation of oligodendrocytes, and the etiology of schizophrenia (Kerns et al., 2010). The connection of *Numbl* to the control of the oligodendrocyte population is especially interesting in light of the fact that *Olig2* – oligodendrocyte transcription factor 2 – was found to have a significantly central role in the schizophrenia transcriptional hierarchy. Like Notch signaling, *Olig2* also has a regulatory role in controlling the fate of oligodendrocytes (Jakovcevski and Zecevic, 2005), and has been shown to directly affect myelination phenotypes (Hwang et al., 2009). It and other oligodendrocyte-related genes have been implicated in schizophrenia (Georgieva et al., 2006), though the results of association studies have been mixed (Allen et al., 2008). Taken together, these genes with significant roles in the transcriptional regulatory hierarchy suggest an important role for the proper maintenance of the pool of myelinating oligodendrocytes in both the etiology of schizophrenia and the proper transcriptional regulation of schizophrenia risk genes.

In addition to the specific conclusions about the regulatory importance of genes relating to myelination and control of the pool of oligodendrocytes, we were able to find a suggestive general relationship between upstreamness of significant genes (in mouse) and the reproducibility of their human orthologs in schizophrenia association studies (Fig. 4.3). This supports the notion that is increasingly appearing in the literature (Nicolae et al., 2010) that eQTL underlie disease loci. This implies that it may be the disruption of transcriptional regulatory programs, rather than a direct functional mutation, that underlies many genetic diseases. Our data suggest two things: first, that this particular transcriptional regulatory hierarchy is conserved between mouse and human. This may not be true in an exact sense, but at least the general priority of these genes is preserved. Secondly, the relationship suggests – intuitively – that mutations of risk genes that are also transcriptional regulators are likely to lead to higher penetrance of schizophrenia, compared to mutations in genes that exist as transcriptional targets in the regulatory hier-

archy of schizophrenia genes. Naturally, since this relationship is based on only eight data points, caution must be exercised, and this relationship must be replicated before these suggestions can be taken as conclusive and generalizable. It is also important to emphasize that the polymorphisms in mouse causing eQTL are not directly comparable to the polymorphisms in human that lead to schizophrenia; the mouse polymorphisms can only inform us about the relationship between genes. The correlation observed in Figure 4.3 is further complicated by the fact that we can only see effects in the mouse data for genes that are actually polymorphic in the population. We may be missing the roles of some genes because of insufficient genetic diversity in the mouse panel.

4.5 Author contributions and acknowledgements

I conceived of and implemented the idea of upstreamness and centrality. I performed the analysis and wrote the manuscript. Andreas Beyer contributed to the refinement of the idea and contributed in an editorial fashion to the manuscript.

Chapter 5

Direct and indirect transcriptional targets

The following publications and presentations relate to the work presented in this chapter:

1. **Michaelson, J. J.**, Trump, S., Rudzok, S., Gräbsch, C., Madureira, D., Dautel, F., Schirmer, K., von Bergen, M., Lehmann, I., and Beyer, A. Transcriptional signatures of regulatory and toxic responses to chemical exposure. (submitted)

2. Dautel, F., Kalkhof, S., Trump, S., **Michaelson, J.J.**, Beyer, A., Lehmann, I., and von Bergen, M. DIGE-based protein expression analysis of B[a]P-exposed hepatoma cells reveals a complex stress response at toxic and subacute concentrations. *J. Proteome Res.* 2010.

3. **Michaelson, J. J.**, Trump, S., Madureira, D., Dautel, F., von Bergen, M., Schirmer, K., Lehmann, I., and Beyer, A. The *Ahr* transcriptional cascade. Helmholtz Alliance on Systems Biology – Status Meeting 2009, Heidelberg, Germany.

5.1 Introduction

The aryl hydrocarbon receptor (*Ahr*) is a ligand-dependent transcription factor that is activated by a wide range of xenobiotic and endogenous compounds. The *Ahr* resides in the cytoplasm in a chaperone complex together with *Xap2* (*Aip*, *Ara9*) and *Hsp90*. After ligand binding, the receptor translocates to the nucleus where it associates with its cofactor *Arnt* (*Ahr* nuclear translocator) yielding a competent transcription factor. This heterodimer binds to so-called xenobiotic response elements (XREs) that function as cis-acting enhancers in the regulatory domain of a wide range of genes commonly referred to as the *Ahr* gene battery (Nebert et al., 1993; Nebert et al., 2004). Given the role of *Ahr* in mediating the transcriptional response to environmental pollutants, it is not surprising that the best defined subset of genes activated by *Ahr* includes genes involved in Phase I/II metabolism like the cytochrome P450 enzyme *Cyp1a1*, NAD(P)H:quinine oxidoreductase (*Nqo1*) or aldehyde dehydrogenase (*Aldh3a1*). Activation of metabolizing enzymes through *Ahr* may lead to the formation of toxic metabolites of the activating ligand itself. This is particularly true for benzo(a)pyrene (B[a]P), a classical *Ahr* agonist. Only after the self-induced metabolism of this procarcinogen is the ultimate genotoxic metabolite anti-benzo(a)pyrene-trans-7, 8-dihydroxy-9,10-epoxid (BPDE) formed.

DNA microarray technology offers an appealing approach to investigate transcriptional changes on a genome-wide scale. Several studies have already been undertaken to examine the effects of *Ahr* activa-

tion in different species and cell types (Kim et al., 2009a; Carlson et al., 2009; Gohlke et al., 2009; Carney et al., 2006). However, deciphering the *Ahr*-specific transcriptional response is not a trivial task considering that *Ahr* activation might trigger the activation of other transcription factors or the generation of toxic metabolites which will add secondary effects to the observed differential gene expression. Therefore, the overall transcriptional response directly related to *Ahr* binding is incompletely elucidated, and the number of well-defined *Ahr* specific genes still remains small.

One strategy to assess *Ahr*-dependence is to compare gene expression of cells or tissues that have the wild type *Ahr* with those of *Ahr*-null cells in a matched genetic background as was shown by Tijet et al. (Tijet et al., 2006). In their study they compared the effect of TCDD in *Ahr* +/+ and *Ahr* -/- mice after long term exposure. This experimental setup, as the authors themselves conceded, bears the problem that although an involvement of the *Ahr* might be necessary for the observed differential gene expression, the influence of secondary effects increases over time.

In an elegant experimental setup, Hockley et al. (Hockley et al., 2006; Hockley et al., 2007) sought to separate the direct effects of *Ahr* activation from the secondary effect caused by the genotoxic metabolite BPDE. They compared the effects of B[a]P, BPDE and TCDD exposure in two different human cell lines. Unfortunately, the first time point they investigated was not until six hours after exposure. Considering that it was shown previously that *Ahr* translocation and nascent transcription is already induced 1h after TCDD exposure (Elbi et al., 2002), we believe that identification of direct *Ahr* targets is only possible by including early time points of exposure in gene expression studies to identify direct *Ahr* targets.

A machine learning approach offers an alternative to relying on basic comparisons in a sophisticated experimental design. For example, it is probable that the direct targets of *Ahr* share expression characteristics distinct from genes that are part of a secondary response. This difference could be learned from time-resolved expression data, given an appropriate training set, and the learned patterns could be used to predict the roles of responding genes. This general strategy of using time course gene expression data to predict transcriptional regulatory roles has been previously explored (Segal et al., 2003; Bansal et al., 2006; Redestig et al., 2007; Ruan et al., 2009), particularly in lower organisms such as bacteria and yeast.

We expect that because such learning methods are less encumbered by methodological assumptions (compared to traditional statistical comparisons), they are more able to find subtle but meaningful patterns in the data. For example, an important assumption of previous attempts to cluster *Ahr*-centric expression data (Dere et al., 2006; Frericks et al., 2008; Kim et al., 2009a; Boutros et al., 2009) is that co-regulated genes should also be co-expressed. Hence, clustering of genes based on expression patterns should identify sets of genes subject to the same regulatory program. However, in time courses such co-expression may only be present during certain phases. In the case of *Ahr* we expect co-expression during early time points, whereas expression may diverge later when the influence of *Ahr* diminishes. The analysis presented here anticipates and effectively deals with this scenario.

Here we employ machine learning techniques coupled to a straightforward yet robust experimental design in order to more clearly define genes that are under the direct transcriptional control of *Ahr*. This is accomplished by training a Random Forest (Breiman, 2001) (RF) classifier to learn the difference between genes responding to B[a]P exposure and secondary effects caused by the B[a]P metabolite BPDE. The trained classifier is then applied to all genes found to be significantly differentially expressed as a result of B[a]P exposure, and their roles as direct targets or secondary effects are predicted. In addition, the patterns learned by the classifier are used as a basis for performing weighted clustering. These clusters facilitate a better understanding of the functional relatedness of the perturbed genes. Finally, we support

predictions with our own experimental follow-up, as well as with data from independent studies.

Definition of open problem

Given an extensive transcriptional response upon induction of a transcription factor, how can direct targets be distinguished from indirect effects? How can the responding genes be clustered in functional groups in a way that accounts for individual transcript differences in synthesis and degradation?

5.2 Materials and Methods

5.2.1 Cell culture and sample preparation

Murine hepatoma cells, Hepa1c1c7 as well as the mutant tao BpRc1 cells (both LG Standards GmbH, Wesel, Germany), deficient in endogenous *Ahr*, were used for all experiments. Cells were cultured in phenol red-free DMEM supplemented with 7% FCS, 1% glutamine and 1% penicillin/streptomycin. Cells were stimulated with different concentrations of benzo(a)pyrene (B[a]P; Sigma Aldrich, Steinheim, Germany), BPDE (Midwest Research Institute, NCI Chemical Repository, Kansas City, MO, USA) and TCDD (Sigma-Aldrich, Steinheim, Germany) respectively.

5.2.2 Microarrays

To investigate the differential kinetic behavior of the transcriptome after B[a]P exposure, and to identify the *Ahr*-specific response, we used two different setups: (1) short term exposure, Hepa1c1c7 cells were treated with 50nM B[a]P for 0, 1, 2, and 4 hours and (2) long term exposure, Hepa1c1c7 cells were treated with 50nM or 5uM B[a]P for 2, 4, 12 and 24h. Time-matched vehicle controls were generated with a DMSO concentration of 0.05%. All experiments were performed in triplicates. Cells were lysed in Trizol reagent (Invitrogen, Darmstadt, Germany) and RNA extracted using RNAeasy kits (Qiagen, Valencia, CA, USA). RNA was quantified and integrity verified on a Bioanalyzer (Agilent Technologies, Palo Alto, CA). Sample preparation for Affymetrix GeneChip Mouse Exon 1.0 ST arrays (Affymetrix, Santa Clara, CA, USA) was performed following the manufacturer's recommendations.

5.2.3 Detection of differential expression

Microarrays were normalized using RMA and the University of Michigan custom CDF file (version 12.1.0) with mappings to Ensembl exon IDs. After normalization, but before proceeding with the analysis, we subtracted the ($\log_2$) DMSO expression values from the corresponding time point and batch of each of the B[a]P treatments. Exon expression values were then summarized to their corresponding Ensembl gene IDs, with the summarized gene expression value being the mean of its constituent exons. A 2-way ANOVA analysis was performed on each gene, with time and concentration as the factors. We then corrected for multiple testing by using the FDR. We considered only genes with an $FDR < 0.05$ for any of the main effects or time*concentration interaction. In addition, we admitted genes with an $FDR < 0.05$ from a simple t-test each B[a]P concentration (all time points pooled) vs. DMSO. Of these, we only considered genes that achieved 2-fold (or greater) differential expression at at least one time point. This left us with a total of 2,338 genes. We interpolated the expression between the measured time points by averaging the

simple linear interpolation with the spline interpolation. Since we have no measurement at time 0 hours, we assume equivalent expression with the DMSO samples, i.e. the expression ratio at time 0 hours is 0 on the $\log_2$ scale. The interpolation gave us a total of 25 values per gene, 1 value every hour from 0 to 24 hours.

5.2.4 Classification with Random Forests

We used the R implementation of Random Forests (Liaw and Wiener, 2002) to perform the two-class classification (*Ahr* direct vs. indirect regulation), using the time course expression measurements of significantly regulated genes as predictors. To derive training labels (Fig. 5.2), we used data available from two BPDE studies in human cell lines (Lu et al., 2009; Lu et al., 2010), combining the *P* values from the studies using Fisher's method. We labeled mouse orthologs of genes with BPDE-perturbed expression ($FDR < 0.05$) as "*Ahr*-indirect" since BPDE does not bind *Ahr*, but indicates affected genes further downstream of *Ahr*. We labeled genes as "*Ahr*-dependent" that showed differential expression ($FDR < 0.05$) in an independent gene expression time course of cells exposed to 50nM B[a]P from 0 to 4 hours, with the additional condition that they were not significantly regulated in the BPDE data (i.e. orthologs had $FDR > 0.05$). These criteria led to 28 "*Ahr*-direct" labeled genes and 559 "*Ahr*-indirect" labeled genes.

With this training set we ran RF with `mtry` set to 5, and `ntree` set to 5,000. We used the built-in outlier measure and removed genes in the 95^{th} percentile of outlier scores (resulting in 27 direct and 530 indirect training cases), then re-ran RF, this time with 1,000 trees. In both cases, to avoid biased predictions (since there are far more "*Ahr*-indirect" samples) we randomly sampled 20 genes from each class for the construction of each tree in the forest. The overall misclassification rate for the final forest was 7% (out of bag error estimate).

Predictions were made for all 2,338 differentially expressed genes, and genes with a proportion of class votes greater than 80% were retained for further analysis. This cutoff was chosen because when the training labels were permuted randomly and a RF trained, no prediction had a proportion of votes greater than 80%. Using these criteria, a total of 82 genes were predicted to be responding to *Ahr* directly, and 1,365 genes were predicted to be indirectly regulated by *Ahr* (e.g. through the presence of B[a]P metabolites). In addition to predictions, the RF proximity measure was calculated for all significant and confidently classified genes, yielding a 1,447 by 1,447 matrix.

5.2.5 Clustering

The RF proximity matrix was used as a distance measure by the transformation $D = \sqrt{1 - P}$, where P is the original proximity matrix and D is the distance matrix. This distance matrix was then used as the input for PAM clustering, available in the R `cluster` package. We tested a range of k values and found that specifying 3 clusters gave the best average silhouette.

To assess the degree of confidence in cluster assignment for each gene, an RF was fit to predict cluster label using the gene expression measurements. The proportion of votes for the correct cluster is an indication of how well a gene fits in the cluster. Genes that were given a lower proportion of votes for the correct class than expected under the null hypothesis (labels permuted randomly) were excluded. When including this additional filtering criterion, the final number of genes classified as direct targets was 81, with 1,308 genes as secondary effects. In addition, the importance measurements obtained in the

construction of this RF give an indication of which time points and which concentrations are important parts of the cluster's identity.

GO enrichment was performed for each cluster using the `topGO` package (Alexa et al., 2006). Enrichment of the clusters for genes perturbed by an *Ahr* mutation was performed using the Kolmogorov-Smirnov test, using *P* values derived from differential expression of genes from (Tijet et al., 2006; Sartor et al., 2009). *P* values were calculated for each study separately, then combined using Fisher's method. Genes used to train the RF classifier were removed prior to calculation of enrichment, to ensure that the results reflected the actual predictive ability of the classifier.

5.2.6 qPCR

In a separate experiment Hepa1c1c7 and tao BpRc1 cells were exposed to B[a]P (50, 5uM), BPDE (50nM, 5uM) and 1nM TCDD for 0.5, 1, 2, and 4h. mRNA was extracted and isolated using the MagNA Pure LC System (Roche Diagnostics GmbH, Mannheim, Germany). 50ng of mRNA was reverse transcribed according to the protocol provided with the AMV reverse transcriptase (Promega, Madison, WI, USA). Resulting cDNA was diluted 1:5 and 4ul of template used in a 12ul PCR reaction. qPCRs were performed for the following example genes: *Cyp1a1*, *Tnfaip2*, *Tiparp*, *Cdkn1b*, *Mpp2*, *Nfe2l2*, *Nfkb1*, *Agfg1*, *Blvrb*, *Cox7a1*, *Cdc6*, *Parp1*, 18S rRNA, and *Gapdh*. All qPCR experiments were carried out on a LightCycler®480 system (Roche Diagnostics GmbH, Mannheim, Germany) with the following settings: touchdown amplification with an initial step of 960 C for 10 min; followed by the first cycle at 950 C for 10 sec. The annealing step started at 680 C for 20 sec (decrease of 0.50 C/cycle with a step delay of 1 cycle) and reaching the annealing temperature of 580 C for the last 25 cycles, followed by 720 C for 20 sec for extension. A total of 45 cycles were performed in each experiment.

5.2.7 ChIP

Hepa1c1c7 cells were grown in 15 cm dishes. 20,000 cells/cm^2 were seeded and treatment started 48h thereafter. Cells were exposed to 50nM B[a]P or DMSO as the vehicle control for 1h respectively. Subsequently cells were cross-linked for 10 min at 37 C in 1% formaldehyde followed by a quenching step for 10 min with 150 mM glycine. After cross-linking, chromatin DNA was sheared into 200-500 bp fragments by sonication using a Bioruptor®Next Gen (UCD-300, Diagenode SA, Liege, Belgium). Sonicated, soluble chromatin was immune-precipitated with 2.5 ug of an anti-*Ahr* antibody (Enzolifesciences/Biomol, Lörrach, Germnay) or anti-Pol II (Millipore, Billerica, MA, USA). Control IPs were performed using rabbit IgG (Millipore, Billerica, MA, USA) corresponding to our specific antibodies. All ChIP experiments were performed at least two times.

study	organism	ligands	timepoints	# DE genes
Hockley, et al. (Hockley et al., 2007)	H. sapiens	TCDD, B[a]P, BPDE	6h, 24h	1,207
Hockley, et al.(Hockley et al., 2006)	H. sapiens	B[a]P, B[e]P	6h, 24h, 48h	202
Kim, et al. (Kim et al., 2009a)	H. sapiens	TCDD	1h, 2h, 4h, 8h, 12h, 24h, 48h	144
Dere, et al. (Dere et al., 2006)	M. musculus	TCDD	1h, 2h, 4h, 8h, 12h, 24h, 48h	285
Frericks, et al. (Frericks et al., 2008)	M. musculus	TCDD	2h, 4h, 6h	201
Michaelson & Trump, et al.	M. musculus	B[a]P	2h, 4h, 12h, 24h	2,338

Table 5.1 Overview of *Ahr*-centric time-resolved microarray studies. A brief description of the experimental factors is given, along with the total number of differentially expressed genes resulting from exposure. If multiple cell lines were tested, the maximum number of differentially expressed genes is reported.

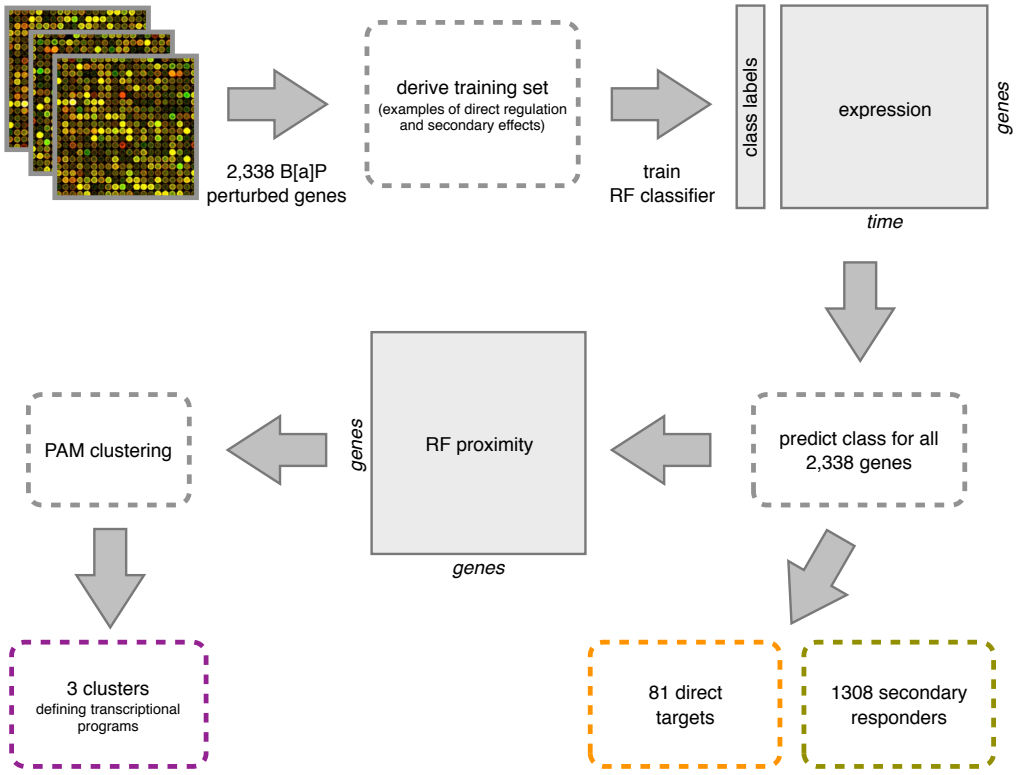


Figure 5.1 Framework for predicting *Ahr* direct targets and secondary effects using gene expression time course data.

5.3 Results

5.3.1 Extensive transcriptional response

A total of 2,338 genes were perturbed significantly ($FDR < 0.05$) by exposure to B[a]P and had at least a 2-fold change (with respect to DMSO-exposed cells) at some point over the course of the experiment. Compared to previous studies of *Ahr*-mediated temporal gene expression, this represents a very substantial transcriptional response (see Table 5.1). These genes were highly enriched for a host of biological processes (summarized in Table 5.2), including mRNA transport, control of the cell cycle, apoptosis, and development.

5.3.2 Prediction of direct vs. indirect targets of *Ahr*

The overall analytical framework used here is summarized in Figure 5.1. Using a matrix of time-resolved gene expression values as predictors (interpolated as described in methods), we trained a Random Forest classifier in a two-class scenario (Ahr direct and indirect effect). Training labels were assigned based on the significant perturbation of a gene in conditions that suggest being either a direct *Ahr* target or responsive to the presence of BPDE (secondary or indirect effect). This yielded 28 genes as direct examples and 559 genes as indirect examples (Fig. 5.2), before filtering for outliers. The final classifier had an estimated misclassification rate of 7%. Performance of the classifier on out-of-bag (OOB) data is depicted as a ROC curve in Figure 5.3, panel A.

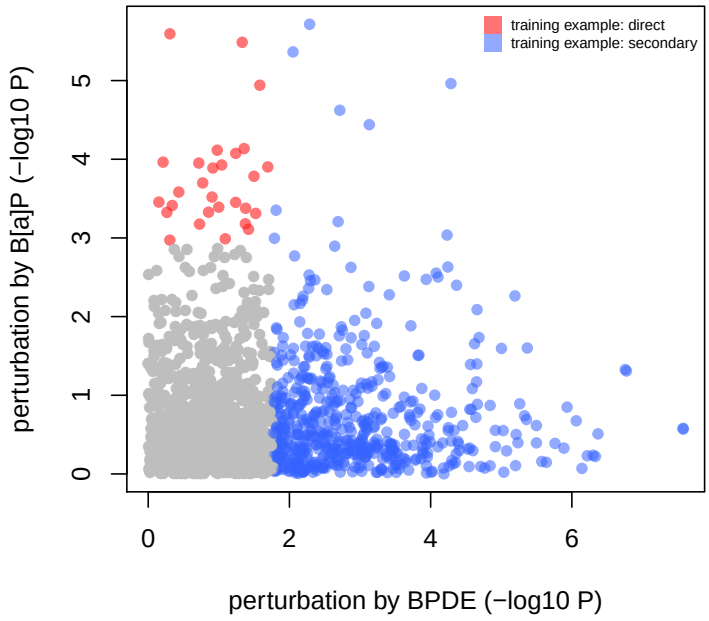


Figure 5.2 Defining the training set. We used perturbation by BPDE ($FDR < 0.05$) as an evidence of a secondary effect, since BPDE does not activate *Ahr* but at the same time is a metabolite of B[a]P. Accordingly, perturbation by B[a]P ($FDR < 0.05$) but not by BPDE ($FDR > 0.05$) was taken as evidence for direct regulation by *Ahr*. A total of 1,663 genes of the 2,338 differentially expressed genes were examined as potential training examples, and 587 genes were then assigned to the training set.

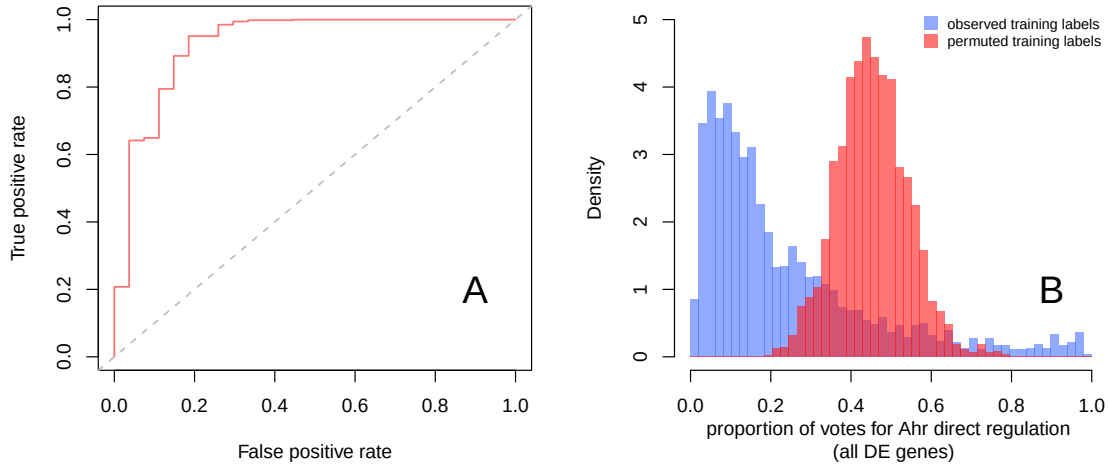


Figure 5.3 Performance and confidence of predictions of the Random Forest classifier. Performance of the classifier predicting out-of-bag (OOB) data, depicted as a ROC curve (A). Confidence in class predictions can be expressed as the proportion of votes cast for the class (B). Here we show the actual proportions of votes of all differentially expressed (DE) genes (blue), and the proportions when the training labels are permuted (red). The classifier's predictions are more reliable for genes that have a proportion of votes outside of the overlapping region. Note that since this is a two-class scenario, a proportion close to 0 in this figure corresponds to a high proportion of votes for the gene being perturbed as a secondary effect.

ID	Term	Annotated	Significant	Expected	P
GO:0051028	mRNA transport	54	18	5.99	1.1e-05
GO:0051301	cell division	264	57	29.30	2.6e-05
GO:0001701	in utero embryonic development	213	44	23.64	3.3e-05
GO:0007050	cell cycle arrest	54	17	5.99	4.6e-05
GO:0043065	positive regulation of apoptosis	265	57	29.41	4.6e-05
GO:0008285	negative regulation of cell proliferatio...	212	43	23.53	6.3e-05
GO:0000059	protein import into nucleus, docking	15	8	1.66	7.1e-05
GO:0032318	regulation of Ras GTPase activity	78	21	8.66	8.3e-05
GO:0045944	positive regulation of transcription fro...	345	62	38.29	8.4e-05
GO:0006468	protein amino acid phosphorylation	802	131	89.01	9.6e-05
GO:0007169	transmembrane receptor protein tyrosine ...	212	48	23.53	0.00011
GO:0051130	positive regulation of cellular componen...	111	26	12.32	0.00016
GO:0051726	regulation of cell cycle	269	50	29.86	0.00017
GO:0043066	negative regulation of apoptosis	241	52	26.75	0.00017
GO:0006511	ubiquitin-dependent protein catabolic pr...	156	33	17.31	0.00019
GO:0001525	angiogenesis	159	33	17.65	0.00027
GO:0007067	mitosis	205	48	22.75	0.00028
GO:0016477	cell migration	298	52	33.07	0.00062
GO:0046777	protein amino acid autophosphorylation	67	17	7.44	0.00081
GO:0015813	L-glutamate transport	20	8	2.22	0.00083
GO:0006309	DNA fragmentation involved in apoptosis	12	6	1.33	0.00095
GO:0008286	insulin receptor signaling pathway	40	12	4.44	0.00096
GO:0006915	apoptosis	812	156	90.12	0.00099

Table 5.2 Enrichement of GO biological processes among 2,338 DE genes (with an enrichment *P* value of less than 0.001).

We then used this trained classifier to predict on all of the 2,338 differentially expressed genes. The predictions have varying degrees of confidence, indicated by the proportion of votes cast for the predicted class. To establish a threshold above which we could be confident that the classifier was predictive, we permuted the original training labels randomly, trained a Random Forest with these labels, and predicted on all 2,338 genes. In general we found that in this "null" scenario, the Random Forest did not predict with a proportion of votes greater than 0.8. Therefore, we consider a class prediction with a proportion of votes greater than 0.8 to be a reliable prediction (Figure 5.3, panel B). After filtering, 81 genes were predicted as direct targets of *Ahr* (Table 5.3), 1,308 genes were predicted as secondary responders, and 949 genes could not be reliably classified.

5.3.3 Characterization of transcriptional response programs

To characterize the expression patterns that underlie the classifier's decision rules, we used the RF proximity measure as an input to PAM clustering. This yielded three coherent clusters, depicted in Figure 5.4. Clusters 1 and 2 are comprised of genes predicted to be secondary effects of *Ahr* activation by B[a]P, while cluster 3 contains genes predicted to be direct targets of *Ahr*. Clusters 1 and 2 are characterized by undulating expression profiles in the low (50 nM) B[a]P exposure, with the mean behavior of each cluster strongly anticorrelated to the other. The high (5 μM) B[a]P exposure shows less cohesive expression patterns, but with the same general trend of anitcorrelation between clusters 1 and 2. In both cases, time points in the 50 nM B[a]P series are more important for the identity of the clusters than time points in the 5 μM B[a]P series. Cluster 3 is characterized by punctuated expression induction at 3 hours in the 50 nM B[a]P time series, and a slightly extended phase of induction in the 5 μM B[a]P time series. Other time points are unimportant for the cluster's identity; indeed, the expression of these genes is fairly divergent outside of the common phase of induction. Although cluster 3's "identity phase" is generally between 3-4 hours after exposure, where all genes in the cluster show elevated expression, several genes (such as

MGI ID	cluster votes	target votes	train/test	MGI ID	cluster votes	target votes	train/test
Hspa4l	1.00	0.98	test	Ccng2	0.98	0.9	test
2410066E13Rik	1.00	0.98	test	Fam198b	0.98	0.9	test
Arl6ip5	1.00	0.98	test	Ddit4	0.98	0.9	test
Plscr2	1.00	0.98	test	Ubl3	0.98	0.87	test
Mpp2	1.00	0.98	training	Nqo1	0.98	0.87	test
Tiparp	1.00	0.98	training	Trp53inp1	0.98	0.87	test
Sdpr	1.00	0.98	test	Cyp1a1	0.98	0.86	test
Ndrg1	1.00	0.97	test	Abca6	0.98	0.86	test
Nrn1	1.00	0.97	test	Hmox1	0.98	0.83	test
Cyp2s1	1.00	0.97	test	Aldh4a1	0.98	0.81	test
Tnfaip2	1.00	0.97	test	Npffr1	0.97	0.91	test
Cpox	1.00	0.97	training	**Btg2**	0.97	0.89	test
Osbpl2	1.00	0.97	test	**Nr3c1**	0.97	0.87	test
Rbks	1.00	0.96	test	Gm10122	0.97	0.87	test
Ppard	1.00	0.96	training	Snx30	0.96	0.96	training
Tbc1d16	1.00	0.95	test	Cdkn1b	0.96	0.92	test
Arrdc3	1.00	0.95	training	Slc26a2	0.96	0.88	test
Lpin1	1.00	0.95	test	Plk2	0.96	0.85	test
Id2	1.00	0.94	test	**Zscan29**	0.96	0.83	test
Xdh	1.00	0.94	test	Zfp608	0.95	0.92	training
Gramd3	1.00	0.94	test	Nrg1	0.95	0.91	test
Serpine1	1.00	0.93	test	Abcd2	0.95	0.8	test
Pfkfb3	0.99	0.98	training	**Klf9**	0.94	0.94	test
Jub	0.99	0.97	test	Dusp1	0.94	0.92	training
Ddx58	0.99	0.97	training	Tnfaip8	0.94	0.88	test
Zfp418	0.99	0.95	test	9330175E14Rik	0.94	0.82	test
Sgk1	0.99	0.94	test	Lrrc30	0.93	0.89	test
Jun	0.99	0.93	test	Eda2r	0.93	0.85	test
Cdkn1a	0.99	0.92	test	Bmf	0.92	0.93	test
Abcc4	0.99	0.91	test	Rnf39	0.91	0.92	training
Slc6a9	0.99	0.91	test	St6gal1	0.9	0.94	training
Adh7	0.99	0.90	test	Zfp36l1	0.89	0.83	test
Usp18	0.99	0.90	test	**Nr1d1**	0.86	0.91	training
Npc1	0.99	0.88	test	Irs2	0.86	0.91	test
Casp3	0.99	0.87	test	**Ets2**	0.84	0.86	training
Aldh3a1	0.99	0.86	test	**Nfe2l2**	0.78	0.86	test
Slc35d1	0.99	0.85	test	**Irf1**	0.76	0.91	training
Cyp1b1	0.98	0.97	test	Cib2	0.71	0.84	test
Intu	0.98	0.95	training	S1pr1	0.7	0.89	training
Pitpnc1	0.98	0.95	training	Traf5	0.59	0.89	training
Sesn2	0.98	0.92	training				

Table 5.3 List of predicted direct targets of Ahr transcriptional regulation. Confidence scores of both membership in cluster 3 (cluster votes) and as an Ahr direct target (target votes) are given. Known transcriptional regulators (annotated with GO terms GO:0045449, GO:0006350, GO:0003700, GO:0008134, GO:0003712, or GO:0030528) are bolded. Additionally, assignment to either the training or test set is indicated for each gene.

Cyp1a1 and Tiparp) in the cluster are highly expressed well before this window.

We evaluated the clusters for enrichment of genes perturbed by an *Ahr* mutation (Fig. 5.5). By using data from previous studies (Tijet et al., 2006; Sartor et al., 2009), we performed a 2-way ANOVA and took *P* values from the genotype*ligand interaction; these *P* values were used as indicators of genes under the direct influence of *Ahr*. Only genes belonging to the test set were used when calculating the enrichment. Cluster 3 showed extreme significance, and was the only cluster to show enrichment for genes perturbed by an *Ahr* mutation. This result further supports the assertion that cluster 3 contains true *Ahr* direct targets, and that the classifier is predictive in practice.

5.3.4 Experimental confirmation of *Ahr* dependency

Two independent experimental approaches were chosen to confirm *Ahr*-dependency for a subset of representative genes: direct comparison of the transcriptional response of *Ahr*-expressing Hepa1c1c7 and

Figure 5.4 Clustering with the RF proximity measure. PAM clustering was performed with a supervised, weighted distance measure, derived during the classification of *Ahr* direct targets and secondary effects. Three distinct programs were found, depicted here as clusters. Color saturation indicates the importance of the time points for the identity of the cluster. To further emphasize these important time points, this same information is shown again for each cluster (black to yellow scale). The classification of each gene is shown as the proportion of RF votes.

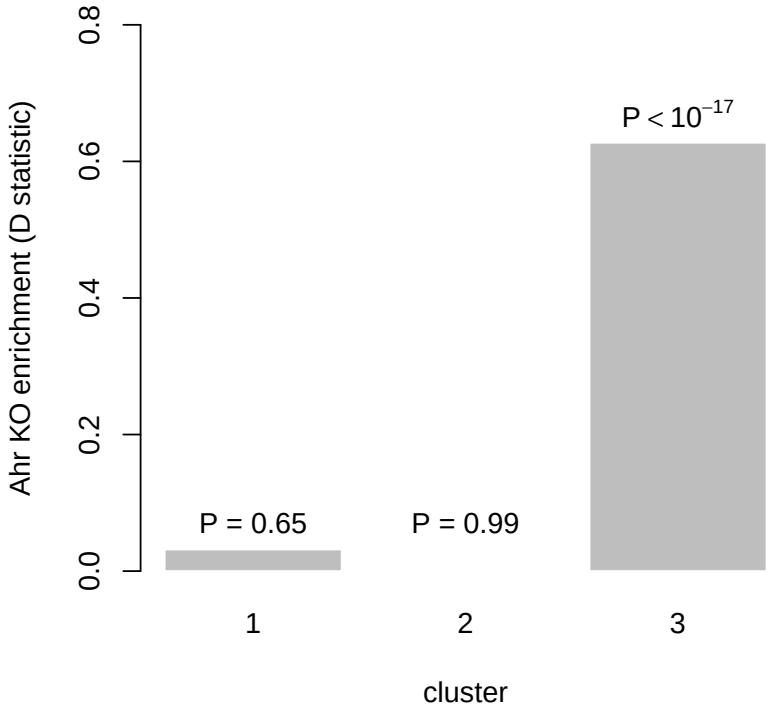


Figure 5.5 Enrichment of each cluster for *Ahr* mutant-perturbed genes. Using data from previous *Ahr* mutant studies (Tijet et al., 2006; Sartor et al., 2009), we assessed whether each cluster was enriched (relative to the other clusters) for genes perturbed by an *Ahr* mutation. Only genes not used in the training of the classifier were used in the calculation of enrichment. Cluster 3 was highly enriched for perturbed genes, suggesting that it is enriched for *Ahr* targets.

mutant tao BpRc1 cells deficient in endogenous *Ahr*, as well as confirmation of binding of *Ahr* in the corresponding promoter regions by ChIP.

Differential expression of *Tiparp*, *Tnfaip2*, *Cdkn1a*, *Cdkn1b*, *Cyp2s1*, *Nfe2l2*, *Mpp2* after treatment with B[a]P or TCDD at different concentrations was investigated by qPCR. After B[a]P and TCDD exposure, the expression of all genes was induced as soon as 1h after the start of treatment in Hepa1c1c7 cells, while there was no significant induction compared to vehicle control samples detectable in tao BpRc1 cells up to four hours after exposure (Fig. 5.6).

Enrichment of *Ahr* binding in the promoter region of all chosen genes could be confirmed by ChIP, with fold changes (compared to vehicle control samples) ranging from 5.9-113.9 (Table 5.4).

5.4 Discussion

Activation of *Ahr* induces a complex transcriptional response. This change in gene expression is a mixture of direct effects, due to *Ahr* binding to gene regulatory sequences, and a secondary response, due in part to stress caused by an active metabolite of the *Ahr* ligand, in our case the B[a]P metabolite BPDE. In addition, other transcription factors might themselves be *Ahr* target genes, leading to regulation of further pathways as a result of *Ahr* activation.

Our approach aimed to predict the direct *Ahr* targets using our time course gene expression data of Hepa1c1c7 cells exposed to different concentrations of B[a]P. From the overall 2,338 genes regulated, 81 were predicted to be direct *Ahr* targets. Among those are 11 transcriptional regulators that are involved in a variety of biological functions.

5.4.1 *Ahr* target genes

Previously well-described members of the *Ahr* gene battery like *Cyp1a1*, *Nqo1 Cyp2s1*, *Aldh3a1*, *Aldh4a1* and *Cyp1b1* (Liu et al., 1994; Nebert et al., 1993; Nebert et al., 2000) were predicted as direct targets of *Ahr*. In addition to this qualitative confirmation of the effectiveness of our computational approach, we could demonstrate *Ahr* dependency experimentally by ChIP and qPCR.

The experimental strategy to confirm our findings is based on comparing the transcriptional response of wild type hepatoma cells (Hepa1c1c7) to the response of corresponding mutants deficient in endogenous *Ahr* (tao BpRc1) by quantitative real time PCR (qPCR). In addition, a representative subset of genes was chosen and *Ahr* binding in corresponding promoter regions was confirmed by ChIP assays (Table 5.4).

As previously mentioned, B[a]P is likely to induce secondary effects independent of *Ahr* activation, therefore we included TCDD – known as an exclusive, non-metabolized *Ahr* ligand – in our confirmation experiments. All of the genes chosen for the qPCR verification showed the expected results (Fig. 5.6) and confirmed our predicted *Ahr*-dependency.

For a functional evaluation of the predicted target genes we performed a GO enrichment analysis. The regulated genes in cluster 3 were enriched for 15 different biological functions including endogenous functions like "lipid transport" or "blood vessel morphology". Moreover, two biological process terms pointed to the influence of *Ahr* on cell cycle control, namely "cell cycle arrest" and "negative regulation of cell proliferation". Experimental confirmation of two of these genes, the cyclin-dependent kinase inhibitors *Cdkn1a* and *Cdkn1b*, showed an exclusive induction in wild type cells by qPCR, together with an en-

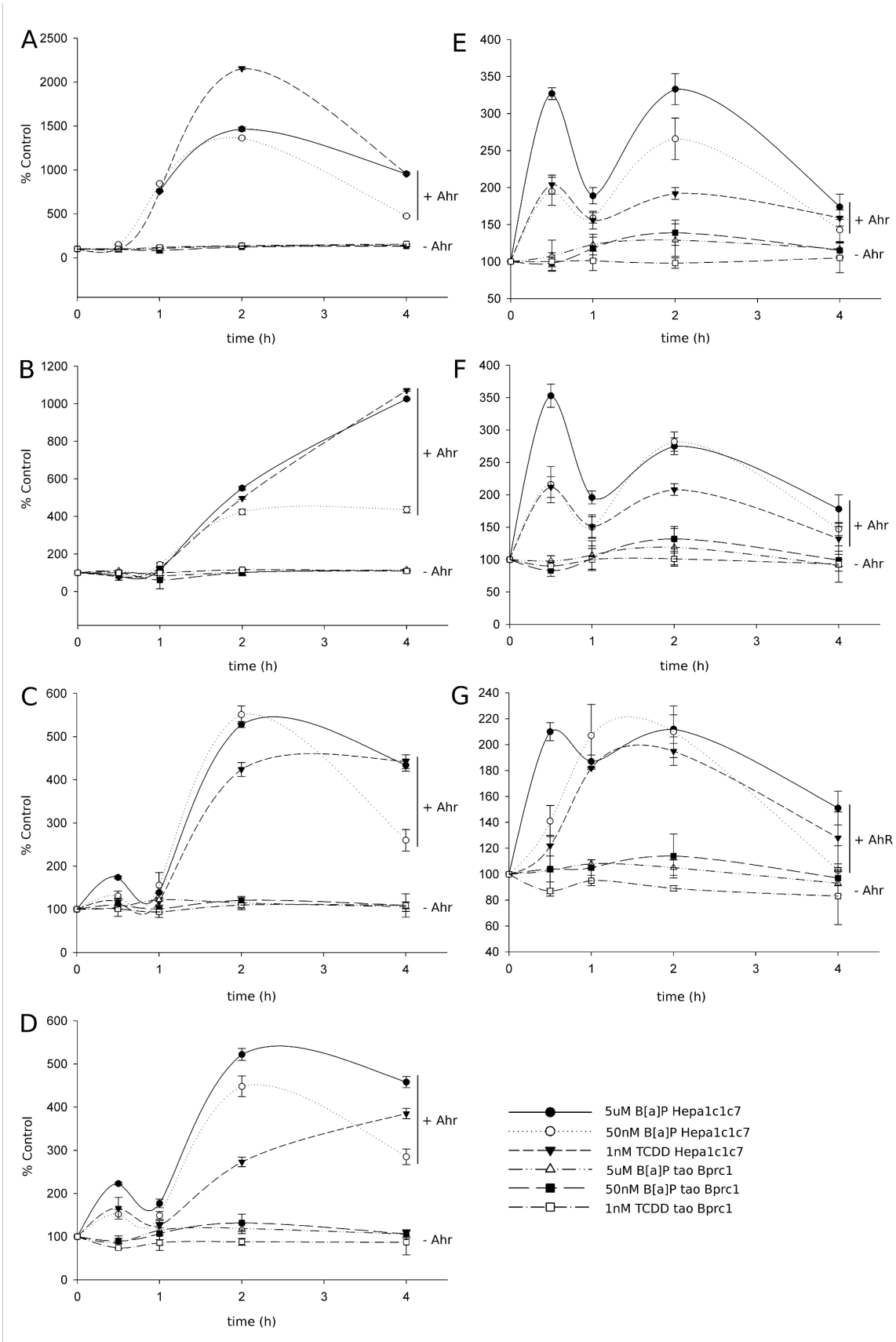


Figure 5.6 Confirmation of predicted targets of *Ahr*. A subset of predicted *Ahr* target genes was confirmed by qPCR. Hepa1c1c7 (+*Ahr*) and tao BpRc1 (-*Ahr*) were exposed to two different concentrations of B[a]P or TCDD respectively. Significant differential expression of *Tiparp* (A), *Tnfaip2* (B), *Mpp2* (C), *Cyp2s1* (D), *Nfe2l2* (E), *Cdkn1a* (F) and *Cdkn1b* (G) was detected exclusively in *Ahr*-expressing cells.

Gene	FC Enrichment
Tiparp	113.9 ± 30.2
Tnfaip2	5.9 ± 4.7
Mpp2	15.5 ± 1.5
Cyp2s1	41.6 ± 11
Nfe2l2	8.6 ± 6.5
Cdkn1a	55.8 ± 18.0
Cdkn1b	6.8 ± 2.1

Table 5.4 Binding of *Ahr* after exposure to B[a]P to promoter sequences of selected predicted targets, assayed by ChIP. Fold change is relative to vehicle control.

richment for *Ahr* binding to the corresponding promoter sites. Another gene known to be involved in cell cycle regulation, but less well-defined, is the palmitoylated membrane protein 2 (*Mpp2*). *Mpp2* was also strongly induced by TCDD and B[a]P in *Ahr*-expressing cells, while no differential expression was elicited in the mutant tao BpRc1 cells. A more indirect effect on cell cycle regulation originates from the TNF alpha activated signaling cascade. Five genes (*Tnfaip2*, *Tnfaip8*, *Traf5*, *Casp3*, *Ddx58*) involved in this pathway were predicted to be direct targets of *Ahr*. *Tnfaip2* and *Casp3* were investigated in our independent experimental confirmation. For both genes induction of expression was only detectable in Hepa1c1c7 cells, while the *Ahr*-deficient counterparts showed no significant differential regulation. Actual binding of *Ahr* to the promoter sites could be confirmed by ChIP. Direct regulation by *Ahr* of the important regulators of the cell cycle *Cdkn1a*, *Cdkn1b* as well as *Mpp2* together with targeting of the TNF alpha signaling pathway emphasizes the impact of this *Ahr* on endogenous cellular functions outside of xenobiotic metabolism.

5.4.2 An *Ahr* transcriptional cascade

Eleven of the genes in cluster 3 (i.e. the *Ahr* target cluster) are known transcriptional regulators. These regulators could constitute a transcriptional cascade that begins with the activation of *Ahr*.

Ahr has been connected to hormone-induced signaling as was reinforced by our GO enrichment analysis that identified "regulation of hormone levels" as one of the biological functions. Crosstalk with the estrogen receptor has been studied extensively (DuSell et al., 2010; Wihlén et al., 2009; Swedenborg and Pongratz, 2010) and glucocorticoid receptor (GR)/*Ahr* crosstalk has also been suggested (Wang et al., 2009; Vrzal et al., 2009). Our classifier predicted the glucocorticoid receptor (*Nr3c1*) itself as an *Ahr* target together with *Sgk1*, a GR-regulated kinase. In addition, the transcription factor *Klf9* known to be induced by GR and involved in adipogenesis, was predicted to be a direct *Ahr* target. Besides *Klf9*, further *Ahr* targets were predicted with an involvement in lipid synthesis and lipid transport, i.e. the transcriptional regulators *Ppard* and *Lpin* play a role in mammary lipid synthesis, and *Npc1*, *Osbpl2*, and *Pitpnc1* are involved in lipid transport. The role of GR in lipid homeostasis and metabolism is well-established (Michailidou et al., 2008; Marissal-Arvy et al., 2010; Hoppmann et al., 2010). From our analysis we can deduce a possible *Ahr*-activated network of genes directly influencing lipid status and its regulation by the glucocorticoid receptor.

The interaction of *Ahr* with another transcription factor *Nfe2l2* (aka *Nrf2*) might also have an influence on lipid status, specifically on adipogenesis (Shin et al., 2007). A bidirectional regulation of these two pathways has been described previously (Köhle and Bock, 2007). Both transcription factors have been shown to bind in the other's promoter region, thereby directly influencing transcription (Miao et al., 2005; Shin et al., 2007). Therefore, the prediction of *Nfe2l2* being an *Ahr* target is very well corroborated by

previous studies and was indeed verified by our experimental follow-up. In addition, a recently described interaction of *Nfe2l2* and *Ahr* confirms one other predicted *Ahr* target gene: *Abcc4*. Xu et al. showed that this multidrug resistant protein is directly activated by *Ahr* and *Nfe2l2* in liver (Xu et al., 2010).

Overall, our GO enrichment analysis together with a more detailed functional analysis of the predicted direct *Ahr* targets supported the accuracy of our classifier and subsequent weighted clustering. Moreover, we experimentally verified *Ahr* involvement for a representative subset of genes by differential expression as well as by binding analysis of *Ahr* to the corresponding promoter regions.

cluster	ID	Term	Annotated	Significant	Expected	P
1	GO:0051301	cell division	35	32	21.83	9.1e-05
1	GO:0019941	modification-dependent protein catabolic...	57	47	35.55	0.00064
1	GO:0006468	protein amino acid phosphorylation	81	63	50.52	0.00162
1	GO:0006260	DNA replication	13	13	8.11	0.00207
1	GO:0007067	mitosis	30	26	18.71	0.00299
1	GO:0016568	chromatin modification	26	23	16.22	0.00301
1	GO:0065002	intracellular protein transmembrane tran...	12	12	7.48	0.00334
1	GO:0016043	cellular component organization	202	155	125.99	0.00678
1	GO:0007265	Ras protein signal transduction	27	23	16.84	0.00841
1	GO:0006606	protein import into nucleus	10	10	6.24	0.00869
1	GO:0051128	regulation of cellular component organiz...	34	28	21.21	0.00924
2	GO:0007186	G-protein coupled receptor protein signa...	71	40	22.04	4.3e-06
2	GO:0015672	monovalent inorganic cation transport	19	12	5.90	0.0036
2	GO:0006952	defense response	30	16	9.31	0.0083
3	GO:0050793	regulation of developmental process	104	16	6.84	0.00064
3	GO:0042221	response to chemical stimulus	82	17	5.40	0.00070
3	GO:0010033	response to organic substance	42	9	2.76	0.00107
3	GO:0043086	negative regulation of catalytic activit...	14	5	0.92	0.00135
3	GO:0055114	oxidation reduction	54	10	3.55	0.00184
3	GO:0048522	positive regulation of cellular process	129	17	8.49	0.00262
3	GO:0010817	regulation of hormone levels	10	4	0.66	0.00268
3	GO:0008285	negative regulation of cell proliferatio...	24	6	1.58	0.00343
3	GO:0007050	cell cycle arrest	11	4	0.72	0.00400
3	GO:0046942	carboxylic acid transport	12	4	0.79	0.00570
3	GO:0032879	regulation of localization	36	7	2.37	0.00709
3	GO:0070887	cellular response to chemical stimulus	20	5	1.32	0.00763
3	GO:0006974	response to DNA damage stimulus	28	6	1.84	0.00777
3	GO:0006869	lipid transport	13	4	0.86	0.00783
3	GO:0048514	blood vessel morphogenesis	29	6	1.91	0.00929

Table 5.5 Enrichment of clusters for GO biological processes. Enrichment was calculated against the pooled annotations of all genes assigned to any of the three clusters (i.e. not against genome-wide annotations).

5.4.3 Secondary effects

Genes in clusters 1 and 2 are predicted to be perturbed not as a result of *Ahr* regulation, but by the presence of the metabolite BPDE. This genotoxic metabolite of B[a]P is known to cause DNA damage by DNA-adduct formation (Jack and Brookes, 1980; Mattsson et al., 2009). DNA damage is followed by initiation of DNA replication (GO:0006260), one of the eleven GO categories enriched in Cluster 1. Further, in the overall set of significantly regulated genes, many different MAP kinases are found, and all of these kinases are members of cluster 1 or 2. The idea that MAP kinases are *Ahr*-independent is supported by (Tan et al., 2002), who could show that *Ahr* ligands were equally effective activating MAPKs in cells that express *Ahr* and those deficient in the receptor.

To further support the predictions of our classifier, we selected some representative genes from clusters 1 and 2 (*Agfg1*, *Anapc1*, *Nfkb*, and *Parp1*) and measured their expression in response to exposure to B[a]P or BPDE in wild type (Hepa1c1c7) or mutant (tao BpRc1) cells (Fig. 5.7). These experiments demonstrate that BPDE causes differential expression with and without *Ahr*, while B[a]P only perturbs

expression in the presence of *Ahr*, i.e. when metabolism of B[a]P to BPDE is made more efficient by a functional *Ahr* pathway. These results demonstrate, as predicted, that these genes are regulated by the presence of BPDE and not directly by *Ahr*, further supporting the predictive power of our classifier.

5.4.4 Utility of weighted clustering

One unique and desirable aspect of the type of learning approach applied here is a side effect of the learning process – the proximity measure. The RF proximity is a type of similarity measure between subjects (in this case genes), based on how often two genes take the same path down the decision trees of the forest. It is in effect a weighted similarity measure because only time points that are useful in the learning process are used in the calculation of the proximity. This is in contrast to the widely used Euclidean distance, in which all features make an equal contribution.

A weighted (dis)similarity measure is advantageous in clustering gene expression time series, especially in complex transcriptional responses of higher eukaryotes, as presented in this work. Additional systems are present in higher eukaryotes that influence the synthesis, stabilization, and degradation of mRNA. These additional systems make it less likely that functionally related genes share precisely the same characteristic expression profile over time. For instance, functionally related genes, induced by a common transcription factor, may share similar expression patterns shortly after induction, but may then diverge as other factors come into play, such as microRNAs. A supervised, weighted metric such as RF proximity could de-emphasize the diverging time points while emphasizing the common phase of induction, finally resulting in the grouping of the functionally related genes together. Conversely, such expression profiles are unlikely to fall into the same cluster when using e.g. the Euclidean distance, and could be a contributing factor to the mixed success of past attempts (Dere et al., 2006; Frericks et al., 2008; Kim et al., 2009a; Boutros et al., 2009) to cluster *Ahr*-induced gene expression time courses in a way that is biologically interpretable.

One technique that is frequently used to address problems such as those described here is biclustering (Supper et al., 2007; Gonçalves et al., 2009; Madeira and Oliveira, 2009; Madeira et al., 2010; Wang et al., 2010a). Briefly, biclustering is a strategy that seeks to cluster in two dimensions simultaneously, e.g. genes and time points. The goal is to find genes that show similar expression in some (though not necessarily all) conditions. There are many algorithms and heuristics that implement biclustering. Strengths and weaknesses of the approach vary by implementation, but in general most biclustering methods are unsupervised and are non-deterministic. Without alleviating assumptions the problem is NP-hard. It can be difficult to judge the quality of the resulting clusters, and clusters are often redundant. In the work presented here, clustering with the RF proximity presented fewer potential pitfalls compared to biclustering, since we had a means of performing supervised learning and the RF proximity was obtained "for free" since it was part of the learning process. In addition, the clusters were non-redundant and judging their quality was fairly straightforward by using another Random Forest to predict the assigned cluster labels of the genes (as described in the methods section). In addition to the work presented here, clustering of genomic data with an RF proximity has been described in (Breiman and Cutler, 2003; Allen et al., 2003; Shi et al., 2005; Shi and Horvath, 2006), and an example using multivariate response Random Forests to examine transcriptional programs in yeast can be found in Xiao and Segal (Xiao and Segal, 2009). We have found that PAM clustering with the RF proximity measure works well in scenarios where weighted clustering is desirable, and is an alternative to biclustering that is worth considering. However, one ob-

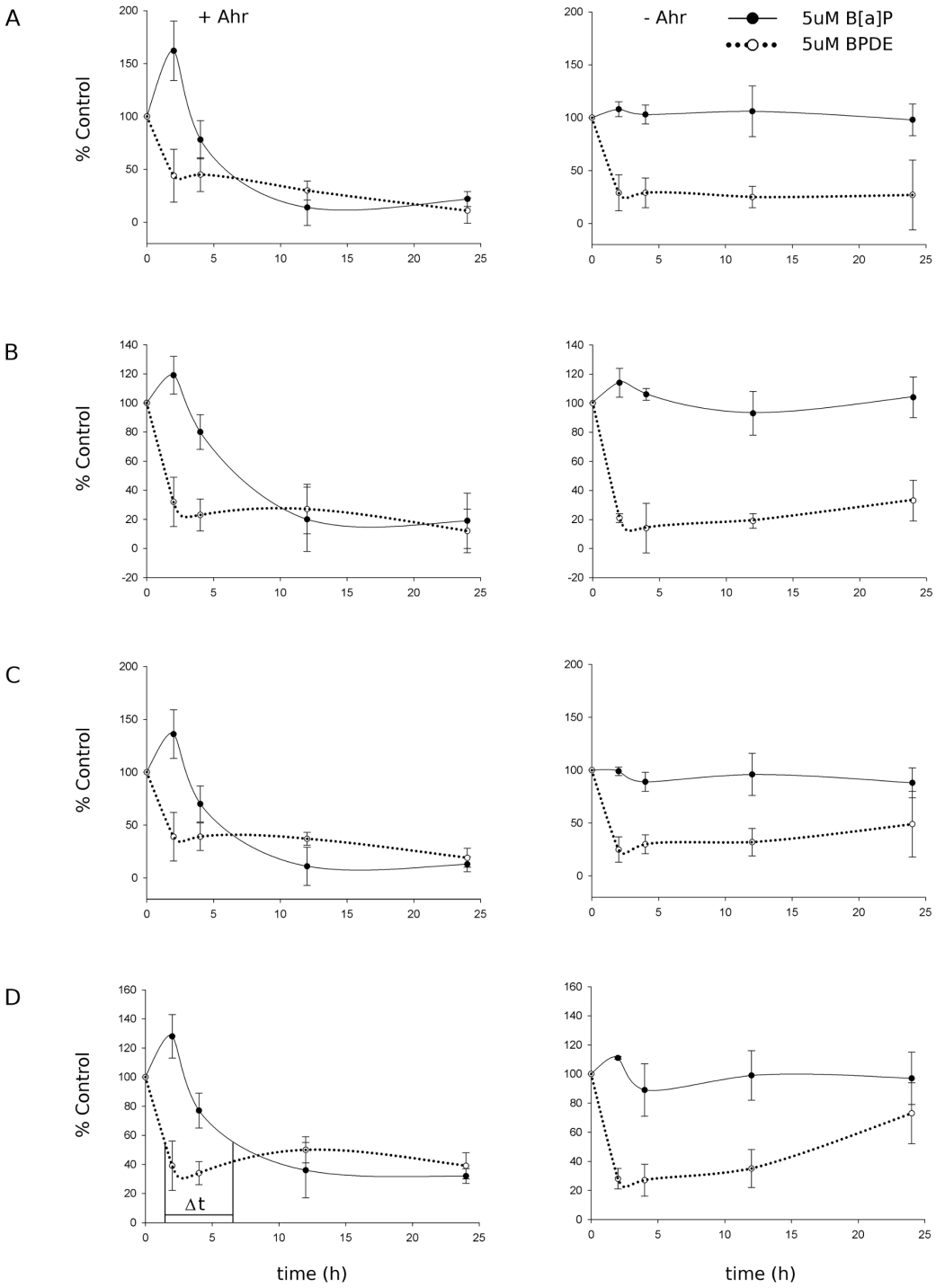


Figure 5.7 Confirmation of predicted BPDE-perturbed genes. qPCRs for *Agfg1* (A), *Anapc1* (B), *Nfkb* (C), *Parp1* (D) were performed in *Ahr* expressing (Hepa1c1c7, +*Ahr*) and in *Ahr* deficient cells (tao BpRc1, -*Ahr*). Cells were exposed to B[a]P or its active metabolite BPDE. Transcriptional response to BPDE was comparable in both cell types. However, since B[a]P will be metabolized to BPDE only in +*Ahr*, but not in -*Ahr* cells, no differential expression of these genes was detectable in -*Ahr* cells for B[a]P exposure, while in +*Ahr* cells differential expression was observed with a time lag (Δt) compared to exposure to the BPDE itself.

vious limitation for any supervised method – including our use of RF here – is the need for a training set. In some situations a training set may be difficult or impossible to assemble – this is an important consideration when selecting a clustering method.

5.5 Author contributions and acknowledgements

Saskia Trump and I drafted the manuscript together. Saskia Trump either performed or directed all experimental work, Franziska Dautel raised cells for the experiment, Carolin Gräbsch and Susanne Rudzok performed confirmation experiments. Danielle Madureira and Kristin Schirmer coordinated the processing of the microarrays. Andreas Beyer and I devised the analytical strategy and I performed all computational work. Irina Lehmann, Martin von Bergen, Kristin Schirmer, and Andreas Beyer functioned in an editorial and supervisory capacity.

Chapter 6

Conclusion and outlook

6.1 Contributions of this dissertation

The work presented in this dissertation led to several specific contributions that are of particular importance to researchers studying transcriptional regulation and its role in disease and responses to the environment. These contributions are presented here in response to the open problems that motivated their development.

6.1.1 Open problem 1 revisited

Open problem How can the performance of eQTL mapping methods be tested using measured data? Can modifications of these methods produce improved results?

Key contributions

We presented a battery of three data-driven benchmarks to assess the performance of competing eQTL mapping methods: the proportion of cis-eQTL recovered, the enrichment of eQTL for genes involved in the same pathway as the target, and the enrichment of eQTL for genes causing expression perturbation in the target when mutated. These benchmarks suggested that multi-locus modeling methods are better at pinpointing loci that are supportable by real data. Specifically, Random Forests (RF) were shown to have the best performance overall, and selection frequency, a novel method for mapping eQTL, was shown to be the most effective RF importance measure in this context.

6.1.2 Open problem 2 revisited

Open problem How can epistasis be efficiently discovered among millions of locus pairs and tens of thousands of traits? How can competing methods be benchmarked using real data?

Key contributions

Because we found Random Forests to be a good performer for mapping eQTL, we were motivated to find a way to extract epistatic relationships between loci from the forest structure. In this way, we could essentially find interactions "for free" while performing a conventional eQTL map. We introduced the concept of RF split asymmetry which finds dependencies between splitting variables in the structure of the forest.

Split asymmetry is caused by epistasis, and is therefore a means to detect it. To demonstrate the method's utility, we were not satisfied with simulated data alone; we devised a test using several genetic interaction data sets in yeast and showed that RF split asymmetry did a better job of recovering engineered interactions using eQTL data than the competing linear model-based approach. Encouraged by these results, we looked for epistasis controlling the expression of mouse orthologs of schizophrenia risk genes. The resulting epistatic interactions made intuitive sense, based on what is known about the processes underlying schizophrenia.

6.1.3 Open problem 3 revisited

Open problem Which disease-associated genes are most likely to be causal, and which are likely to be symptomatic? How can systems genetics data be used to give clues about the etiology of a disease or the drivers of a phenotype?

Key contributions

Network approaches are frequently used to interpret eQTL data and construct transcriptional regulatory networks. However, their ability to recover the regulatory roles of genes is impaired when the data are noisy. We developed a network-free approach that uses distributional enrichment of eQTL scores, rather than an arbitrary threshold, to define the nature of a gene's regulatory role relative to other disease-related genes. This approach can be more sensitive than an explicit network. We demonstrated its use in connection with schizophrenia risk genes, and derived a regulatory hierarchy based on systems genetics data in mouse. We found that the measure of how upstream a gene was in the mouse hierarchy was a good indicator of how reproducible an association with schizophrenia would be in human studies, suggesting that genes that are higher in the hierarchy are more likely to be causal. In addition, the most significant of the genes had a very suggestive role in the molecular etiology in schizophrenia that is well-supported by the literature.

6.1.4 Open problem 4 revisited

Open problem Given an extensive transcriptional response upon induction of a transcription factor, how can direct targets be distinguished from indirect effects? How can the responding genes be clustered in functional groups in a way that accounts for individual transcript differences in synthesis and degradation?

Key contributions

The data from our *Ahr* experiments comprised an enormous transcriptional response – 2,338 genes. We used data from independent studies to derive training labels so that we could train an RF classifier to recognize the difference between direct targets of *Ahr* and secondary effects caused by toxic metabolites. Predictions were then made for all differentially expressed genes. This was a crucial step in making sense of the enormous transcriptional response. In addition, the training of the classifier led to a supervised and weighted distance measure that was used to cluster the genes. The fact that a weighted distance was used was vital because the direct targets of *Ahr* were only co-expressed during the induction phase, diverging noticeably at later time points. Thus, these early time points of induction are given more weight in the

distance calculation, leading to a more appropriate clustering of functionally related genes. This demonstration of weighted clustering is especially useful for researchers working on time course gene expression data, where functionally related genes might not always show consistent co-regulation throughout the time series.

6.2 Outlook

The methods presented in this work show promise beyond the applications demonstrated. For instance, our method for mapping regulatory hierarchies could be applied to large-scale screen data, such as that from RNAi screens. An approach similar to our demonstration of expression time course classification coupled with weighted clustering could be used with data relating to stem cell differentiation – this could establish which time points are crucial for which functional gene groups.

New applications of our methods will undoubtedly expose further weaknesses in their implementation and provide an opportunity for their improvement. Indeed, there are already some well-known limitations of methods based on Random Forests (RF). For instance, because it is a stochastic method, its results are not strictly reproducible. Biologists are often incredulous when told that a particular method does not give the same answer each time you run it. The following "philosophical note" by Leo Breiman and Adele Cutler provides a good perspective on the practical use of RF:

> RF is an example of a tool that is useful in doing analyses of scientific data. But the cleverest algorithms are no substitute for human intelligence and knowledge of the data in the problem. Take the output of random forests not as absolute truth, but as smart computer generated guesses that may be helpful in leading to a deeper understanding of the problem.

In this way, RF can be seen as a tool that, when properly applied, can point the researcher in the right direction. However, it might not always be appropriate as a final result in and of itself. Although we repeatedly found that RF performed very well when applied to real biological datasets, we still recommend careful interpretation of specific results, and where possible, corroboration of interesting results with complimentary methods and data.

Perhaps nothing has become more clear to me through the course of my doctoral work than the fact that there are veritable mountains of biological data available, and that we have only scratched the surface of what they have to tell us about how life works. The sheer breadth of what a modern high-throughput experiment collects almost always exceeds the scope of the question that motivated its investigation. Our ability to generate data is not yet matched by our ability to fully interpret it. This represents a challenge and an opportunity for computational biologists to both continue to develop sophisticated analytical methods, and to work more effectively in promoting their adoption and use among non-computational biologists. This will not only ensure many more new discoveries from old data, but will improve the balance between newly acquired data and newly acquired understanding.

Appendix A

Selected functions

The following R functions are used in the tutorials included in this appendix. All functions were written by the author.

A.1 Simulate genotypes

```
1  ##############################################################################
2  ### a function to simulate (0,1) genotype markers with tunable
3  ### linkage disequilibrium
4  ### n=number of individuals
5  ### nmarkers=number of markers
6  ### recombprob=probablities of recombination; at each successive locus, one of
7  ### these values is sampled randomly to determine how many of the individuals
8  ### will "flip" their genotype; the more '0' values in this vector, the more
9  ### LD there will be in the markers.  Higher values will result in more
10 ### recombination.
11 ##############################################################################
12
13 simgeno = function(n,nmarkers=1000,recombprob=c(0,0,0,0,0.01,0.3)){
14  geno = matrix(0,n,nmarkers)
15  xn = sample(c(0,1),n,replace=T)
16  geno[,1] = xn
17  draw = ceiling(recombprob*n)
18  for(i in 2:nmarkers){
19   these = sample(1:n,sample(draw,1))
20   xn[these] = !xn[these]
21   geno[,i] = xn
22  }
23  return(geno)
24 }
```

A.2 Simulate a trait

```
################################################################################
### a function to simulate traits, given an input logical vector and an effect
### size for those "with" the trait (i.e. TRUE)
### logic=a logical vector indicating which samples "have" the trait (TRUE) and
### which do not (FALSE)
### effect=effect size (separation between those that have the trait and those
### that don't)
### spread=value that controls the spread (SD) of the distributions
################################################################################

simtrait = function(logic,effect=1,spread=0.1){
 spread = spread*abs(effect)
 logic = as.logical(logic)
 out = numeric(length(logic))
 out[!logic] = rnorm(sum(!logic),0,spread)
 out[logic] = rnorm(sum(logic),effect,spread)
 return(out)
}
```

A.3 Extract selection frequencies

```
################################################################################
### function for extracting selection frequencies from an RF
### rf=the RF from which selection frequencies are desired
################################################################################
rfsf = function(rf){
 vu = randomForest::varUsed(rf)
 sf = vu/sum(vu)
 names(sf) = rownames(rf$importance)
 return(sf)
}
```

A.4 Estimate score bias in RFSF

```
##############################################################################
### function to estimate the selection bias in RF when the null hypothesis is
### true (i.e. no association between response and predictors)
### x=predictor matrix (genotype matrix)
### ntree=number of trees
### verbose=print progress?
### ...=additional arguments to be passed to 'randomForest'
##############################################################################

estBias = function(x,ntree,verbose=TRUE,mult=FALSE,...){
 if(ntree%%10!=0) stop("ntree must be a multiple of 10")
 vu = numeric(ncol(x))
 ni = ntree/10
 for(i in 1:ni){
  rf = randomForest::randomForest(y=rnorm(nrow(x)),x=x,ntree=10,...)
  vu = vu + randomForest::varUsed(rf)
  if(i%%10==0 && verbose){
   cat(round(100*i/ni,0),"percent complete")
   cat("\n")
  }
 }
 vu = vu/sum(vu)
 if(mult){
  return(mean(vu)/vu)
 }else{
  return(vu-mean(vu))
 }
}
```

A.5 Plot a density with data points

```
#########################################################################
### plot density with data points
#########################################################################
pdens = function(x,...){
 plot(density(x),...)
 points(x,rep(0,length(x)),col='red',pch="|")
}
```

A.6 Extract M_l and M_r from a Random Forest

```
1  ##########################################################################
2  ### compile information needed for EPI() from an rfdev randomForest
3  ### object
4  ### rf=randomForest object (must be created with rfdev package)
5  ### testset=logical indicating whether the testset (OOB) predictions
6  ### or the training predictions should be used
7  ##########################################################################
8  predsymm = function(rf,testset=TRUE){
9   l = matrix(0,nrow=nrow(rf$importance), ncol=nrow(rf$importance))
10  colnames(l) = rownames(rf$importance)
11  rownames(l) = rownames(rf$importance)
12  r = l
13  ct = l
14
15  xx = rf$forest$bestvar
16  rr = rf$forest$rightDaughter
17  ll = rf$forest$leftDaughter
18  if(testset){
19  pred = rf$forest$oobpred
20  }else{
21  pred = rf$forest$nodepred
22  }
23  xx[xx==0]=NA
24  rr[rr==0]=NA
25  ll[ll==0]=NA
26
27  for(i in 1:ncol(xx)){
28   if(all(is.na(xx[,i]))) next
29   p = pred[ll[,i],i] - pred[rr[,i],i]
30   ind = 1:nrow(xx)
31   parents = xx[ifelse(ind%%2==0,match(ind,ll[,i]),match(ind,rr[,i])),i]
32   ind = cbind(1:nrow(xx),parents,xx[,i],p)
33   ind = ind[apply(ind,1,function(x) all(!is.na(x))),]
34
35   if(is.null(nrow(ind))) next
36   p = ind[,4]
37   side = ind[,1]
38   ind = ind[,2:3]
39
40
```

```
  il = side%%2!=0
  ir = !il
  l[ind[il,]] = l[ind[il,]] + p[il]
  r[ind[ir,]] = r[ind[ir,]] + p[ir]
  ct[ind] = ct[ind] + 1
 }
 return(list(l=l/rf$ntree,r=r/rf$ntree,counts=ct/sum(ct)))

}
```

A.7 Score epistasis with RF split asymmetry

```
#############################################################################
### score epistasis using the output from predsymm()
### x=output from predsymm()
#############################################################################
EPI = function(x){
 d = abs(x$l-x$r)
 m = (abs(x$l)+abs(x$r))/2
 out = d-m
 out = out
 out = pmin(out,t(out))
 out[out<0] = 0
 return(out)

}
```

A.8 Plot an eQTL map with epistatic connections

```
##########################################################################
### plot eQTL map with epistatic connections
### rf=randomForest object (must be created with rfdev package)
### P=matrix of P values
### cutoff=FDR threshold to be used
### main=title of plot
##########################################################################
plotEpi = function(rf,P,cutoff=0.01,main=""){
 P[!upper.tri(P)] = NA
 P = as.data.frame(as.table(P))
 P = as.matrix(P[!is.na(P[,3]) & p.adjust(P[,3],'fdr')<cutoff,1:2])
 mode(P) = 'numeric'
 yl = max(rf$importance[,1])
 yl = c(-0.25*yl,1.1*yl)
 plot(rf$importance[,1],type='h',ylim=yl,axes=F,ylab='importance',
 main=main,col=grey(0.2))
 if(nrow(P)>0){
  apply(P,1,function(x) arc(min(x[1:2]),max(x[1:2]),0.7*yl[1],col='coral1'))
 }
 axis(1)
 axis(2,at=pretty(c(0,yl[2]),4))
}
## function to draw the connections
arc = function(x1,x2,depth=-1,col){
 x = c(x1,0.5*(x2-x1)+x1,x2)
 y = c(0,depth,0)
 lines(spline(x,y,x2-x1),col=col)
}
```

A.9 Get upstreamness and centrality from eQTL data

```
#########################################################################
### get upstreamness and centrality scores from eQTL data
### eqtl=a matrix of eQTL scores (with row and column names)
### genes=a named logical vector indicating which genes
### are interesting
### markers=a named logical vector indicating which markers map to the
### interesting genes
### cis.map=a named character vector of marker names, whose names are
### gene names; this indicates the mapping from genes to markers
### nulldist=logical; indicates if null distribution values should
### be obtained
#########################################################################
ucScores = function(eqtl,genes,markers,cis.map,nulldist=FALSE){
 if(nulldist){
  genes = structure(sample(genes),names=names(genes))
  markers = colnames(eqtl)%in%cis.map[names(genes)[genes]]
  names(markers) = colnames(eqtl)
 }
 dwn = apply(scale(eqtl[genes,]),1,
  function(x) ks.test(x[markers],x[!markers],alternative='less')$stat)
 up = apply(eqtl[,markers],2,
  function(x) ks.test(x[genes],x[!genes],alternative='less')$stat)
 up = up[cis.map[names(genes)[genes]]]
 names(up) = names(genes)[genes]
 s = up-dwn
 centrality = (up+dwn)-abs(s)
 return(list(upstreamness=s,centrality=centrality))
}
```

A.10 Get *P* values from a 2D empirical null distribution

```
##############################################################################
### get P values from a 2D empirical null distribution
### x=observed x values
### y=observed y values
### xn=x values under the null
### yn=y values under the null
### ...=additional arguments to be passed to kde2d
##############################################################################
p2d = function(x,y,xn,yn,...){
 require(MASS)
 rgy = range(c(y,yn))
 rgx = range(c(x,xn))
 kd = kde2d(xn,yn,lims=c(rgx,rgy),...)
 z = numeric(length(x))
 names(z) = names(x)
 for(i in 1:length(z)){
  z[i] = kd$z[which.min(abs(x[i]-kd$x)),which.min(abs(y[i]-kd$y))]
 }
 p = numeric(length(x))
 names(p) = names(x)
 for(i in 1:length(p)) p[i] = sum(kd$z[kd$z<z[i]])/sum(kd$z)
 return(p)
}
```

A.11 Get quantiles of a 2D density

```
##############################################################################
### get quantiles of a 2D density (used to draw contour lines)
### x=x values
### y=y values
### quant=quantile desired
### ...=additional arguments to be passed to kde2d
##############################################################################
q2d = function(x,y,quant=0.05,...){
 require(MASS)
 kd = kde2d(x,y,...)
 dens = sort(as.numeric(kd$z))
 cdens = cumsum(dens)/sum(dens)
 out = numeric(length(quant))
 names(out) = as.character(quant)
 for(i in 1:length(quant)){
  ind = max(which(cdens <= quant[i]))
  out[i] = dens[ind]
 }
return(out)
}
```

A.12 Retrieve PubMed IDs via NCBI's eUtils

```
##########################################################################
### function to retrieve PMIDs associated with a search term
### term=text to search for (character vector of length 1)
### retmax=number of IDs to return (default all)
##########################################################################
getIDs = function(term,retmax=NULL){
 require(XML)
 if(!is.null(retmax)){
  ct = retmax
 }else{
  ct = getCount(term)
 }
 term = gsub(" ","+",term,fixed=T)
 uri = "http://www.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&usehistory=y"
 uri = paste(uri,"&term=",term,"&retmax=",ct,sep="")
 webenv = XML::xmlTreeParse(uri)
 r = XML::xmlRoot(webenv)
 out = XML::xmlSApply(r[["IdList"]],xmlValue)
 return(out)
}
```

A.13 Assessment of topic-specific gene citation significance

```
##########################################################################
### Construct 2x2 contingency tables and run Fisher's exact test,
### given:
### gene=vector of Entrez gene IDs cited in target publications
### dat=3-column matrix: Taxonomy ID, Gene ID, PubMed ID
### wgene=vector of PMIDs that cite at least one gene of interest
##########################################################################

fetSpec = function(gene,dat,wgene){
 gene = as.integer(gene)
 if(length(gene)==1){
  y1 = dat[,2]==gene
  y2 = dat[,3]%in%wgene
  tt = ftable(y1,y2)
  out = list(table=tt,test=fisher.test(tt,alternative="g"))
 }else{
  y2 = dat[,3]%in%wgene
  out = vector("list",length(gene))
  names(out) = gene
  for(i in 1:length(gene)){
   y1 = dat[,2]==gene[i]
   tt = ftable(y1,y2)
   out[[i]] = list(table=tt,test=fisher.test(tt,alternative="g"))
   if(i%%100==0) print(i)
  }
 }
 return(out)
}

### a 'fast' 2x2 table
ftable = function(y1,y2){
 m = matrix(c(0,0,0,0),2,2)
 colnames(m) = c("F","T")
 rownames(m) = c("F","T")
 m[1,1] = sum(!y1 & !y2)
 m[1,2] = sum(!y1 & y2)
 m[2,1] = sum(y1 & !y2)
 m[2,2] = sum(y1 & y2)
 return(m)
}
```

Appendix B

Tutorial: Using the bias-corrected RFSF to map eQTL

B.1 Introduction

As described in Chapter 2, we found that Random Forests mapped the most biologically-supportable eQTL, when compared to other available methods. It has previously been demonstrated that Random Forests importance scores, including RFSF, are biased when there is strong correlation in the predictors (in the case of eQTL, genotype data) (Strobl et al., 2007; Altmann et al., 2010; Goldstein et al., 2010). Here we present a method and strategy for harnessing RF's usefulness in mapping eQTL while at the same time correcting for the bias induced in the importance scores by the correlations in the genotype data.

B.2 Simulating data

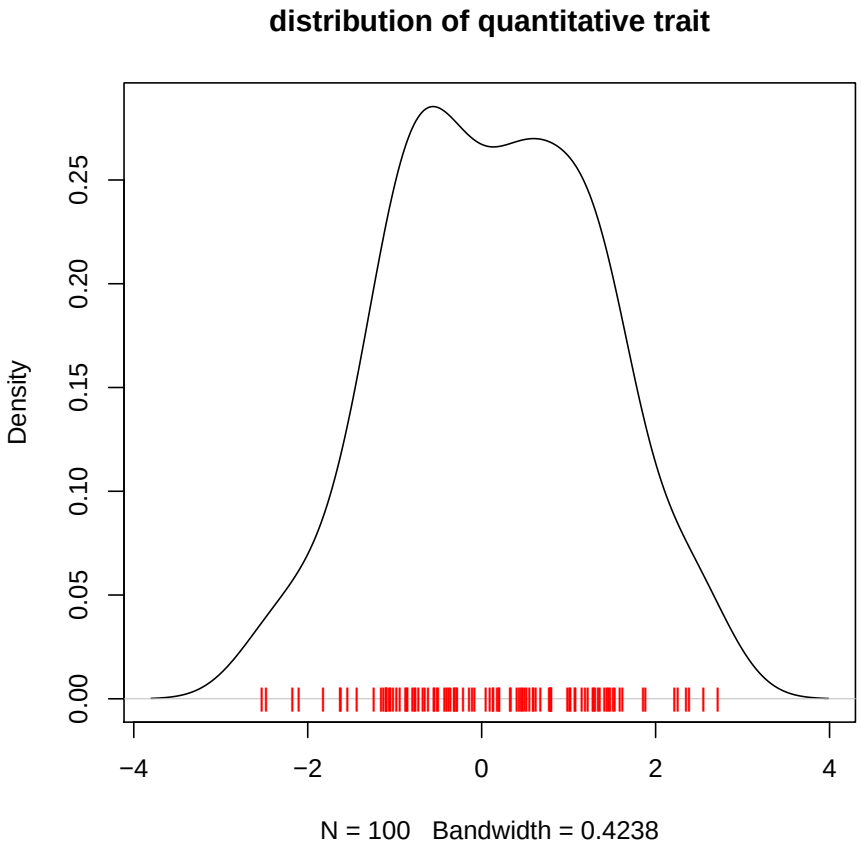
All that is needed to map eQTL with Random Forests is a matrix of genotypes and a vector of gene expression values. Although there are many publicly available expression and genotype data sets (see the GeneNetwork website, for instance), for convenience, we simulate these data here.

First we simulate the genotypes for 100 individuals:

```
> source("functions.R")
> set.seed(2341)
> x = simgeno(100)
```

Then we simulate a trait that is the additive effect of three loci (the 200^{th} marker, the 500^{th} marker, and the 750^{th} marker):

```
> set.seed(213756)
> y = simtrait(x[, 200], 1, 0.4) + simtrait(x[, 500], 0.75) - simtrait(x[,
+     750], 1.5, 0.3)
> pdens(y, main = "distribution of quantitative trait")
```



B.3 Mapping eQTL with RFSF

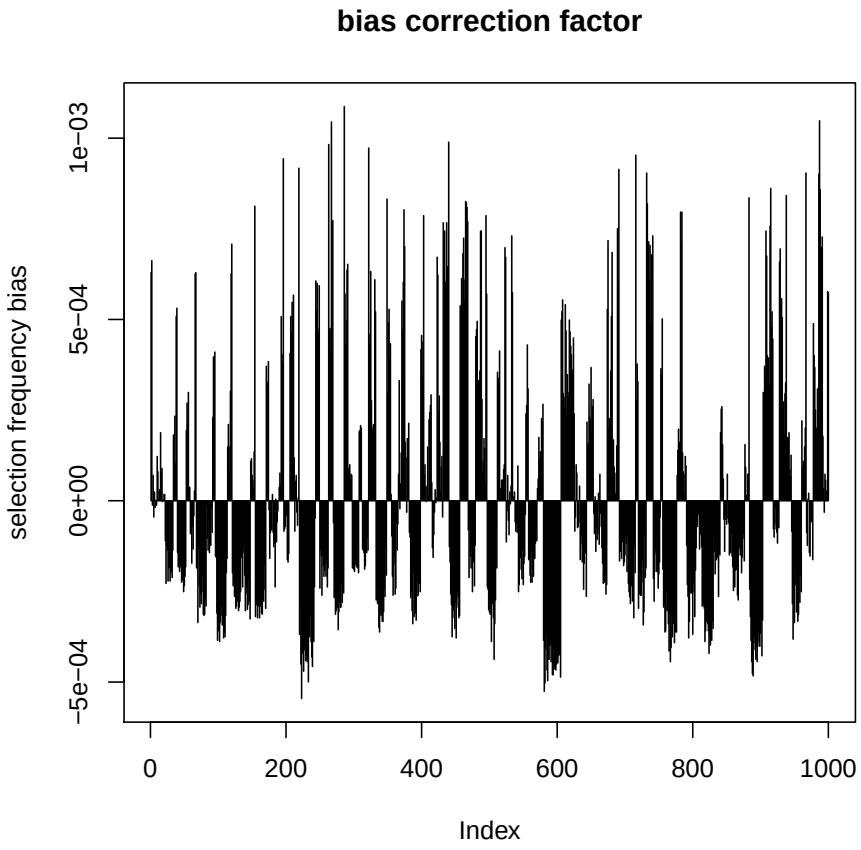
Next we map the eQTL with Random Forests, and extract the selection frequencies:

```
> library(randomForest)
> rf = randomForest(y = y, x = x, ntree = 2000)
> sf = rfsf(rf)
```

B.4 Estimating and accounting for selection bias

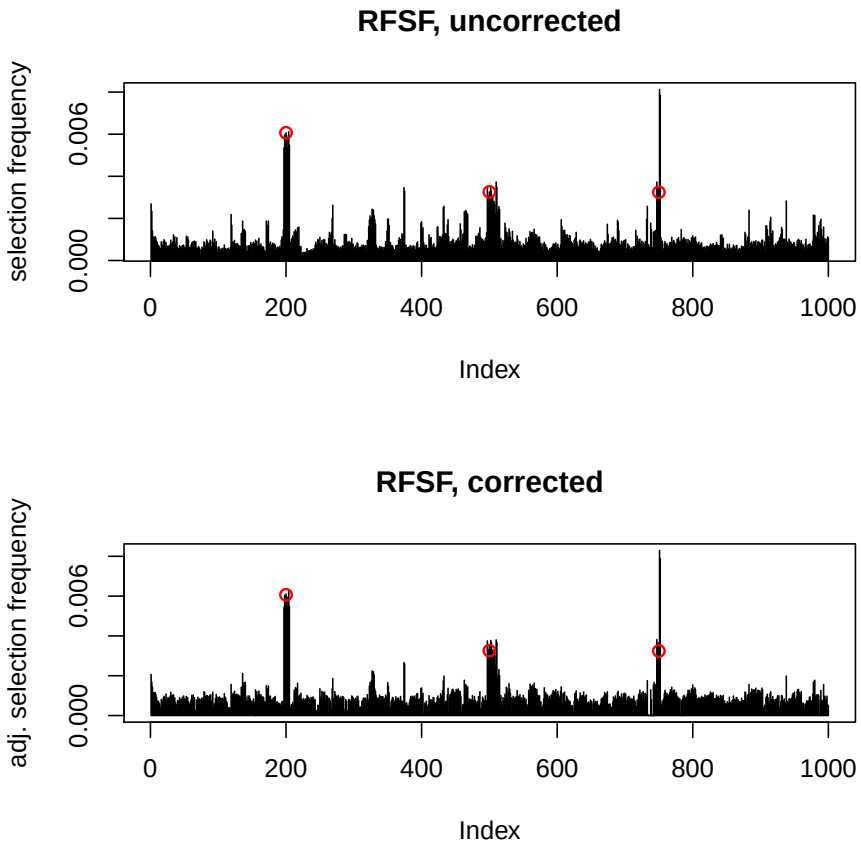
Now we estimate the marker-specific selection bias when we know the null hypothesis to be true (no association between genotypes and trait). The more trees we use, the more stable the estimate — here we use 10,000 trees. Also note that whatever additional or non-default arguments (e.g. `mtry` or `nodesize`) are used when mapping should also be passed to `estBias()`.

```
> corr = estBias(x, 10000, verbose = F)

> plot(corr, type = "h", ylab = "selection frequency bias",
+     main = "bias correction factor")
```



At this point we can just subtract `corr` from `sf` to get the corrected selection frequencies. Notice the peaks that vanish after the correction. Here we have fit a synthetic trait with a moderate amount of noise. When fitting traits with a high signal to noise ratio, the effect of bias (and hence its correction) is less noticable — the background effects of the bias are very small in comparison to the signal.

```
> sf.corr = sf - corr
> sf.corr[sf.corr < 0] = 0

> par(mfrow = c(2, 1))
> plot(sf, type = "h", ylab = "selection frequency", main = "RFSF, uncorrected")
> points(c(200, 500, 750), sf[c(200, 500, 750)], col = "red", lwd = 1.5)
> plot(sf.corr, type = "h", ylab = "adj. selection frequency",
+     main = "RFSF, corrected")
> points(c(200, 500, 750), sf[c(200, 500, 750)], col = "red", lwd = 1.5)
```



The estimated selection bias is independent of the response — it is only a function of the predictors (genotypes). This means that the same bias correction can be used for each new expression trait — it does not need to be re-estimated.

B.4.1 Multiplicative bias correction

In the paper where this idea was first presented, the bias correction was introduced as an additive correction. The utility of this approach was validated through simulations as well as on real data, where bias-corrected results were compared to results of an approach that had a separate empirical null distribution for each marker – results were essentially the same. A new, experimental approach suggests that a multiplicative correction – instead of the additive correction previously introduced – might perform better in some situations. This correction factor is calculated with the `estBias` function using the `mult=TRUE` argument:

```
> corr = estBias(x, 10000, verbose = F, mult = T)
```

The correction is then applied via multiplication, rather than subtraction:

```
> sf.corr = sf * corr
```

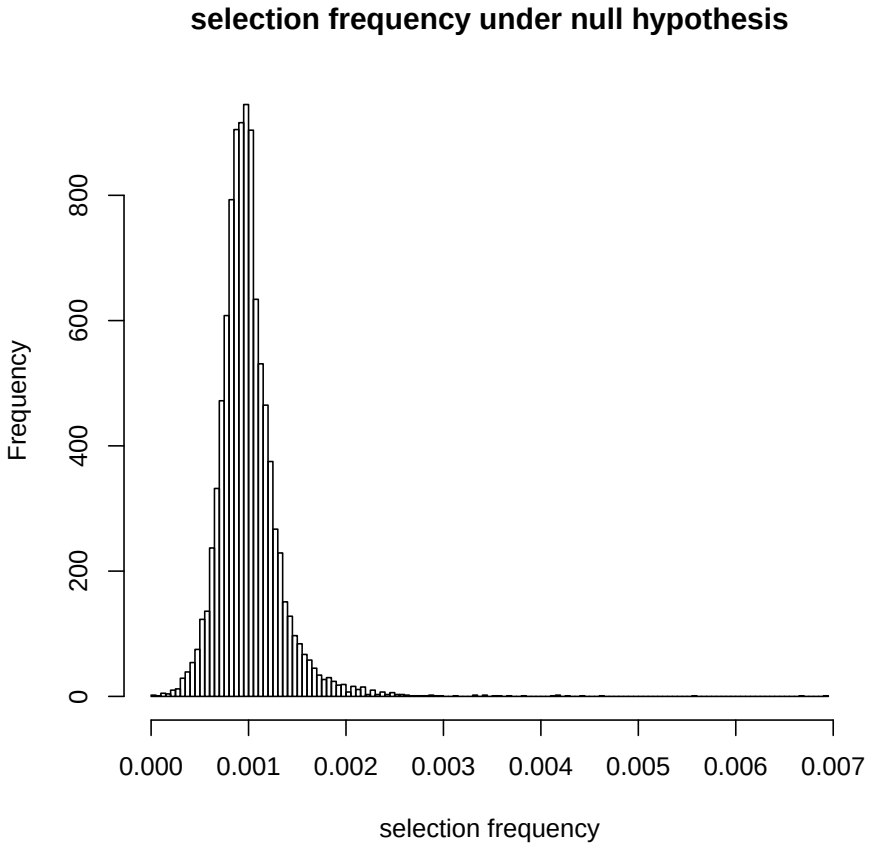
This correction factor uses ratios relative to the mean selection frequency (under the null hypothesis) as coefficients. That is, if a marker is selected at a frequency of half the mean selection frequency, it is multiplied by 2 in order to correct.

This approach is experimental and needs further validation using both simulated and real data sets – use it at your own risk.

B.5 Significance

The bias-adjusted RFSF can be useful as-is for ranking markers for further analysis. However, if *P* values are desired, a few more steps must be taken. First, we need to get an idea of what kind of selection frequencies we can expect under the null hypothesis:

```
> sf.null = numeric()
> for (i in 1:10) {
+     rf.null = randomForest(y = sample(y), x = x, ntree = 2000)
+     tmp = rfsf(rf.null) - corr
+     tmp[tmp < 0] = 0
+     sf.null = c(sf.null, tmp)
+     print(i)
+ }

> hist(sf.null, main = "selection frequency under null hypothesis",
+     xlab = "selection frequency", breaks = 100)
```



We now have an empirical null distribution of the selection frequencies. Using the `ecdf()` function, we can get estimates of *P* values. Note, however, that since we only have 10,000 data points in our null distribution, any *P* value less than 0.0001 will be 0. In other words, we cannot estimate *P* values between 0.0001 and 0 without adding more data points to the null distribution. An alternative might be to fit a parametric distribution (such as a mixture of beta densities) to the empirical null distribution, but for simplicity's sake, we'll use the empirical null distribution to get *P* values.

```
> pnull = ecdf(sf.null)
> P = 1 - pnull(sf)
```

Adjust for multiple testing and check what the FDR is at our "causal markers" (your numbers may vary since we did not use `set.seed()` when generating the null distribution).

```
> Q = p.adjust(P, "fdr")
> sum(Q < 0.05)

[1] 11

> Q[c(200, 500, 750)]

[1] 0.02727273 0.06923077 0.06923077
```

B.6 Running RF in parallel

Running RF over tens of thousands of expression traits can be time consuming. However, it is quite straightforward to take advantage of multi-core hardware to make the problem tractable. We use the `snow` package to create a cluster that distributes the workload over several processors or cores.

```
> library(snow)
> cl = makeSOCKcluster(8)
> clusterExport(cl, "rfsf")
```

Next, we write a small wrapper function that will run RF given a response (expression) and a matrix of predictors (markers), and then return the selection frequencies.

```
> ff = function(y, x, ntree) {
+     rf = randomForest::randomForest(y = y, x = x, ntree = ntree)
+     sf = rfsf(rf)
+ }
```

As an example here, we create a toy matrix `expr` of expression traits, where each row corresponds to a gene and the columns to individuals.

```
> expr = matrix(rnorm(8 * 100), 8, 100)
```

Finally, we use the `parApply` function to parallelize the mapping of eQTL with RF. This distributes the mapping tasks as evenly as possible across the available processors (which were determined when we called `makeSOCKcluster`).

```
> sf = parApply(cl, expr, 1, ff, x, 1000)
```

Now that we have the selection frequencies, we apply the bias correction:

```
> sf = t(sf - corr)
```

Appendix C

Tutorial: Finding epistasis in Random Forests

C.1 Introduction

Epistasis – interactions between genetic loci – is an important contributing factor to complex traits such as disease susceptibility (Carlborg and Haley, 2004). Observation of epistatic relationships between genes has been used as a means to infer molecular pathways and functional networks (Costanzo et al., 2010; Schuldiner et al., 2005; Hannum et al., 2009). Defining these epistatic relationships is a desirable feature for eQTL mapping methods, as it may serve to better clarify the regulatory relationships between targets and regulators. We previously found that Random Forests (RF) map the most biologically-supportable eQTL (Michaelson et al., 2010). Further work led to the development of a method to extract epistatic relationships between loci from information available in the forest structure, all with minimal additional computational cost compared to the basic eQTL mapping with RF. This method and its use are presented in this tutorial.

C.2 Setup

We'll generate some simulated data (genotype and expression) that we can use to demonstrate how interactions can be found in the structure of Random Forests (RF). We will create a variety of traits (purely additive, purely epistatic, and mixed).

```
> set.seed(1337)
> source("functions.R")
> library(rfdev)
> x = simgeno(100)
> x = x[,sort(sample(1:1000,200))] ## keep small for speed
## a purely additive trait
> y1 = simtrait(x[,50]==1,0.5,0.3) + simtrait(x[,75]==1,0.5,0.3)
## a purely epistatic (2-way) trait
> y2 = simtrait(x[,50]==1 & x[,75]==1,,0.3)
## a 3-way epistatic interaction
```

```
> y3 = simtrait(x[,50]==1 & x[,75]==1 & x[,150]==1,,0.2)
## mixed additive and epistatic
> y4 = simtrait(x[,25]==1 & x[,75]==1,,0.2) -
+         simtrait(x[,180]==1,1.8,0.3)
```

C.3 Fitting a Random Forest

Next we will fit an RF to each of the traits we created. We'll do this with a special modified version of the R package `randomForest`, called `rfdev`. This package includes modifications to the regression version of RF to enable some bookkeeping features we will need, such as collecting the OOB (out-of-bag) predictions at each decision node of each tree. The OOB prediction of each node is simply the mean of the OOB values at each decision node in the trees of the forest. This is needed for our demonstration of split asymmetry later on.

```
> rf1 = randomForest(y=y1,x=x,ntree=1000,mtry=50,nodesize=2)
> rf2 = randomForest(y=y2,x=x,ntree=1000,mtry=50,nodesize=2)
> rf3 = randomForest(y=y3,x=x,ntree=1000,mtry=50,nodesize=2)
> rf4 = randomForest(y=y4,x=x,ntree=1000,mtry=50,nodesize=2)
```

C.4 Split asymmetry

The basis of this approach of detecting variable interactions (here in the context of epistasis) is what we call *split asymmetry*. Consider a sequence of two decision splits in a tree, involving two splitting variables, first X_A and then X_B. After splitting on X_B, there will be some difference in means between the values in its left and right daughter nodes (Fig. C.1). We can view this difference in means as a slope. If the mean of the right daughter is greater than that of the left daughter, the slope is positive, and in the opposite case the slope is negative. If there is no dependency between X_A and X_B when considering the response values, we would expect that the slope after splitting on X_B would be the same regardless of whether X_B splits on data in the left or right daughter node of the X_A split. On the other hand, if there is a dependency between X_A and X_B, we expect that the decision at X_A will influence the outcome of the split at X_B, thus resulting in different slopes for X_{Bl} (split on left daughter of X_A) vs. X_{Br} (split on right daughter of X_A). Given this context, we say that a split is *asymmetric* in a sequence of variables with dependencies, and a split is *symmetric* in a sequence of variables with no dependencies.

All such slopes involving all 2-variable decision sequences encountered in the forest are summed according to their "sidedness", leading to two square matrices: M_l for the sequence corresponding to $X_A \rightarrow X_{Bl}$, the "left" matrix, and M_r for the sequence corresponding to $X_A \rightarrow X_{Br}$, the "right" matrix (Fig. C.1). In both matrices, the row indicates the first variable in the decision sequence, and the column indicates the second variable in the sequence. We can use the function `predsymm` to build the M_l and M_r matrices:

```
> ps1 = predsymm(rf1)
> ps2 = predsymm(rf2)
> ps3 = predsymm(rf3)
> ps4 = predsymm(rf4)
```

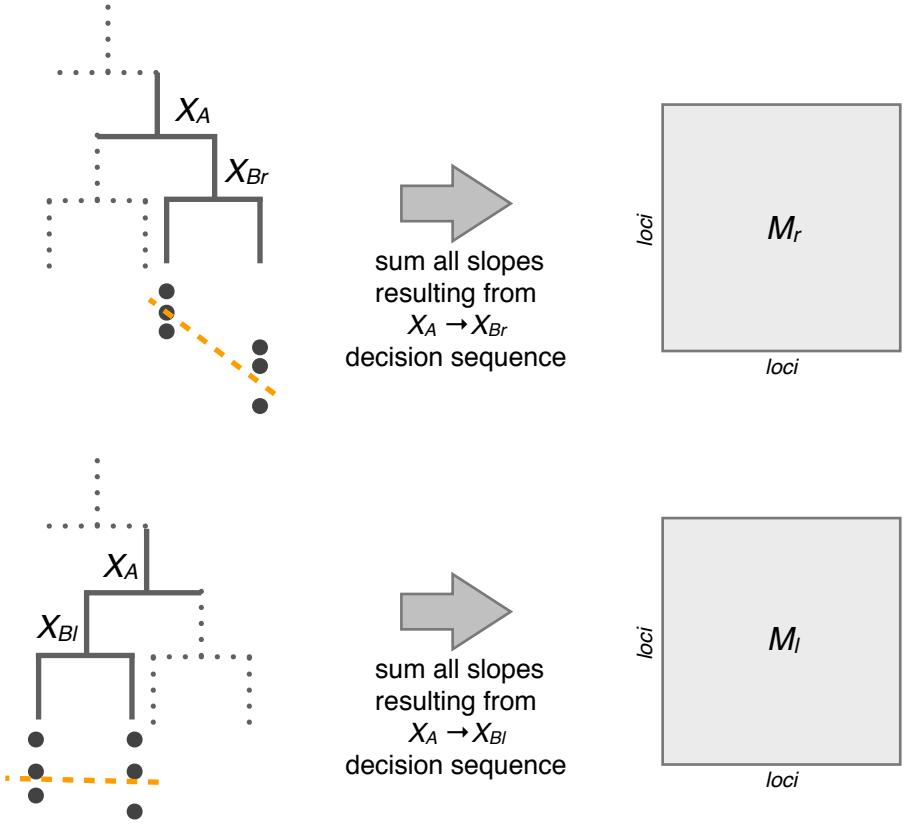


Figure C.1 Searching a forest structure for indications of split asymmetry. In this representation, the decision sequences $X_A \rightarrow X_{Br}$ and $X_A \rightarrow X_{Bl}$ lead to different characteristic slopes, hence the split sequence $X_A \rightarrow X_B$ is *asymmetric*.

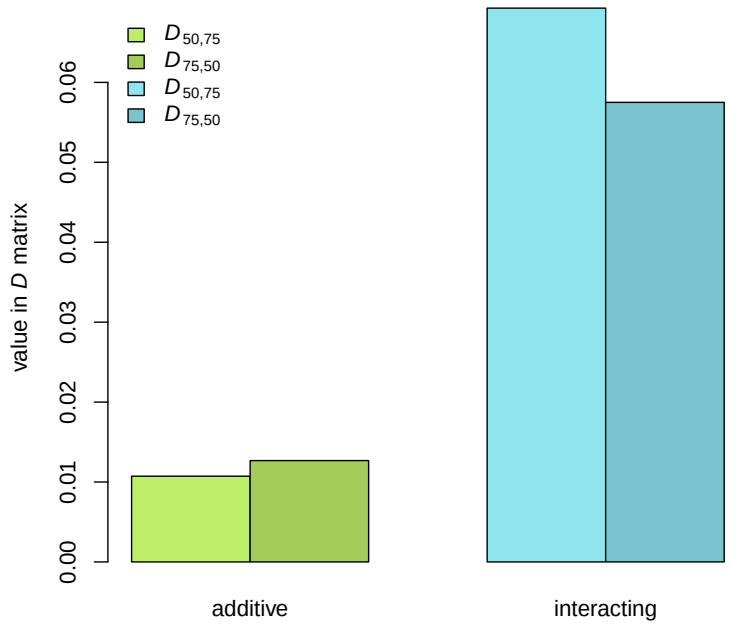


Figure C.2 Comparison of values in D when the two variables in question are additive and independent (left) and interacting and dependent (right).

In cases of extreme dependencies, X_{Br} (for example) might be used frequently, yet X_{Bl} might never be suitable as a splitting variable, and therefore might not occur at all in the forest (leading to an entry of 0 in M_l). In any case, the magnitude of the absolute difference of the aggregated slopes (a matrix D) is an indicator of the strength of the dependency between the involved splitting variables. Note: while $|M|$ traditionally denotes the determinant of M when M is a matrix, for convenience we use it here to mean the matrix resulting from taking the absolute values of the entries in M.

$$D = |M_r - M_l| \tag{C.1}$$

The simulated traits `y1` and `y2` involve the same causal "loci" – the 50^{th} and 75^{th} markers (we'll call them X_{50} and X_{75} here). However, the models generating the traits are quite different, with `y1` being a linear combination of X_{50} and X_{75}, and `y2` being the result of an interaction of the two. From our description of the matrix D, we expect that $D_{50,75}$ (also $D_{75,50}$) should have higher values when X_{50} and X_{75} interact (as is the case for `y2`) than if they are additive and independent (as is the case for `y1`). This is indeed the case, since an interaction leads to large split asymmetry, whereas additivity will not (Fig. C.2).

We'll now introduce a few modifications to D to further refine the score it represents. First, we'll subtract as a penalty the mean absolute slope, here the matrix S:

$$S = \frac{|M_r| + |M_l|}{2} \tag{C.2}$$

$$D' = D - S \tag{C.3}$$

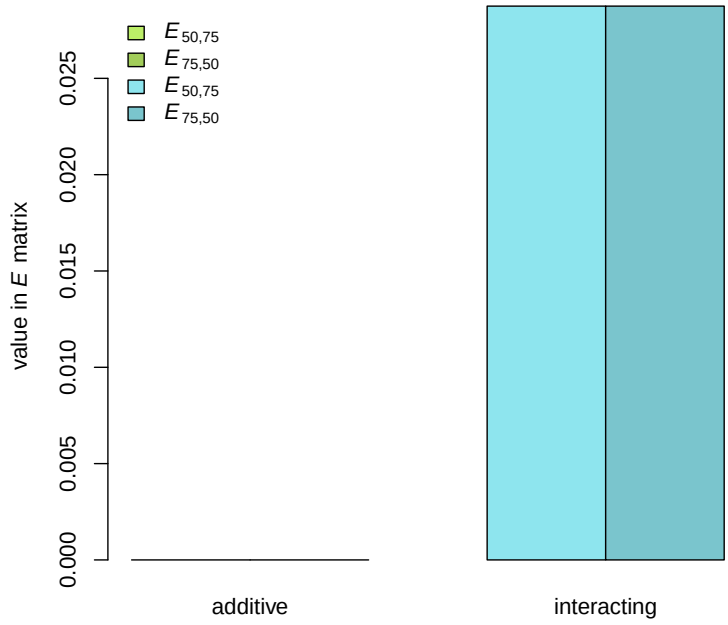


Figure C.3 Comparison of values in E when the two variables in question are additive and independent (left) and interacting and dependent (right). The operations performed via the `EPI` function mitigate false positives in comparison to the D matrix (Fig. C.2).

We will additionally constrain negative values to be 0, so that only pairs whose difference in slope exceeds the average magnitude are considered.

$$D''_{ij} = \begin{cases} 0 & \text{for } D'_{ij} \leq 0 \\ D'_{ij} & \text{for } D'_{ij} > 0 \end{cases} \tag{C.4}$$

Finally, we will take the minimum of the corresponding values D''_{ij} and D''_{ji}, since purely interacting variables (i.e. without an additive effect), will be "order-agnostic", meaning that the sequences $X_A \rightarrow X_B$ and $X_B \rightarrow X_A$ should both be asymmetric; we take the minimum of the two scenarios to be conservative. In practice, this has reduced the number of false positives encountered. This final epistasis score is stored in a (symmetric) matrix E:

$$E_{ij} = \min\{D''_{ij}, D''_{ji}\} \tag{C.5}$$

These operations (equations C.1-C.5) have been combined into a single function, `EPI`, for convenience. This function takes the output of `predsymm` as its argument, and returns the matrix corresponding to E as described above. Let's look at how E compares to the original D when we look at the additive vs. interacting scenarios (Fig. C.3).

In Figure C.3 we can see that any indication of an interaction in the additive scenario vanishes (left), while the indication of an interaction in the epistatic scenario remains (right). Additionally, the values for $E_{50,75}$ and $E_{75,50}$ are the same (i.e. the matrix is symmetric).

C.5 Scoring interactions

In this section we will look at the ability of RF split asymmetry to recover the known models that we embedded in the traits.

C.5.1 Significance

Before we score each trait looking for epistasis, we'll want to have appropriate null distributions so that we can calculate *P* values for the interactions.

```
## null distributions for each trait
> rfn1 = randomForest(y=sample(y1),x=x,ntree=1000,mtry=50,nodesize=2)
> psn1 = predsymm(rfn1)
> N1 = EPI(psn1)
> rfn2 = randomForest(y=sample(y2),x=x,ntree=1000,mtry=50,nodesize=2)
> psn2 = predsymm(rfn2)
> N2 = EPI(psn2)
> rfn3 = randomForest(y=sample(y3),x=x,ntree=1000,mtry=50,nodesize=2)
> psn3 = predsymm(rfn3)
> N3 = EPI(psn3)
> rfn4 = randomForest(y=sample(y4),x=x,ntree=1000,mtry=50,nodesize=2)
> psn4 = predsymm(rfn4)
> N4 = EPI(psn4)
## pool the null values
> n = c(N1[N1>0],N2[N2>0],N3[N3>0],N4[N4>0])
```

The null distribution closely resembles a beta density (Fig. C.4). We can fit parameters for such a density so that we have a parametric null distribution rather than an empirical one, which would lead to inaccurate *P* values for extreme values in *E*.

```
## fit the parameters of a beta distribution
> b.obj = function(p,x){
+  e = dbeta(x,p[1],p[2])
+  -sum(log(e))
+ }
## parameter guesses
> p = c(1,10000)
## optimize
> est = nlm(b.obj,p,n)
## parameters for parametric null dist
> est = est$est
```

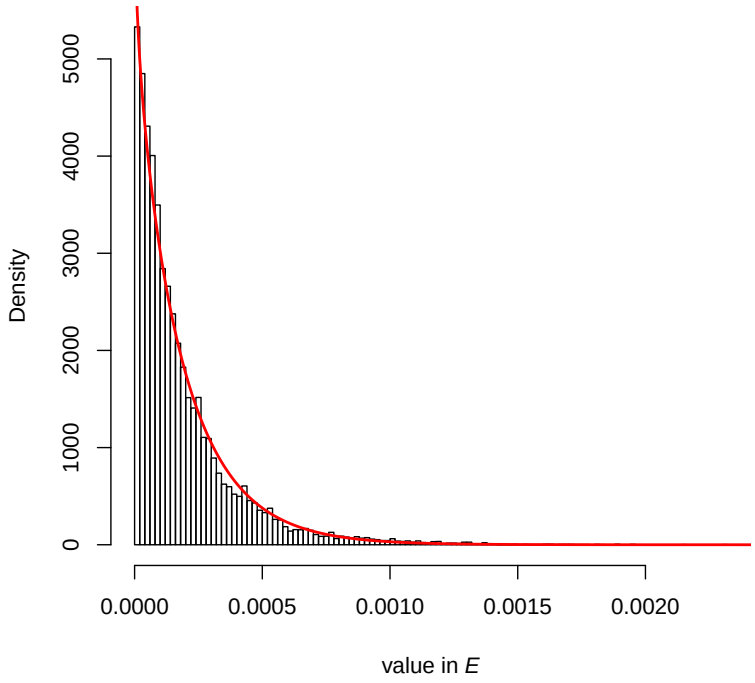


Figure C.4 Empirical null distribution for E, with fitted beta density (red line).

C.5.2 Results

With the parametric null distribution in hand, we can compute P values and look for cases of significant epistasis in each of the four Random Forests we fit. We will take those relationships that have $FDR < 0.01$ to be examples of epistasis.

```
> E1 = EPI(ps1)
> E2 = EPI(ps2)
> E3 = EPI(ps3)
> E4 = EPI(ps4)
> P1 = ifelse(E1>0,1-pbeta(E1,est[1],est[2]),1)
> P2 = ifelse(E2>0,1-pbeta(E2,est[1],est[2]),1)
> P3 = ifelse(E3>0,1-pbeta(E3,est[1],est[2]),1)
> P4 = ifelse(E4>0,1-pbeta(E4,est[1],est[2]),1)
```

Figure C.5 shows that in our example traits, epistatic relationships were recovered as they were simulated. Additionally, in these examples no false positive epistatic relationships were found.

In this tutorial, we've demonstrated that RF split asymmetry can be an effective means for pinpointing epistatic relationships between loci. The needed information is contained in the forest structure, and only minimal additional effort (beyond fitting the forest) is required to extract these relationships.

```
> par(mfrow=c(2,2))
> plotEpi(rf1,P1,,"additive")
> plotEpi(rf2,P2,,"2-way epistatic")
> plotEpi(rf3,P3,,"3-way epistatic")
> plotEpi(rf4,P4,,"additive and 2-way epistatic")
```

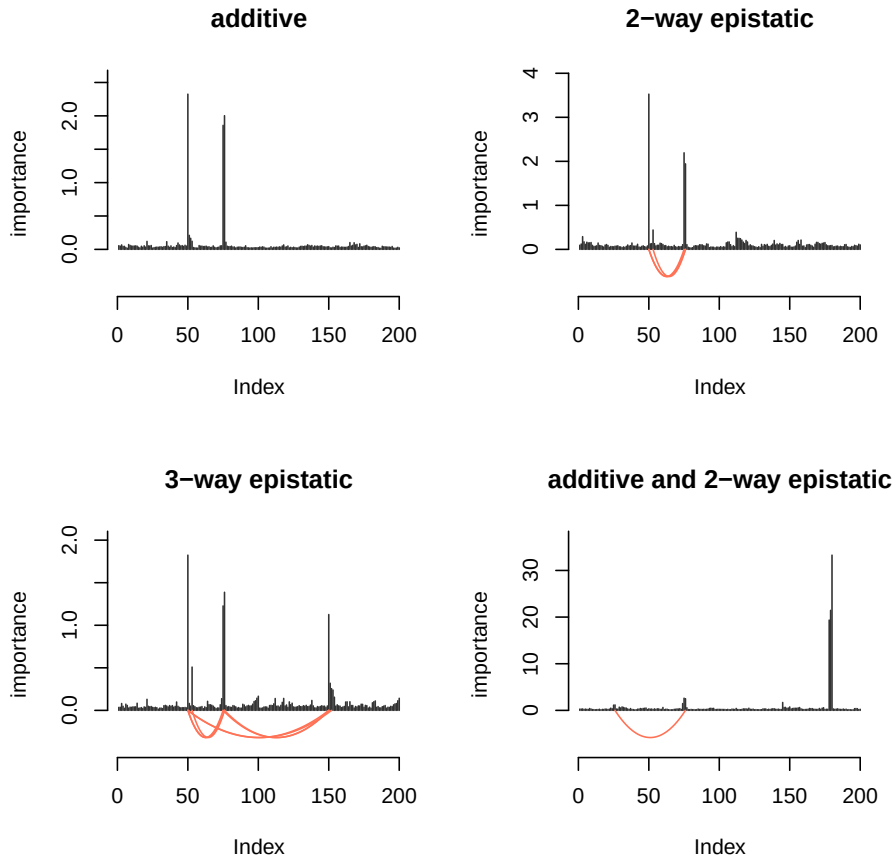


Figure C.5 eQTL profiles for the four simulated traits are shown here, with epistatic relationships recovered via RF split asymmetry shown in orange.

Appendix D

Tutorial: Using the RF proximity measure

D.1 Introduction

Clustering algorithms rely on a distance measure to indicate the (dis)similarity of objects to be clustered. There are many such measures, such as the familiar Pearson correlation, the Euclidean distance, and the Manhattan distance, among many others. Another such measure, produced during the construction of a Random Forest (RF), is the RF proximity measure. Briefly, this is a measure of how frequently two observations end up in the same terminal node, throughout the decision trees of the forest. A high proximity value indicates that observations take the same decision path to their ultimate classifications. This measure provides implicit weighting of features, since only features with discriminatory power are used repeatedly in the RF. For further treatment of the topic, see Shi and Horvath (Shi and Horvath, 2006), who present a case study thoroughly describing the use of RF proximity for unsupervised clustering. In addition, Xiao and Segal (Xiao and Segal, 2009) present the use of the proximity measure to detect regulatory programs in yeast.

D.2 Setup

In this tutorial we will look at some basic examples of using the RF proximity measure as a basis for clustering time course gene expression data, and compare it to other distance measures such as the Euclidean and Manhattan distances.

First, we will synthesize some data that reflects some of the characteristics of time course gene expression data. We will include elements of periodicity in the expression, segments of correlation and divergence within the synthetic clusters, and noise.

```
> library(cluster)
> library(randomForest)
> ## a function to simulate periodic data
> ff = function(n=100,x=0,y=10){
+  sin(seq(0,runif(1,min(x,y),max(x,y)),length.out=n))
+  }
> set.seed(114399)
> W1 = matrix(runif(400,0.5,3),400,120) ## vary amplitude
```

```
> W2 = matrix(rnorm(400*120,0,1),400,120) ## noise
> M = matrix(0,400,120)
> ## true cluster labels
> truth = rep(1:4,each=100)
> ## first three clusters
> M[1:100,21:120] = t(replicate(100,ff(100,4.9*pi,5.1*pi)))
> M[101:200,21:120] = t(replicate(100,ff(100,5.9*pi,6.1*pi)))
> M[201:300,21:120] = t(replicate(100,ff(100,3.9*pi,4.1*pi)))
> ## a more difficult cluster...
> M[301:400,21:120] = t(replicate(100,ff(100,pi,7*pi)))
> ## ...that has a common initial phase
> M[301:400,1:20] = t(replicate(100,ff(20,0.9*pi,pi)))
> ## diverging initial phases for other clusters
> M[1:300,1:20] = t(replicate(300,ff(20,pi,3*pi)))
> ## incorporate the noise elements
> M = (M*W1)+W2
```

Now let's look at the results (Fig. D.1):

```
> cc = colorRampPalette(c(grey(0.1),grey(0.95)))
> img(M,col=cc(256),ylab="genes",xlab="time")
> abline(h=c(100,200,300),lwd=2.5,lty=2,col='firebrick')
```

D.3 Comparing performance of distance measures

Here we will compare the ability of several distance measures in recovering the known clusters we embedded in the data. We will look at the Euclidean and Manhattan distances, as well as unsupervised and supervised versions of the RF proximity measure.

```
> ## try the Euclidean distance
> d1 = dist(M)
> ## try the Manhattan distance
> d2 = dist(M,'manhattan')
> ## try the (unsupervised) RF proximity
> rf = randomForest(M,proximity=T,ntree=1000)
> d3 = as.dist(sqrt(1-rf$prox))
```

For the last test we'll perform supervised learning with RF. This can be helpful if we have some external information (for instance from experiments) about the genes that we could incorporate as training labels. In this example, we will simply select a handful of the genes that we know to be "in" cluster 4 and some that we know to be "out" of cluster 4, then train an RF classifier on these examples. We'll then use the classifier to predict on all genes, telling us which genes are likely to be "in" cluster 4 and which are not. We can extract the proximity during the prediction step, as shown here.

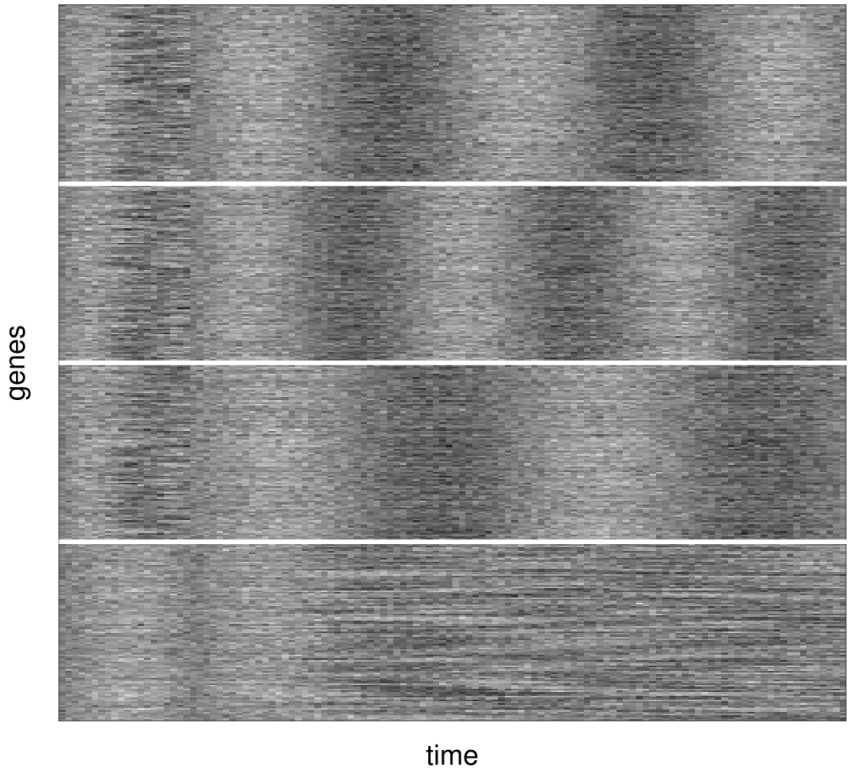


Figure D.1 The synthesized data, arranged by cluster.

```
> ind = c(sample(1:300,100),sample(301:400,33)) ## training set
> X = M[ind,]
> y = as.factor(c(rep("out",100),rep("in",33)))
> rf = randomForest(y=y,x=X,sampsize=c(25,25),replace=T,strata=y,ntree=2000,
+         importance=T,mtry=3)
> prd = predict(rf,M,proximity=T)
> d4 = as.dist(sqrt(1-prd$prox))
```

Let's check to see to what degree each distance measure can recapitulate the actual cluster assignments:

```
> ## real cluster labels
> actual = matrix(0,400,400)
> actual[1:100,1:100] = 1
> actual[101:200,101:200] = 2
> actual[201:300,201:300] = 3
> actual[301:400,301:400] = 4
> cl = pam(d1,4)
> m1 = outer(cl$clust,cl$clust,'==')
> cl = pam(d2,4)
> m2 = outer(cl$clust,cl$clust,'==')
> cl = pam(d3,4)
> m3 = outer(cl$clust,cl$clust,'==')
> cl = pam(d4,4)
> m4 = outer(cl$clust,cl$clust,'==')
```

Now we can visualize these results to help us get a feeling for which distance measure gives the "cleanest" and most accurate clustering (Fig. D.2).

```
> par(mfrow=c(2,2),mar=c(2,2,2,2)+0.1)
> img(m1,col=grey.colors(2))
> mtext("Euclidean",1,0.5)
> img(m2,col=grey.colors(2))
> mtext("Manhattan",1,0.5)
> img(m3,col=grey.colors(2))
> mtext("RF unsupervised",1,0.5)
> img(m4,col=grey.colors(2))
> mtext("RF supervised",1,0.5)
```

We can also look at the proportions of recovered true cluster assignments as a barplot (Fig. D.3):

```
> x = matrix(0,4,4)
> rownames(x) = c("Euclidean","Manhattan",
+         "RF unsupervised","RF supervised")
> colnames(x) = 1:4
```

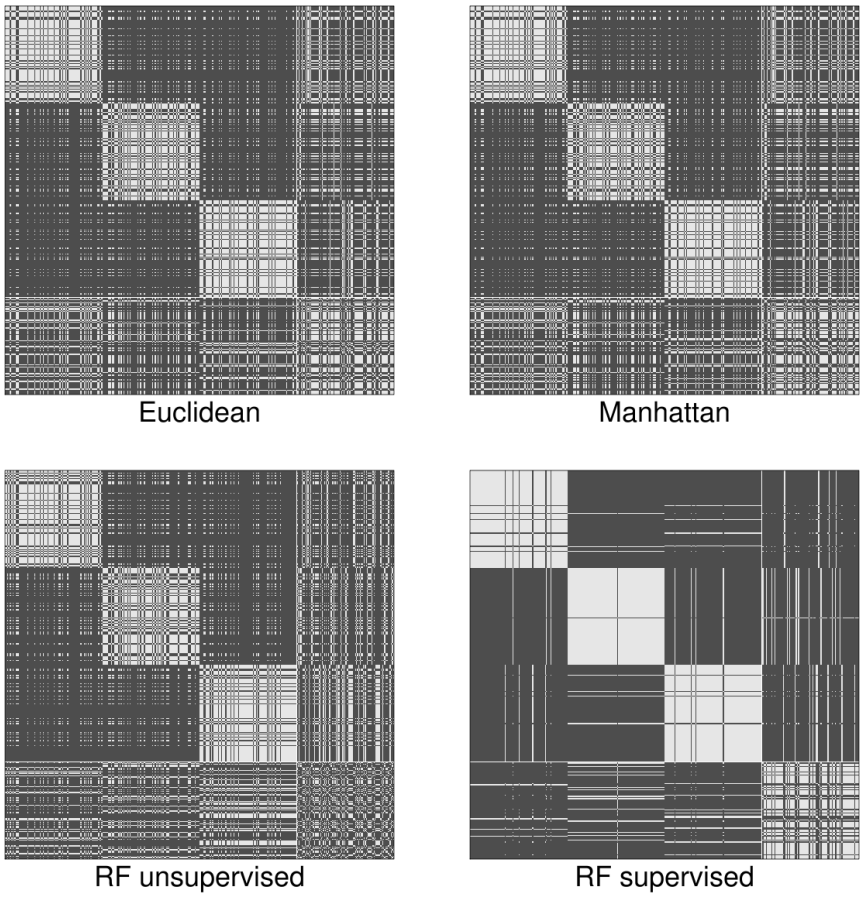


Figure D.2 Ability of each distance measure to recover true cluster assignments. A matrix entry has a lighter gray if the corresponding row and column have the same cluster assignment.

```
> for(i in 1:4) x[1,i] = sum(m1[actual==i])/1e4
> for(i in 1:4) x[2,i] = sum(m2[actual==i])/1e4
> for(i in 1:4) x[3,i] = sum(m3[actual==i])/1e4
> for(i in 1:4) x[4,i] = sum(m4[actual==i])/1e4

> cc2 = c("olivedrab4","olivedrab3","paleturquoise4","paleturquoise3")
> barplot(x,beside=T,border=NA,col=cc2,legend=T,
+         args.legend=list(x="topright",bty='n'),
+         xlab="cluster",
+         ylab="proportion recovered",
+         ylim=c(0,1.2))
```

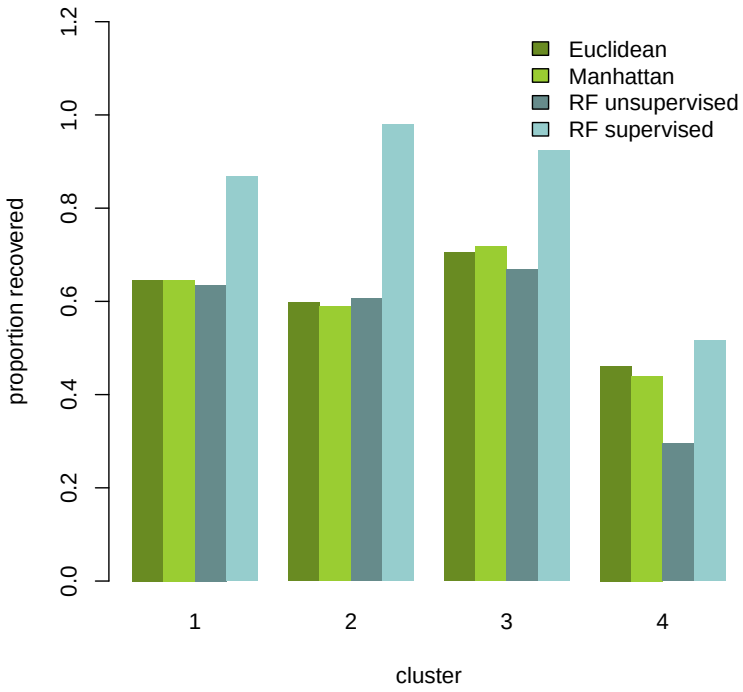


Figure D.3 Proportions of cluster members correctly identified when clustering with each distance measure.

The Euclidean, Manhattan and unsupervised RF proximity all perform comparably. The clusters can be resolved much more clearly when some additional information is available that can turn the unsupervised task into a supervised one. Note that we did not need information about the identity of each of the four clusters to accomplish this. We only had a small sample from cluster 4, and a sample of the others were labeled "not 4", leading to a two-class learning problem. This information was enough to lead to an RF proximity that clearly performs better at recovering the true clustering identities.

D.4 PAM Clustering

Here we'll demonstrate some basic steps that might be taken in a typical cluster analysis with the RF proximity measure.

First, if we don't know *a priori* how many clusters to look for, we need some justification for our choice of k, the number of clusters in the data. Here we'll test a range of k and evaluate the silhouette for each value. In addition, we'll perform a Kolmogorov-Smirnov (KS) test, comparing the distribution of silhouette values at each value of k against that of the previous value of k, looking for significant increases. Since we're comparing against previous values of k, our first test takes place at $k = 3$. The results are shown in Figure D.4.

```
> K = sapply(2:10,function(k) silhouette(pam(d4,k))[,3])
> colnames(K) = 2:10
> Pk = sapply(1:8,function(i) ks.test(K[,i+1],K[,i],alternative='l')$p.v)

> layout(matrix(c(1,2),2,1),heights=c(2,1))
> par(mar=c(5,5,4,2)+0.1)
> boxplot(as.data.frame(K),xlab='k',ylab='silhouette',col='olivedrab3')
> par(mar=c(5,5,0,2)+0.1)
> barplot(c(NA,-log10(Pk)),beside=T,ylab="-log10 P \n (KS test)")
```

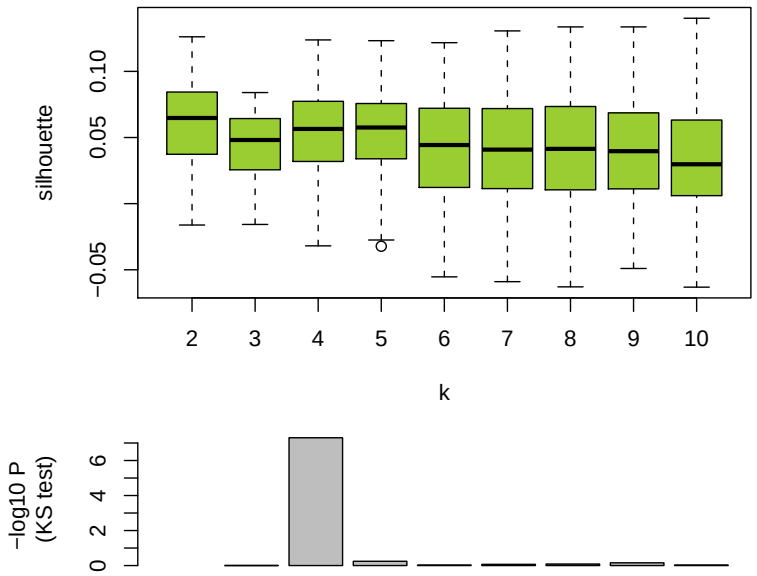


Figure D.4 Finding a suitable value of k. The distribution of the silhouette values for each value of k is shown (top), along with a corresponding P value (bottom) indicating whether the distribution represents a significant improvement over the predecessor.

Our intuition from the boxplot is confirmed by performing the KS test: 4 clusters is a significant improvement over 3 clusters, and 5 clusters is no better than 4. Therefore, a k of 4 is optimal (as we would hope to see). Notice that the silhouette distribution at $k = 2$ is noticably higher than those at other values of k. This is likely an artifact that can be traced back to the fact that this distance is derived from a 2-class classification; these clusters likely represent these two classes, and not the full extent of the structure in the data.

Now we can perform the actual clustering using PAM, which is described in detail in (Kaufman and Rousseeuw, 1990).

```
> cl = pam(d4,4)
```

We can use these cluster labels as a response variable when training a Random Forest with the original data. By using an outlier measure derived from the RF proximity matrix, we can get an estimate of the confidence in the predicted cluster label.

```
> y = as.factor(cl$clust)
> rf = randomForest(y=y,x=M,ntree=1000,importance=T,proximity=T)
```

Before we look at the observed outlier measures, it will be helpful to get an idea of what we can expect when the null hypothesis is true (no meaning of cluster assignments).

```
> yn = sample(y)
> rfn = randomForest(y=yn,x=M,ntree=1000,importance=T,proximity=T)
> cutoff = max(outlier(rfn))
> cutoff

[1] 2.029517
```

Let's take a look at the results (Fig. D.5):

```
> ol = outlier(rf)
> plot(ol,col=c("darkorange","lightskyblue4","olivedrab","firebrick")[y],
+         pch=16,ylab="outlier measure")
> abline(h=cutoff,col='grey',lty=2,lwd=2)
```

We'll drop genes that have an outlier measure that is greater than what we observed under the null hypothesis.

```
> yhat = rf$predict
> yhat[yhat=="1" & ol > cutoff] = NA
> yhat[yhat=="2" & ol > cutoff] = NA
> yhat[yhat=="3" & ol > cutoff] = NA
> yhat[yhat=="4" & ol > cutoff] = NA
```

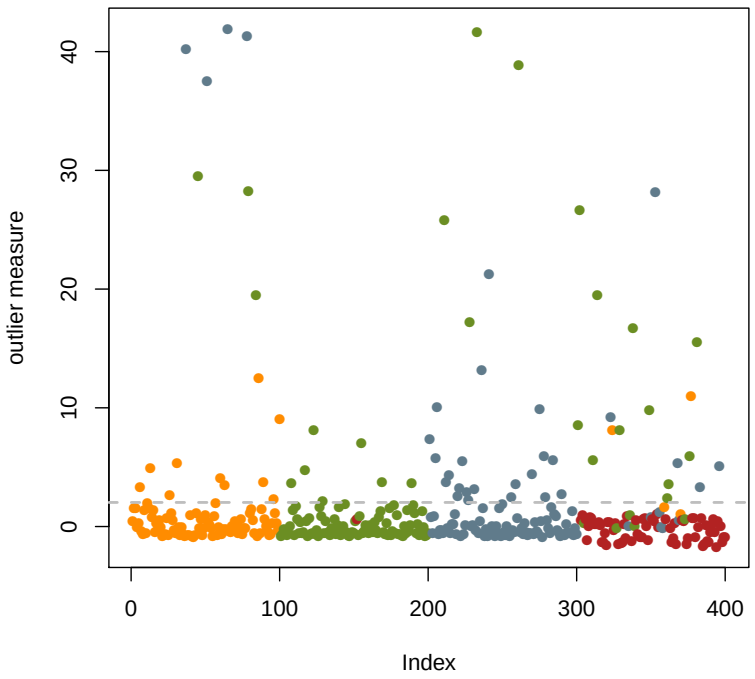


Figure D.5 The outlier measure. Each point is colored according to the cluster to which it was predicted to belong. The maximum outlier value observed under the null hypothesis is shown with a horizontal line.

Now let's look at the predicted clustering and overlay some information about which features were important for defining each class (Fig. D.6).

```
> these = which(!is.na(yhat))
> M1 = M[these,]
> img(M1[order(yhat[these]),],col=cc(256),
+         ylab="genes",xlab="time",cex.lab=3)
> imp = apply(rf$importance,2,function(x) x/max(x))[,1:4]
> ct = table(yhat)
> imp = matrix(c(rep(imp[,1],ct[1]),rep(imp[,2],ct[2]),
+         rep(imp[,3],ct[3]),rep(imp[,4],ct[4])),
+         length(yhat),120,byrow=T)
> pal = rgb(1,0.55,0,seq(0,0.5,length.out=100))
> img(imp,col=pal,add=T)
> abline(h=cumsum(ct)[-4],lwd=8,lty=1,col='white')
```

This figure shows us that the RF clustering recapitulates the original grouping fairly well. In addition, the overlaid color gives us an idea about the identifying features of each cluster.

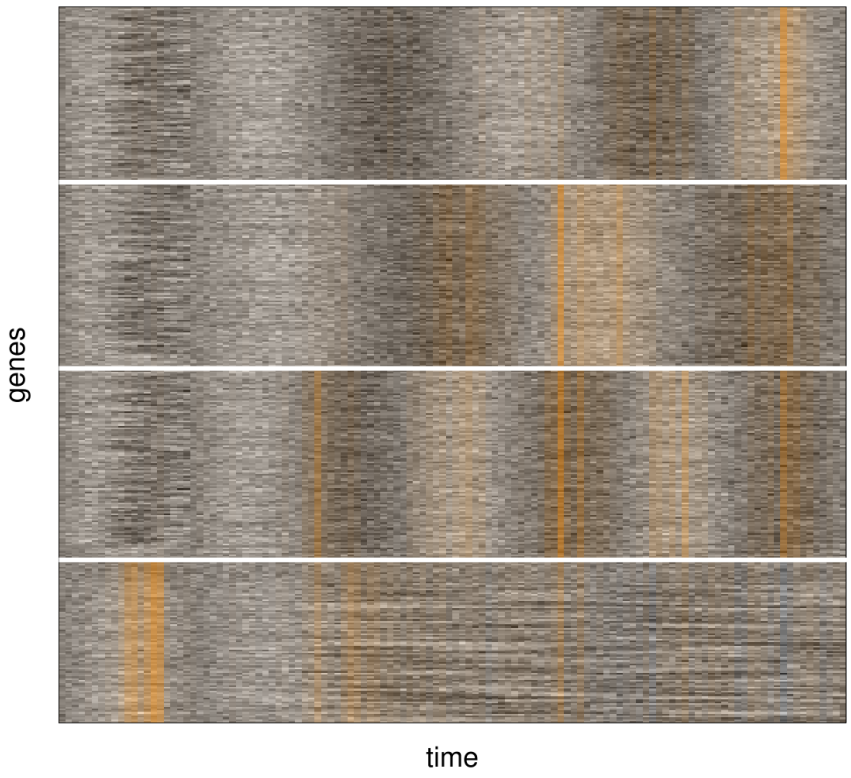


Figure D.6 Importance of features for cluster identity. Features are colored according to their contribution to a cluster's identity.

Appendix E

Tutorial: Finding trait-specific transcriptional hierarchies

E.1 Introduction

One common aim of eQTL studies is to find regulatory relationships of genes that underlie disease (Chen et al., 2008; La Merrill et al., 2010). Often eQTL data are used to derive a network, and the roles of genes are inferred by their placement within this network. Genes with well-defined roles are then implicated in the etiology of the disease. This approach will work well when the role of the gene is defined by multiple regulatory relationships of high significance. However, a network approach is less effective when the regulatory relationships are subtle and do not individually meet the threshold requirement for the creation of edges in the network.

We have developed an approach that elucidates regulatory roles of genes using distributional properties of eQTL data. This method does not require the thresholding of the eQTL scores, and is sensitive to small eQTL score enrichments that cannot be effectively accounted for in a network-centric approach to eQTL analysis. In this tutorial, we demonstrate the use of this method on simulated data and compare it to a graph-theoretic approach.

E.2 Setup

First we'll set up a matrix of synthetic data that imitates eQTL data – we'll draw values randomly from a beta density.

```
> source("functions.R")
> set.seed(1492)
> M = matrix(rbeta(1e7,1,1e3),1e4,1e3)
> rownames(M) = paste("G",1:10000,sep="")
> colnames(M) = paste("M",1:1000,sep="")
```

Next we'll take the first 100 rows and columns and give them values that tend to be *slightly* higher than the background. We do this in a particular way such that 5 genes are "upstream" of the other 95 (i.e. these columns have enriched scores), 5 genes are "downstream" of the other 95 (i.e. these rows have enriched

scores), and 5 genes are "central" to the other 95 (i.e. both these rows and corresponding columns are enriched compared to background).

```
> up = sample(1:100,5)
> dwn = sample((1:100)[-up],5)
> ctr = sample((1:100)[-c(up,dwn)],5)
> M[1:100,up] = rbeta(500,1.5,1e3)
> M[dwn,1:100] = rbeta(500,1.5,1e3)
> M[1:100,ctr] = rbeta(500,1.5,1e3)
> M[ctr,1:100] = rbeta(500,1.5,1e3)
```

This set of "spike in" data imitates what we would want to find among a set of functionally related genes (e.g. the schizophrenia genes as presented in Chapter 4), that is, in terms of transcription, some genes tend to be regulators, some tend to be targets, and some tend to be fixed somewhere in the middle of the hierarchy.

Knowing the roles that these genes play in transcriptional regulation can give us crucial insight into how the genes affect the disease or process that ties them together. This insight can help to focus research in the right areas, leading to more accurate diagnosis and more effective treatment.

E.3 Graph-theoretic approach

Perhaps the most obvious approach to finding the regulatory roles of our genes of interest would be to construct a directed graph from the eQTL data, and then look at the placement of the genes within the graph for insight about their roles. Since the eQTL data are dense, we need to sparsify the data (i.e. apply a threshold) before the graph structure can become helpful.

In our toy example, we know that the background distribution is a beta density with parameters 1 and 1000. Let's use this knowledge to get *P* values for our subgraph of interest:

```
> library(graph)
> G = M[1:100,1:100]
> G = 1-pbeta(G,1,1e3)
```

How many edges have an $FDR < 0.05$?

```
> sum(p.adjust(G,'fdr')<0.05)

[1] 0
```

So at this point we could say that the graph-theoretic approach failed to help us because we don't have *any* edges we can use reliably – they all look like values we would expect from the null distribution. For the sake of the tutorial we'll proceed a bit further, but would exercise caution if we were working with real data. Let's just use $P < 0.05$ as a threshold for the edges, then construct the graph object and get the node degrees.

```
> G = G<0.05
> colnames(G)=1:100
```

```
> rownames(G)=1:100
> G = as(t(G),'graphAM')
> dg = degree(G)
```

Probably the most obvious metric to use when inquiring about the regulatory role of a gene (node) in a directed graph is the (in and out) degree. We would hope that our upstream spike-ins would have systematically higher out degrees than in degrees. Likewise we would want our downstream spike-ins to have higher in degrees than out degrees. We would want our central spike-ins to have high total degrees. Let's look at how this works out with the graph approach (Fig. E.1).

```
> par(mfrow=c(3,1))
> plot(1:100,dg$out-dg$inD,col=ifelse(1:100%in%up,'red','black'),
+         ylab="outD-inD",xlab='gene')
> plot(1:100,dg$inD-dg$out,col=ifelse(1:100%in%dwn,'red','black'),
+         ylab="inD-outD",xlab='gene')
> plot(1:100,dg$inD+dg$out,col=ifelse(1:100%in%ctr,'red','black'),
+         ylab="outD+inD",xlab='gene')
```

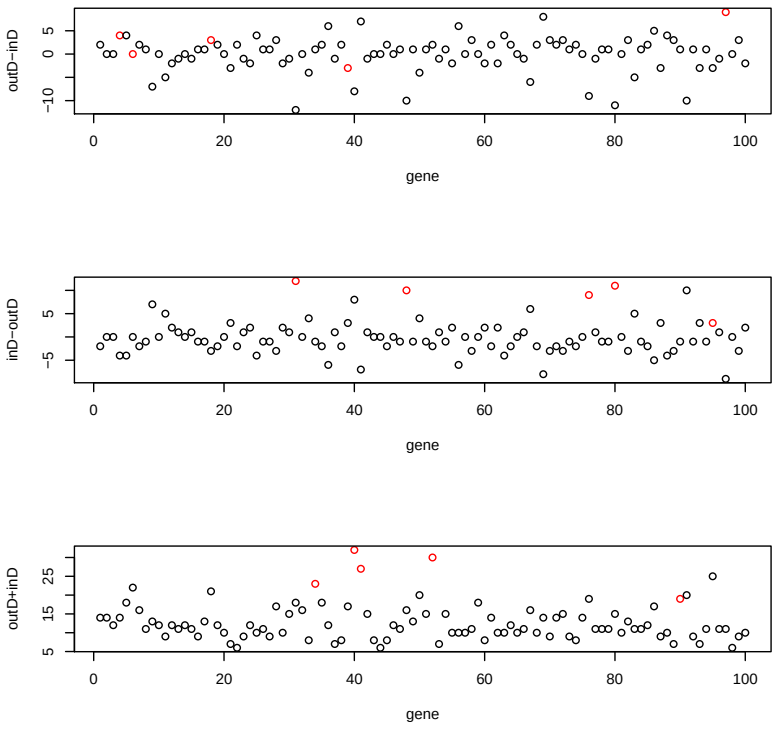


Figure E.1 Ability of degree measures to properly prioritize the spike-ins (red dots). The upstream spike-ins are not well-recovered (top), while the downstream (middle) and central (bottom) spike-ins are recovered with minimal false-positives.

The upstream spike-ins show very little departure from the genes who were not given any regulatory role. This would make it very difficult to infer master regulators using this approach. Downstream and central spike-ins are somewhat easier to discern from the background.

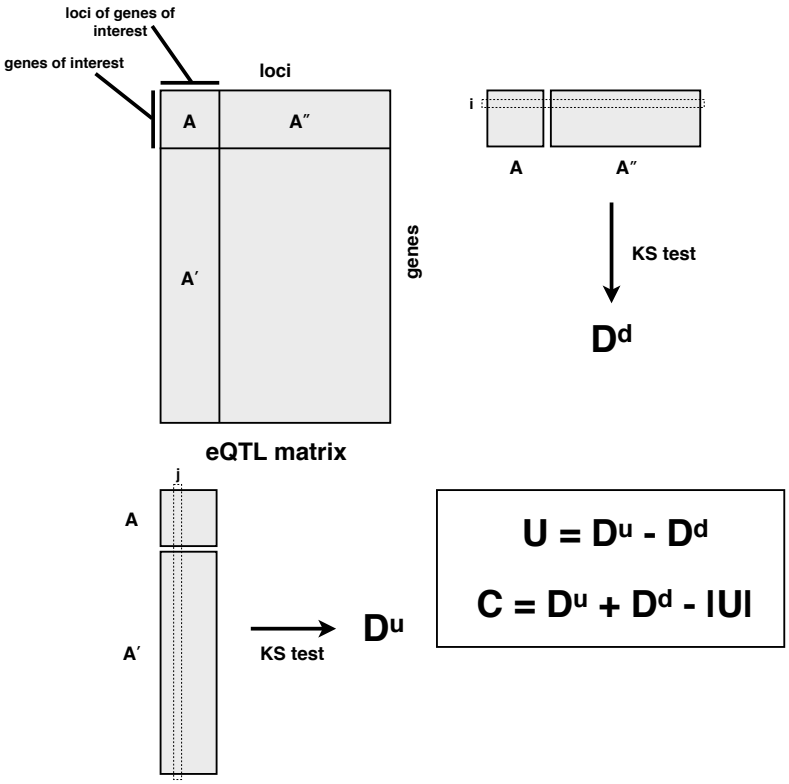


Figure E.2 Using eQTL data and the KS test to derive regulatory upstreamness (U) and centrality (C).

E.4 Distributional approach

The graph-theoretic approach had difficulties recovering the regulatory roles of our spike-ins. Here we'll present an alternative approach that looks at distributional differences, rather than thresholded binary values, to infer regulatory roles.

The question we are asking in this case is nuanced: we are interested in the *role* of a gene *with respect to functionally related genes*. This implies a few things. First, we are not interested in *specific* connections to other genes, but rather tendencies over a number of genes. Thus, we might be better off examining distributional properties than specific values (as we did in the graph approach). Second, we are focusing on roles *within* a subset of all genes, meaning that we can compare a role among functionally related genes to a role among non-related genes. Since the role should be specific to a group of genes, we can use all genes *not* belonging to that group as a reference distribution.

In this context, the ideas behind the Kolmogorov-Smirnov test become quite useful. First let's divide up the eQTL matrix into several smaller matrices (Fig. E.2). The matrix A is defined by our genes of interest (rows) and their genetic loci, here represented by the closest genetic marker (columns), thus, A_{ij} represents the effect of the j^{th} locus on the transcription of the i^{th} gene. The matrix A' has the same columns as A, but as rows has all genes not found in A. Likewise, A'' has the same rows as A, but as columns has all genetic loci (i.e. markers) not found in A. We first define a statistic, D^u, that represents the tendency of a genetic locus to be an upstream regulator of genes in our group of interest:

$$D_j^u = D_{A'_j,A_j} = \sup_x \{F_{A'_j}(x) - F_{A_j}(x)\} \tag{E.1}$$

Where $F_{A'_j}(x)$ is the empirical cumulative distribution function of the values in the j^{th} column of A', and $F_{A_j}(x)$ is the empirical cumulative distribution function of the values in the j^{th} column of A. If D_j^u is large, it suggests that the locus corresponding to j (and by extension, the gene at that locus) is upstream of the genes defining A, but not the genes defining A' (genes not belonging to our group of interest). We note here that if multiple genes map to a locus (marker), each of the genes is assigned the corresponding value of D_j^u.

Next, we define a statistic D^d, representing the tendency of a gene to be downstream of genes in our group of interest:

$$D_i^d = D_{A''_i,A_i} = \sup_x \{F_{A''_i}(x) - F_{A_i}(x)\} \tag{E.2}$$

Where $F_{A''_i}(x)$ is the empirical cumulative distribution function of the values in the i^{th} row of A'', and $F_{A_i}(x)$ is the empirical cumulative distribution function of the values in the i^{th} row of A. If D_i^d is large, it suggests that the gene corresponding to i tends to be downstream of the loci defining A, but not the loci defining A'' (genes not belonging to our group of interest).

From these two statistics, we define "upstreamness", which will be positive for regulators, negative for targets, and close to zero for less well-defined genes.

$$\text{upstreamness}_i = D_j^u - D_i^d \tag{E.3}$$

In this case, the subscript j corresponds to the locus containing the gene i.

If we have a gene that has substantial values for both D_j^u and D_i^d, upstreamness will be close to zero. Nevertheless, we would like to capture this as an interesting gene (since this is the scenario that fits our "central" spike-ins). We define centrality to be the sum of D^u and D^d, with the absolute value of upstreamness subtracted as a penalty.

$$\text{centrality}_i = D_j^u + D_i^d - |\text{upstreamness}_i| \tag{E.4}$$

Again, the subscript j corresponds to the locus containing the gene i.

In practice, we use the `ks.test` function in R to acquire D^u and D^d. Upstreamness and centrality are then quite straightforward to compute. All of this is wrapped in the function `ucScores`, which takes as arguments `eqtl` (the named eQTL matrix), `genes` (a named logical vector indicating which genes are in the group of interest), `markers` (a named logical vector indicating which loci correspond to the genes of interest), and `cis.map` (a character vector with gene names as names and marker names as entries in the vector, indicating the mapping from genes to markers). The `ucScores` function has a logical switch, `nulldist`, that, when true, calculates upstreamness and centrality scores under the null hypothesis (genes not functionally related, but rather sampled randomly from the data).

We can plot the upstreamness and centrality values for our data, and then overlay the density of values under the null hypothesis. Values that lie outside of this density are unlikely to occur by chance, and so represent genes with significant positions in the regulatory hierarchy of our genes of interest (Fig. E.3).

```
## cis.map
> cm = structure(sample(colnames(M),1e4,replace=T),names=rownames(M))
> cm[1:100] = colnames(M)[1:100]
```

```
## genes
> genes = structure(logical(1e4),names=rownames(M))
> genes[1:100] = TRUE
## markers
> markers = structure(logical(1e3),names=colnames(M))
> markers[1:100] = TRUE
## actual values
> uc = ucScores(M,genes,markers,cm)
## 20x null distribution
> null = lapply(1:20,function(i) ucScores(M,genes,markers,cm,TRUE))
> x = do.call('c',lapply(null,'[[',2))
> y = do.call('c',lapply(null,'[[',1))
## significance and FDR
> P = p2d(uc[[2]],uc[[1]],x,y,n=100)
> these = p.adjust(P,'fdr')<0.05

## plot the results
> plot(uc[[2]],uc[[1]],pch=16,col='grey',cex=0.75,ylab="upstreamness",
+         xlab="centrality")
> points(uc[[2]][these],uc[[1]][these],col='black',pch=1,cex=1.6)
> points(uc[[2]][up],uc[[1]][up],col='red2',pch=16,cex=1.1)
> points(uc[[2]][dwn],uc[[1]][dwn],col='steelblue',pch=16,cex=1.1)
> points(uc[[2]][ctr],uc[[1]][ctr],col='gold2',pch=16,cex=1.1)
> lv = q2d(x,y,c(0.05,seq(0.1,0.9,0.1)))
> contour(kde2d(x,y,n=100),levels=lv,labels=names(lv),add=T,lwd=0.75)
```

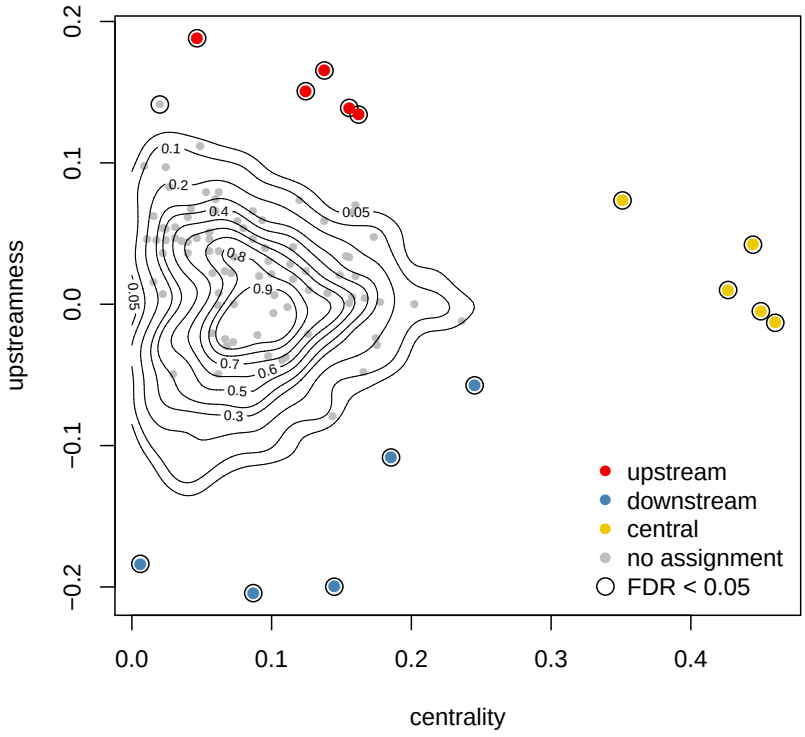


Figure E.3 Upstreamness and centrality, with their joint null distribution shown with contour lines. Upstream, downstream, and central spike-ins all had significant regulatory tendencies ($FDR < 0.05$).

We can see that our upstream spike-ins have positive upstreamness, the downstream spike-ins have negative upstreamness, and the central spike-ins have high centrality. In addition, all spike-ins are well-separated from the 2D null distribution, and all have $FDR < 0.05$, in contrast to our attempt with the graph theoretical attempt. There is one false positive, which can be seen near the upstream spike-ins.

In this exercise we have demonstrated that our approach can perform quite well on data where regulatory roles are determined more by distributional enrichment than by individual connections.

Appendix F

Tutorial: Finding topic-related genes in PubMed

F.1 Introduction

It is often quite useful to know which genes are cited to a significant degree in connection with a concept, medical condition, or other phenotype. There are several online tools which make this possible, such as GOPubMed, but here we demonstrate a simple and low-level way to accomplish essentially the same thing. Doing this programmatically, rather than through a GUI or web interface, makes it easier to integrate the results into a larger workflow.

Here we make use of two valuable resources that NCBI provides: the `gene2pubmed` file, which is a semi-curated mapping between genes and the PubMed IDs (PMIDs) of papers where they are mentioned, and the eUtils web service API, which allows us to perform PubMed queries programmatically.

F.2 Setup

Download the `gene2pubmed` file from ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/, uncompress it, and place it in your current working directory. Then use the `source` function to load the functions needed for this tutorial, and read in the contents of `gene2pubmed`.

```
> source("functions.R")
> dat = read.table("gene2pubmed", sep = "\t", skip = 1)
> head(dat)
```

```
  V1      V2      V3
1  9 1246500 9873079
2  9 1246501 9873079
3  9 1246502 9812361
4  9 1246502 9873079
5  9 1246503 9873079
6  9 1246504 9873079
```

The `gene2pubmed` file contains three columns: taxonomy ID, Entrez Gene ID, and PMID.

F.3 Retrieving PubMed IDs of interest

Now that we have mappings between genes and papers where they are cited, we would like to retrieve a list of PMIDs related to a search term, and then cross-reference the two lists of publications. As an example, we will retrieve PMIDs of papers dealing with schizophrenia. We'll use a few R wrapper functions that interface with NCBI's eUtils platform.

```
> sz.pprs = getIDs("schizophrenia")
```

F.4 Cross-referencing the lists

First let's decide which organism we're interested in. Look at http://www.ncbi.nlm.nih.gov/Taxonomy/ to get your organism's taxonomy ID, then you can reduce the number of records you're dealing with.

```
> dat = as.matrix(dat[dat[, 1] == "9606", ])
```

Now get papers that cite at least one gene:

```
> wgene = sz.pprs %in% dat[, 3]
> wgene = sz.pprs[wgene]
```

Look at it the other way around – find genes that are cited at least once:

```
> genes = dat[dat[, 3] %in% wgene, 2]
```

How many genes are cited in connection with schizophrenia? What are the top 5 most-cited genes?

```
> length(unique(genes))

[1] 2198

> sort(table(genes), dec = T)[1:5]

genes
 1312  1813  3084 84062 27185
  209   129   115    99    93
```

F.5 Significance of association

We now know which are the most frequently cited genes in connection with schizophrenia. Unfortunately at this point we don't know which genes are *significantly* associated with schizophrenia, based on literature citation. Some of the frequently-cited genes might just be genes that are frequently cited in the literature in general, regardless of the context.

We can get at the question of significance if we construct a 2×2 contingency table and then perform Fisher's exact test, which will tell us if the frequency of citation in schizophrenia papers is significant, given the overall citation frequency of a gene. Such a use of contingency tables in text mining is intuitive and has been treated previously (Pedersen, 1996). Here we use a convenience wrapper function to simplify

the process of repeatedly constructing and testing the contingency tables. This will take several minutes, and mostly depends on the number of genes tested (length of `genes` vector) and the number of rows in `dat`. The `fetSpec` function prints its progress, which will help you gauge how long it will take to complete.

```
> schizo.specific = fetSpec(names(sort(table(genes), dec = T)),
+     dat, wgene)
> P = sapply(schizo.specific, function(x) x$test$p.v)
> OR = sapply(schizo.specific, function(x) x[[2]]$est)
```

The `P` vector now contains *P* values indicating the significance of association between a gene and schizophrenia, as determined by citation frequencies. The `OR` vector contains the corresponding odds ratios. These results may be used together with other evidences to develop a list of genes relevant to schizophrenia.

Appendix G

Performance notes for Random Forests

Here we'll look at the runtime of Random Forests and how it is affected when the most important arguments are changed. The hardware used here is an 8-core workstation (Intel Xeon X5472 at 3 GHz) with 12 GB RAM.

G.1 Setup

```
> source("functions.R")
> library(randomForest)
> X = list()
> X[[1]] = simgeno(200)
> X[[2]] = simgeno(400)
> X[[3]] = simgeno(600)
> X[[4]] = simgeno(800)
> X[[5]] = simgeno(1000)
> X[[6]] = simgeno(1500)
> X[[7]] = simgeno(2000)
> X[[8]] = simgeno(3000)
> Y = list()
> Y[[1]] = rnorm(200)
> Y[[2]] = rnorm(400)
> Y[[3]] = rnorm(600)
> Y[[4]] = rnorm(800)
> Y[[5]] = rnorm(1000)
> Y[[6]] = rnorm(1500)
> Y[[7]] = rnorm(2000)
> Y[[8]] = rnorm(3000)
```

G.2 Sample size

Let's take a look at how the number of observations influences the runtime of Random Forests. The number of variables is 1,000, `ntree` is 1,000 and `mtry` and `nodesize` are set to their defaults of $\frac{p}{3}$ and 5,

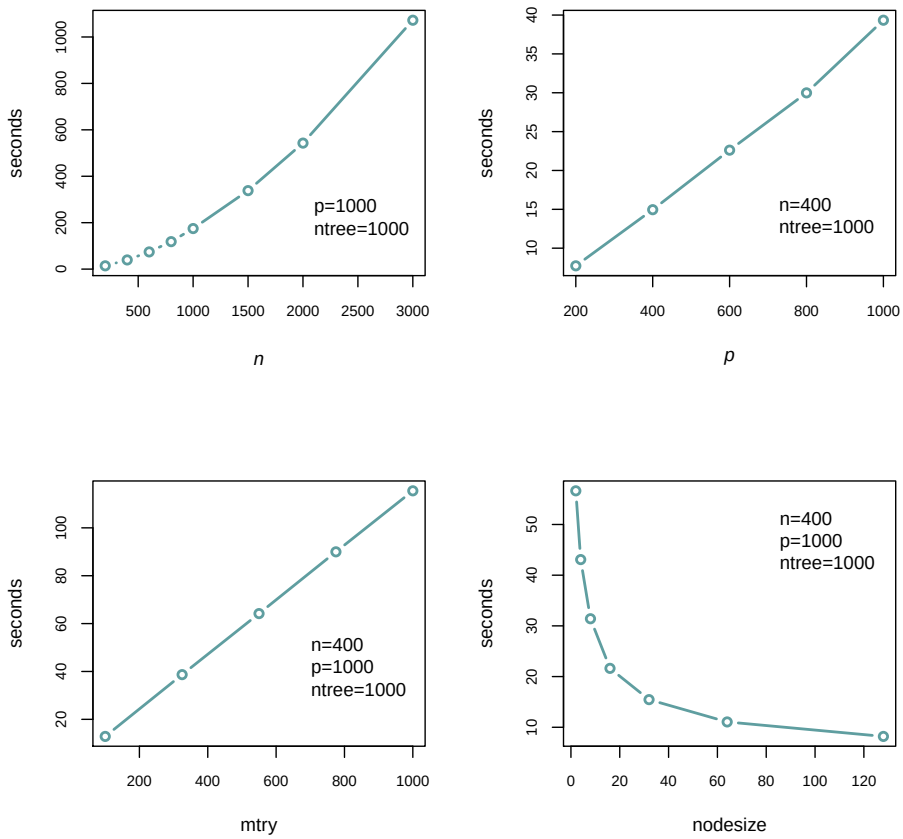


Figure G.1 Effect of data dimension and RF arguments on wall time performance. Unless noted, arguments were left at their default values.

respectively.

```
> t1 = sapply(1:8,function(i) system.time(randomForest(y=Y[[i]],x=X[[i]],ntree=1000))[3])
```

Changing *n* influences the depth of the trees grown. If *n* is higher, deeper trees are needed to reach the specified value of `nodesize`. Increasing tree depth leads to a near exponential increase in splits.

G.3 Number of features

Here we will fix the number of observations (*n*) at 400. `ntree` will be 1,000 and all other arguments will be set to default values. We will vary *p* from 200 to 1000, with increments of 200.

```
> p = seq(200,1000,200)
> t2 = sapply(1:5,function(i)
+   system.time(randomForest(y=Y[[2]],x=X[[2]][,1:p[i]],ntree=1000))[3])
```

Changing *p* leads to a roughly proportional change in runtime.

G.4 Varying `mtry`

Here we will keep the number of observations (*n*) and variables (*p*) fixed at 400 and 1,000, respectively. `ntree` will be 1,000 and all others will be set to default values. We will vary `mtry` from 10% to 100% of *p*.

```
> mt = seq(0.1,1,length.out=5) * 1000
> t3 = sapply(1:5,
+ function(i) system.time(randomForest(y=Y[[2]],x=X[[2]],ntree=1000,mtry=mt[i]))[3])
```

As seen in Figure G.1, changing `mtry` essentially causes proportional changes in runtime. This is not surprising as changing `mtry` is similar to changing *p*. Note: we'll skip looking at `ntree` because it's fairly trivial to see that runtime will be linear with the number of trees.

G.5 Varying `nodesize`

We'll now try varying `nodesize`, which is the argument that controls how deep trees are grown.

```
> ns = 2^(1:7)
> t4 = sapply(1:7,
+ function(i) system.time(randomForest(y=Y[[2]],x=X[[2]],ntree=1000,nodesize=ns[i]))[3])
```

Of all of the parameters probes so far, `nodesize` shows the most extreme nonlinear behavior. This makes sense, since it controls the depths of the trees, which in turn influences the number of splits to be performed in a near-exponential way.

G.6 Scalability of RF across multiple processors

Until this point, all calculations have been performed on a single processor. Since the individual trees in a Random Forest are independent of each other, they can be easily spread across multiple processors to improve performance. Let's take a look at how performance scales up to 8 processors (Fig. G.2).

```
> parRF = function(ntree,x,y,cl){
+  require(randomForest)
+  ssplit = function(x,ngroups){
+   structure(split(x,cut(x,ngroups)),names=NULL)
+  }
+  clusterEvalQ(cl,library(randomForest))
+  ntree = ssplit(1:ntree,length(cl))
+  ntree = unlist(lapply(ntree,length))
+  rf = parLapply(cl,ntree,
+   function(i,y,x) randomForest(y=y,x=x,ntree=i,mtry=0.9*ncol(x)),
+   y,
+   x)
+  rf = do.call('combine',rf)
```

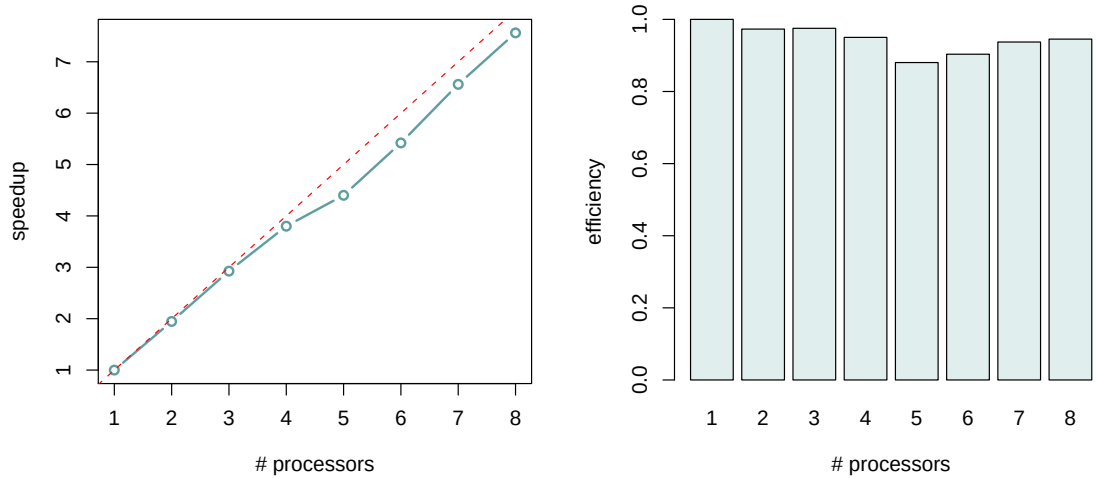


Figure G.2 Parallel scalability of RF when run across multiple processors.

```
+  return(rf)
+ }
> cl = lapply(1:8,makeSOCKcluster)
> t5 = numeric(8)
> t5[1] = system.time(randomForest(x=X[[1]],y=Y[[1]],ntree=5000,mtry=900))[3]
> t5[2] = system.time(parRF(5000,X[[1]],Y[[1]],cl[[2]]))[3]
> t5[3] = system.time(parRF(5000,X[[1]],Y[[1]],cl[[3]]))[3]
> t5[4] = system.time(parRF(5000,X[[1]],Y[[1]],cl[[4]]))[3]
> t5[5] = system.time(parRF(5000,X[[1]],Y[[1]],cl[[5]]))[3]
> t5[6] = system.time(parRF(5000,X[[1]],Y[[1]],cl[[6]]))[3]
> t5[7] = system.time(parRF(5000,X[[1]],Y[[1]],cl[[7]]))[3]
> t5[8] = system.time(parRF(5000,X[[1]],Y[[1]],cl[[8]]))[3]
```

As shown in figure G.2, performance scales as we would expect, which is close to linear. There is a noticeable dip in efficiency at 5 cores, perhaps due to uneven loading when the number of threads "spills over" to occupy two of the physical processors. However, speedup and efficiency are nearly recovered at 8 cores.

List of Figures

List of Tables

References

Alexa A, Rahnenführer J, and Lengauer T (2006). Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics* **22**: 1600–7

Allen E, Horvath S, Tong F, Kraft P, Spiteri E, Riggs AD, and Marahrens Y (2003). High concentrations of long interspersed nuclear element sequence distinguish monoallelically expressed genes. *Proc. Natl. Acad. Sci. U.S.A.* **100**: 9940–5

Allen NC, Bagade S, McQueen MB, Ioannidis JPA, Kavvoura FK, Khoury MJ, Tanzi RE, and Bertram L (2008). Systematic meta-analyses and field synopsis of genetic association studies in schizophrenia: the SzGene database. *Nat. Genet.* **40**: 827–34

Altmann A, Toloşi L, Sander O, and Lengauer T (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics* **26**: 1340–7

Bansal M, Della Gatta G, and Di Bernardo D (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics* **22**: 815–22

Benjamini Y and Yekutieli D (2005). Quantitative trait Loci analysis using the false discovery rate. *Genetics* **171**: 783–90

Beyer A, Workman C, Hollunder J, Radke D, Möller U, Wilhelm T, and Ideker T (2006). Integrated assessment and prediction of transcription factor binding. *PLoS Comput. Biol.* **2**: e70

Bing N and Hoeschele I (2005). Genetical genomics analysis of a yeast segregant population for transcription network inference. *Genetics* **170**: 533–42

Boutros PC, Bielefeld KA, Pohjanvirta R, and Harper PA (2009). Dioxin-dependent and dioxin-independent gene batteries: comparison of liver and kidney in AHR-null mice. *Toxicol. Sci.* **112**: 245–56

Braff DL, Freedman R, Schork NJ, and Gottesman II (2007). Deconstructing schizophrenia: an overview of the use of endophenotypes in order to understand a complex disorder. *Schizophr Bull* **33**: 21–32

Breiman L (2001). Random Forests. *Machine Learning* **45**: 5

Breiman L and Cutler A (2003). *Random Forests Manual v4.0*. Tech. rep. UC Berkeley. URL: ftp://ftp.stat.berkeley.edu/pub/users/breiman/Using_random_forests_v4.0.pdf

Brem RB and Kruglyak L (2005). The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proc. Natl. Acad. Sci. U.S.A.* **102**: 1572–7

Brem RB, Storey JD, Whittle J, and Kruglyak L (2005). Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature* **436**: 701–3

Broman KW and Speed TP (2002). A model selection approach for the identification of quantitative trait loci in experimental crosses. *J. Roy. Statist. Soc. B.* **64**: 641

Broman KW, Wu H, With ideas from Gary Churchill, Sen S, and Contributions from Brian Yandell (2008). *qtl: Tools for analyzing QTL experiments*. R package version 1.09-43. URL: http://www.rqtl.org

Bureau A, Dupuis J, Hayward B, Falls K, and Van Eerdewegh P (2003). Mapping complex traits using Random Forests. *BMC Genet.* **4 Suppl 1**: S64

Bureau A, Dupuis J, Falls K, Lunetta KL, Hayward B, Keith TP, and Van Eerdewegh P (2005). Identifying SNPs predictive of phenotype using random forests. *Genet. Epidemiol.* **28**: 171–82

Bystrykh L et al. (2005). Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics'. *Nat. Genet.* **37**: 225–32

Carlborg O and Haley CS (2004). Epistasis: too often neglected in complex trait studies? *Nat. Rev. Genet.* **5**: 618–25

Carlson EA, McCulloch C, Koganti A, Goodwin SB, Sutter TR, and Silkworth JB (2009). Divergent transcriptomic responses to aryl hydrocarbon receptor agonists between rat and human primary hepatocytes. *Toxicol. Sci.* **112**: 257–72

Carney SA, Chen J, Burns CG, Xiong KM, Peterson RE, and Heideman W (2006). Aryl hydrocarbon receptor activation produces heart-specific transcriptional and toxic responses in developing zebrafish. *Mol. Pharmacol.* **70**: 549–61

Chan EKF, Hawken R, and Reverter A (2009). The combined effect of SNP-marker and phenotype attributes in genome-wide association studies. *Anim. Genet.* **40**: 149–56

Chang JS et al. (2008). Pathway analysis of single-nucleotide polymorphisms potentially associated with glioblastoma multiforme susceptibility using random forests. *Cancer Epidemiol. Biomarkers Prev.* **17**: 1368–73

Chen X et al. (2010). An eQTL analysis of partial resistance to Puccinia hordei in barley. *PLoS ONE* **5**: e8598

Chen Y et al. (2008). Variations in DNA elucidate molecular networks that cause disease. *Nature* **452**: 429–35

Chun H and Keles S (2009). Expression quantitative trait loci mapping with multivariate sparse partial least squares regression. *Genetics* **182**: 79–90

Cordell HJ (2002). Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Hum. Mol. Genet.* **11**: 2463–8

Cordell HJ (2009). Detecting gene-gene interactions that underlie human diseases. *Nat. Rev. Genet.* **10**: 392–404

Costanzo M et al. (2010). The genetic landscape of a cell. *Science* **327**: 425–31

Dere E, Boverhof DR, Burgoon LD, and Zacharewski TR (2006). In vivo-in vitro toxicogenomic comparison of TCDD-elicited gene expression in Hepa1c1c7 mouse hepatoma cells and C57BL/6 hepatic tissue. *BMC Genomics* **7**: 80

Dong C, Chu X, Wang Y, Wang Y, Jin L, Shi T, Huang W, and Li Y (2008). Exploration of gene-gene interaction effects using entropy-based methods. *Eur. J. Hum. Genet.* **16**: 229–35

Druka A et al. (2008). Exploiting regulatory variation to identify genes underlying quantitative resistance to the wheat stem rust pathogen Puccinia graminis f. sp. tritici in barley. *Theor. Appl. Genet.* **117**: 261–72

DuSell CD, Nelson ER, Wittmann BM, Fretz JA, Kazmin D, Thomas RS, Pike JW, and McDonnell DP (2010). Regulation of aryl hydrocarbon receptor function by selective estrogen receptor modulators. *Mol. Endocrinol.* **24**: 33–46

Elbi C, Misteli T, and Hager GL (2002). Recruitment of dioxin receptor to active transcription sites. *Mol. Biol. Cell* **13**: 2001–15

Foster SD (2007). Incorporating LASSO Effects into a Mixed Model for Quantitative Trait Loci Detection. *J. Agric. Biol. Envir. S.* **12**: 300

Frericks M, Burgoon LD, Zacharewski TR, and Esser C (2008). Promoter analysis of TCDD-inducible genes in a thymic epithelial cell line indicates the potential for cell-specific transcription factor crosstalk in the AhR response. *Toxicol. Appl. Pharmacol.* **232**: 268–79

García-Magariños M, López-de Ullibarri I, Cao R, and Salas A (2009). Evaluating the ability of tree-based methods and logistic regression for the detection of SNP-SNP interaction. *Ann. Hum. Genet.* **73**: 360–9

Georgieva L et al. (2006). Convergent evidence that oligodendrocyte lineage transcription factor 2 (OLIG2) and interacting genes influence susceptibility to schizophrenia. *Proc. Natl. Acad. Sci. U.S.A.* **103**: 12469–74

Ghazalpour A et al. (2006). Integrating genetic and network analysis to characterize genes related to mouse weight. *PLoS Genet.* **2**: e130

Gohlke JM, Stockton PS, Sieber S, Foley J, and Portier CJ (2009). AhR-mediated gene expression in the developing mouse telencephalon. *Reprod. Toxicol.* **28**: 321–8

Goldstein BA, Hubbard AE, Cutler A, and Barcellos LF (2010). An application of Random Forests to a genome-wide association dataset: methodological considerations and new findings. *BMC Genet.* **11**: 49

Gonçalves JP, Madeira SC, and Oliveira AL (2009). BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data. *BMC Res Notes* **2**: 124

Haley CS and Knott SA (1992). A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. *Heredity* **69**: 315–24

Han L (2006). Random Forests Feature Selection with Kernel Partial Least Squares: Detecting Ischemia from MagnetoCardiograms. In: *The European Symposium on Artificial Neural Networks*, pp. 221–226

Hannum G, Srivas R, Guénolé A, Van Attikum H, Krogan NJ, Karp RM, and Ideker T (2009). Genome-wide association data reveal a global map of genetic interactions among protein complexes. *PLoS Genet.* **5**: e1000782

Hockley SL, Arlt VM, Brewer D, Giddings I, and Phillips DH (2006). Time- and concentration-dependent changes in gene expression induced by benzo(a)pyrene in two human cell lines, MCF-7 and HepG2. *BMC Genomics* **7**: 260

Hockley SL, Arlt VM, Brewer D, Te Poele R, Workman P, Giddings I, and Phillips DH (2007). AHR- and DNA-damage-mediated gene expression responses induced by benzo(a)pyrene in human cell lines. *Chem. Res. Toxicol.* **20**: 1797–810

Hoppmann J, Perwitz N, Meier B, Fasshauer M, Hadaschik D, Lehnert H, and Klein J (2010). The balance between gluco- and mineralo-corticoid action critically determines inflammatory adipocyte responses. *J. Endocrinol.* **204**: 153–64

Huang Y, Wuchty S, Ferdig MT, and Przytycka TM (2009). Graph theoretical approach to study eQTL: a case study of Plasmodium falciparum. *Bioinformatics* **25**: i15–20

Hughes TR et al. (2000). Functional discovery via a compendium of expression profiles. *Cell* **102**: 109–26

Hwang DH, Kim BG, Kim EJ, Lee SI, Joo IS, Suh-Kim H, Sohn S, and Kim SU (2009). Transplantation of human neural stem cells transduced with Olig2 transcription factor improves locomotor recovery and enhances myelination in the white matter of rat spinal cord following contusive injury. *BMC Neurosci* **10**: 117

Jack P and Brookes P (1980). The binding of benzo(a)pyrene to DNA components of differing sequence complexity. *Int. J. Cancer* **25**: 789–95

Jakovcevski I and Zecevic N (2005). Olig transcription factors are expressed in oligodendrocyte and neuronal cells in human fetal CNS. *J. Neurosci.* **25**: 10064–73

Jegga AG, Gowrisankar S, Chen J, and Aronow BJ (2007). PolyDoms: a whole genome database for the identification of non-synonymous coding SNPs with the potential to impact disease. *Nucleic Acids Res.* **35**: D700–6

Jiang C and Zeng ZB (1995). Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics* **140**: 1111–27

Kanehisa M and Goto S (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**: 27–30

Kaufman L and Rousseeuw P (1990). *Finding groups in data: an Introduction to cluster analysis.* New York: Wiley

Kempermann G, Chesler EJ, Lu L, Williams RW, and Gage FH (2006). Natural variation and genetic covariance in adult hippocampal neurogenesis. *Proc. Natl. Acad. Sci. U.S.A.* **103**: 780–5

Kerns D, Vong GS, Barley K, Dracheva S, Katsel P, Casaccia P, Haroutunian V, and Byne W (2010). Gene expression abnormalities and oligodendrocyte deficits in the internal capsule in schizophrenia. *Schizophr. Res.* **120**: 150–8

Keurentjes JJB, Fu J, Terpstra IR, Garcia JM, Van den Ackerveken G, Snoek LB, Peeters AJM, Vreugdenhil D, Koornneef M, and Jansen RC (2007). Regulatory network construction in Arabidopsis by using genome-wide gene expression quantitative trait loci. *Proc. Natl. Acad. Sci. U.S.A.* **104**: 1708–13

Kim S, Dere E, Burgoon LD, Chang CC, and Zacharewski TR (2009a). Comparative analysis of AhR-mediated TCDD-elicited gene expression in human liver adult stem cells. *Toxicol. Sci.* **112**: 229–44

Kim Y, Wojciechowski R, Sung H, Mathias RA, Wang L, Klein AP, Lenroot RK, Malley J, and Bailey-Wilson JE (2009b). Evaluation of random forests performance for genome-wide association studies in the presence of interaction effects. *BMC Proceedings* **3**: S64

Köhle C and Bock KW (2007). Coordinate regulation of Phase I and II xenobiotic metabolisms by the Ah receptor and Nrf2. *Biochem. Pharmacol.* **73**: 1853–62

La Merrill M, Gordon RR, Hunter KW, Threadgill DW, and Pomp D (2010). Dietary fat alters pulmonary metastasis of mammary cancers through cancer autonomous and non-autonomous changes in gene expression. *Clin. Exp. Metastasis* **27**: 107–16

Lander ES and Botstein D (1989). Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics* **121**: 185–199

Lee SSF, Sun L, Kustra R, and Bull SB (2008). EM-random forest and new measures of variable importance for multi-locus quantitative trait linkage analysis. *Bioinformatics* **24**: 1603–10

Lee SI, Pe'er D, Dudley AM, Church GM, and Koller D (2006). Identifying regulatory mechanisms using individual variation reveals key role for chromatin modification. *Proc. Natl. Acad. Sci. U.S.A.* **103**: 14062–7

Lee SI, Dudley AM, Drubin D, Silver PA, Krogan NJ, Pe'er D, and Koller D (2009). Learning a prior on regulatory potential from eQTL data. *PLoS Genet.* **5**: e1000358

Liaw A and Wiener M (2002). Classification and Regression by randomForest. *R News* **2**: 18–22

Liu B, De la Fuente A, and Hoeschele I (2008). Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics* **178**: 1763–76

Liu RM, Vasiliou V, Zhu H, Duh JL, Tabor MW, Puga A, Nebert DW, Sainsbury M, and Shertzer HG (1994). Regulation of [Ah] gene battery enzymes and glutathione levels by 5,10-dihydroindeno[1,2-b]indole in mouse hepatoma cell lines. *Carcinogenesis* **15**: 2347–52

Lu X, Shao J, Li H, and Yu Y (2009). Early whole-genome transcriptional response induced by benzo[a]pyrene diol epoxide in a normal human cell line. *Genomics* **93**: 332–42

Lu X, Shao J, Li H, and Yu Y (2010). Temporal gene expression changes induced by a low concentration of benzo[a]pyrene diol epoxide in a normal human cell line. *Mutat. Res.* **684**: 74–80

Lunetta K, Hayward L, Segal J, and Van Eerdewegh P (2004). Screening large-scale association study data: exploiting interactions using random forests. *BMC Genet.* **5**: 32

Madeira SC and Oliveira AL (2009). A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series. *Algorithms Mol Biol* **4**: 8

Madeira SC, Teixeira MC, Sá-Correia I, and Oliveira AL (2010). Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. *IEEE/ACM Trans Comput Biol Bioinform* **7**: 153–65

Marissal-Arvy N, Langlois A, Tridon C, and Mormede P (2010). Functional variability in corticosteroid receptors is a major component of strain differences in fat deposition and metabolic consequences of enriched diets in rat. *Metab. Clin. Exp.*

Mattsson A, Jernström B, Cotgreave IA, and Bajak E (2009). H2AX phosphorylation in A549 cells induced by the bulky and stable DNA adducts of benzo[a]pyrene and dibenzo[a,l]pyrene diol epoxides. *Chem. Biol. Interact.* **177**: 40–7

Maycox PR et al. (2009). Analysis of gene expression in two large schizophrenia cohorts identifies multiple changes associated with nerve terminal function. *Mol. Psychiatry* **14**: 1083–94

McKinney BA, Reif DM, Ritchie MD, and Moore JH (2006). Machine learning for detecting gene-gene interactions: a review. *Appl. Bioinformatics* **5**: 77–88

McKinney BA, Crowe JE, Guo J, and Tian D (2009). Capturing the spectrum of interaction effects in genetic association studies by simulated evaporative cooling network analysis. *PLoS Genet.* **5**: e1000432

Miao W, Hu L, Scrivens PJ, and Batist G (2005). Transcriptional regulation of NF-E2 p45-related factor (NRF2) expression by the aryl hydrocarbon receptor-xenobiotic response element signaling pathway: direct cross-talk between phase I and II drug-metabolizing enzymes. *J. Biol. Chem.* **280**: 20340–8

Michaelson JJ, Loguercio S, and Beyer A (2009). Detection and interpretation of expression quantitative trait loci (eQTL). *Methods* **48**: 265–76

Michaelson JJ, Alberts R, Schughart K, and Beyer A (2010). Data-driven assessment of eQTL mapping methods. *BMC Genomics* **11**: 502

Michailidou Z et al. (2008). Glucocorticoid receptor haploinsufficiency causes hypertension and attenuates hypothalamic-pituitary-adrenal axis and blood pressure adaptions to high-fat diet. *FASEB J.* **22**: 3896–907

Mnaimneh S et al. (2004). Exploration of essential gene functions via titratable promoter alleles. *Cell* **118**: 31–44

Moore JH (2003). The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Hum. Hered.* **56**: 73–82

Moore JH and Williams SM (2009). Epistasis and its implications for personal genetics. *Am. J. Hum. Genet.* **85**: 309–20

Motsinger-Reif AA, Reif DM, Fanelli TJ, and Ritchie MD (2008). A comparison of analytical methods for genetic association studies. *Genet. Epidemiol.* **32**: 767–78

Naeve GS, Ramakrishnan M, Kramer R, Hevroni D, Citri Y, and Theill LE (1997). Neuritin: a gene induced by neural activity and neurotrophins that promotes neuritogenesis. *Proc. Natl. Acad. Sci. U.S.A.* **94**: 2648–53

Narayan S, Tang B, Head SR, Gilmartin TJ, Sutcliffe JG, Dean B, and Thomas EA (2008). Molecular profiles of schizophrenia in the CNS at different stages of illness. *Brain Res.* **1239**: 235–48

Nebert DW, Puga A, and Vasiliou V (1993). Role of the Ah receptor and the dioxin-inducible [Ah] gene battery in toxicity, cancer, and signal transduction. *Ann. N. Y. Acad. Sci.* **685**: 624–40

Nebert DW, Roe AL, Dieter MZ, Solis WA, Yang Y, and Dalton TP (2000). Role of the aromatic hydrocarbon receptor and [Ah] gene battery in the oxidative stress response, cell cycle control, and apoptosis. *Biochem. Pharmacol.* **59**: 65–85

Nebert DW, Dalton TP, Okey AB, and Gonzalez FJ (2004). Role of aryl hydrocarbon receptor-mediated induction of the CYP1 enzymes in environmental toxicity and cancer. *J. Biol. Chem.* **279**: 23847–50

Nelson MR, Kardia SL, Ferrell RE, and Sing CF (2001). A combinatorial partitioning method to identify multilocus genotypic partitions that predict quantitative trait variation. *Genome Res.* **11**: 458–70

Nicolae DL, Gamazon E, Zhang W, Duan S, Dolan ME, and Cox NJ (2010). Trait-associated SNPs are more likely to be eQTLs: annotation to enhance discovery from GWAS. *PLoS Genet.* **6**: e1000888

Overall RW et al. (2009). Genetics of the hippocampal transcriptome in mouse: a systematic survey and online neurogenomics resource. *Front Neurosci* **3**: 55

Pang H and Zhao H (2008). Building pathway clusters from Random Forests classification using class votes. *BMC Bioinformatics* **9**: 87

Pang H, Lin A, Holford M, Enerson BE, Lu B, Lawton MP, Floyd E, and Zhao H (2006). Pathway analysis using random forests classification and regression. *Bioinformatics* **22**: 2028–36

Pang H, Datta D, and Zhao H (2010). Pathway analysis using random forests with bivariate node-split for survival outcomes. *Bioinformatics* **26**: 250–8

Passos Gregorio S, Gattaz WF, Tavares H, Kieling C, Timm S, Wang AG, Berg Rasmussen H, Werge T, and Dias-Neto E (2006). Analysis of coding-polymorphisms in NOTCH-related genes reveals NUMBL poly-glutamine repeat to be associated with schizophrenia in Brazilian and Danish subjects. *Schizophr. Res.* **88**: 275–82

Pedersen T (1996). Fishing for Exactness. In: *In Proceedings of the South-Central SAS Users Group Conference*, pp. 188–200

Petretto E et al. (2006). Heritability and tissue specificity of expression quantitative trait loci. *PLoS Genet.* **2**: e172

Phillips PC (2008). Epistasis–the essential role of gene interactions in the structure and evolution of genetic systems. *Nat. Rev. Genet.* **9**: 855–67

Redestig H, Weicht D, Selbig J, and Hannah MA (2007). Transcription factor target prediction using multiple short expression time series from Arabidopsis thaliana. *BMC Bioinformatics* **8**: 454

Ritchie MD, Hahn LW, Roodi N, Bailey LR, Dupont WD, Parl FF, and Moore JH (2001). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *Am. J. Hum. Genet.* **69**: 138–47

Rockman MV and Kruglyak L (2006). Genetics of global gene expression. *Nat. Rev. Genet.* **7**: 862–72

Ruan J, Deng Y, Perkins EJ, and Zhang W (2009). An ensemble learning approach to reverse-engineering transcriptional regulatory networks from time-series gene expression data. *BMC Genomics* **10 Suppl 1**: S8

Rudd MF, Williams RD, Webb EL, Schmidt S, Sellick GS, and Houlston RS (2005). The predicted impact of coding single nucleotide polymorphisms database. *Cancer Epidemiol. Biomarkers Prev.* **14**: 2598–604

Sartor MA et al. (2009). Genomewide analysis of aryl hydrocarbon receptor binding targets reveals an extensive array of gene clusters that control morphogenetic and developmental programs. *Environ. Health Perspect.* **117**: 1139–46

Schadt EE and Lum PY (2006). Thematic review series: systems biology approaches to metabolic and cardiovascular disorders. Reverse engineering gene networks to identify key drivers of complex disease phenotypes. *J. Lipid Res.* **47**: 2601–13

Schadt EE et al. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nat. Genet.* **37**: 710–7

Schuldiner M et al. (2005). Exploration of the function and organization of the yeast early secretory pathway through an epistatic miniarray profile. *Cell* **123**: 507–19

Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, and Friedman N (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.* **34**: 166–76

Shao H et al. (2008). Genetic architecture of complex traits: large phenotypic effects and pervasive epistasis. *Proc. Natl. Acad. Sci. U.S.A.* **105**: 19910–4

Shi T and Horvath S (2006). Unsupervised Learning With Random Forest Predictors. *J. Comput. Graph. Stat.* **15**: 118

Shi T, Seligson D, Belldegrun AS, Palotie A, and Horvath S (2005). Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma. *Mod. Pathol.* **18**: 547–57

Shin S, Wakabayashi N, Misra V, Biswal S, Lee GH, Agoston ES, Yamamoto M, and Kensler TW (2007). NRF2 modulates aryl hydrocarbon receptor signaling: influence on adipogenesis. *Mol. Cell. Biol.* **27**: 7188–97

Sieberts SK and Schadt EE (2007). Moving toward a system genetics view of disease. *Mamm. Genome* **18**: 389–401

Strobl C, Boulesteix AL, Zeileis A, and Hothorn T (2007). Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics* **8**: 25

Supper J, Strauch M, Wanke D, Harter K, and Zell A (2007). EDISA: extracting biclusters from multiple time-series of gene expression profiles. *BMC Bioinformatics* **8**: 334

Suthram S, Beyer A, Karp RM, Eldar Y, and Ideker T (2008). eQED: an efficient method for interpreting eQTL associations using protein networks. *Mol. Syst. Biol.* **4**: 162

Swedenborg E and Pongratz I (2010). AhR and ARNT modulate ER signaling. *Toxicology* **268**: 132–8

Tan Z, Chang X, Puga A, and Xia Y (2002). Activation of mitogen-activated protein kinases (MAPKs) by aromatic hydrocarbons: role in the regulation of aryl hydrocarbon receptor (AHR) function. *Biochem. Pharmacol.* **64**: 771–80

Tibshirani R (1996). Regression Shrinkage and Selection Via the Lasso. *J. Roy. Statist. Soc. B.* **58**: 267–288

Tijet N, Boutros PC, Moffat ID, Okey AB, Tuomisto J, and Pohjanvirta R (2006). Aryl hydrocarbon receptor regulates distinct dioxin-dependent and dioxin-independent gene batteries. *Mol. Pharmacol.* **69**: 140–53

Tong AHY et al. (2004). Global mapping of the yeast genetic interaction network. *Science* **303**: 808–13

Veyrieras JB, Kudaravalli S, Kim SY, Dermitzakis ET, Gilad Y, Stephens M, and Pritchard JK (2008). High-resolution mapping of expression-QTLs yields insight into human gene regulation. *PLoS Genet.* **4**: e1000214

Viñuela A, Snoek LB, Riksen JAG, and Kammenga JE (2010). Genome-wide gene expression regulation as a function of genotype and age in C. elegans. *Genome Res.* **20**: 929–37

Vrzal R, Stejskalova L, Monostory K, Maurel P, Bachleda P, Pavek P, and Dvorak Z (2009). Dexamethasone controls aryl hydrocarbon receptor (AhR)-mediated CYP1A1 and CYP1A2 expression and activity in primary cultures of human hepatocytes. *Chem. Biol. Interact.* **179**: 288–96

Wang G, Yin L, Zhao Y, and Mao K (2010a). Efficiently mining time-delayed gene expression patterns. *IEEE Trans. Syst. Man. Cybern. B. Cybern.* **40**: 400–11

Wang J, Yu H, Xie W, Xing Y, Yu S, Xu C, Li X, Xiao J, and Zhang Q (2010b). A global analysis of QTLs for expression variations in rice shoots at the early seedling stage. *Plant J.* **63**: 1063–74

Wang J, Williams RW, and Manly KF (2003). WebQTL: web-based complex trait analysis. *Neuroinformatics* **1**: 299–308

Wang M, Chen X, and Zhang H (2010c). Maximal conditional chi-square importance in random forests. *Bioinformatics* **26**: 831–7

Wang SH, Liang CT, Liu YW, Huang MC, Huang SC, Hong WF, and Su JGJ (2009). Crosstalk between activated forms of the aryl hydrocarbon receptor and glucocorticoid receptor. *Toxicology* **262**: 87–97

Wang X, Yang N, Uno E, Roeder RG, and Guo S (2006). A subunit of the mediator complex regulates vertebrate neuronal development. *Proc. Natl. Acad. Sci. U.S.A.* **103**: 17284–9

Wessel J, Zapala MA, and Schork NJ (2007). Accommodating pathway information in expression quantitative trait locus analysis. *Genomics* **90**: 132–42

Wihlén B, Ahmed S, Inzunza J, and Matthews J (2009). Estrogen receptor subtype- and promoter-specific modulation of aryl hydrocarbon receptor-dependent transcription. *Mol. Cancer Res.* **7**: 977–86

Wu C et al. (2008). Gene set enrichment in eQTL data identifies novel annotations and pathway regulators. *PLoS Genet.* **4**: e1000070

Xiao Y and Segal MR (2009). Identification of yeast transcriptional regulation networks using multivariate random forests. *PLoS Comput. Biol.* **5**: e1000414

Xu S, Weerachayaphorn J, Cai SY, Soroka CJ, and Boyer JL (2010). Aryl hydrocarbon receptor and NF-E2-related factor 2 are key regulators of human MRP4 expression. *Am. J. Physiol. Gastrointest. Liver Physiol.* **299**: G126–35

Yoon K and Gaiano N (2005). Notch signaling in the mammalian central nervous system: insights from mouse mutants. *Nat. Neurosci.* **8**: 709–15

Yu YX, Shen L, Xia P, Tang YW, Bao L, and Pei G (2006). Syntaxin 1A promotes the endocytic sorting of EAAC1 leading to inhibition of glutamate transport. *J. Cell. Sci.* **119**: 3776–87

Zeng ZB (1994). Precision mapping of quantitative trait loci. *Genetics* **136**: 1457–68

Zou H and Hastie T (2005). Regularization and variable selection via the Elastic Net. *J. Roy. Statist. Soc. B.* **67**: 301–320