
easyGWAS: An Integrated Computational Framework for Advanced Genome-Wide Association Studies

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Dominik Gerhard Grimm
aus Burglengenfeld

Tübingen
2015

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der
Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation: 19.11.2015

Dekan: Prof. Dr. Wolfgang Rosenstiel

1. Berichterstatter: Prof. Dr. Karsten Borgwardt

2. Berichterstatter: Prof. Dr. Detlef Weigel

Abstract

Recent advances in sequencing technologies have made it possible for the first time to sequence and analyse the genomes of whole populations of individuals in both a cost-effective manner and in a reasonable amount of time. One of the primary applications of this data is to better understand and investigate the genetic basis of common traits or diseases. For this purpose, genome-wide association studies (GWASs) are often used to find loci that are associated with a phenotype of interest. However, conducting GWASs is a challenging endeavour: first, different types of hidden confounding factors, such as population structure, environmental or technical influences, could lead to spurious associations. Second, it has been shown in several studies that associated loci often fail to explain much of the phenotypic variability — a phenomenon referred to as the problem of *missing heritability*. Many tools have been developed to partly address these challenges. The large diversity of these tools, however, have led to a highly fragmented and confusing landscape of tools. In addition, most of these tools do not share a common data format and do not provide straightforward solutions to visualise and annotate their results.

In this thesis, we aim to explain more of the missing heritability, while simultaneously simplifying the usage of different methods, by providing an integral solution for performing, visualising and annotating GWASs. Therefore, we develop **easyGWASCore**, an integrated framework for performing GWASs and meta-analyses. Our framework facilitates the use of popular association methods by providing a common data structure, an application programming interface and a `Python` command line interface. In addition, **easyGWASCore** offers an out-of-the-box visualisation and annotation pipeline. We compare the runtime of the **easyGWASCore** framework to other well-established tools and find that it is at least as efficient as the individual software tools.

Next, we enrich the **easyGWASCore** annotation pipeline with pathogenicity prediction scores to prioritise associated loci for further biological investigation, as well as to narrow down potentially causal loci. However, a large variety of such pathogenicity prediction tools exists and it is not obvious which of these tools work best. We therefore investigate the question whether there are systematic differences in the quality of the predictive performance of pathogenicity prediction tools when evaluated on a large number of variant databases. We find that the evaluation is hindered by two types of circularity and that these types of circularity might lead to spurious biological interpretation. Hence, it is important that scientists are aware of these different types of circularity when pathogenicity prediction tools are used for further experiments or analyses.

Increasing sample sizes and combining the results of several GWASs could help to explain parts of the missing heritability. For this purpose, we develop a cloud and web-service, called **easyGWAS**, to provide a platform to share and publish data and results of GWASs and meta-analyses in a straightforward manner. Simultaneously, **easyGWAS** facilitates the use of the **easyGWASCore** framework by providing an easy-to-

use step-by-step procedure to conduct different types of GWASs and meta-analyses via a web-browser. In addition, **easyGWAS** offers dynamic visualisations and annotations of GWAS results to obtain more detailed information about specific regions.

The joint effect of multiple loci could also help to explain parts of the missing heritability. However, multi-locus methods that focus on multiplicative effects are often unfeasible to compute for genome-wide settings and methods that focus on additive effects are often hard to interpret. We here develop a novel method that is able to integrate biological networks as prior knowledge to guide the detection of sets of genetic markers that are maximally associated with a given phenotype. Furthermore, we show how this framework can be extended to multiple correlated traits. Both methods are integrated into the **easyGWASCore** framework. We find that they have improved abilities to discover novel genetic loci and are able to account for parts of the missing heritability by explaining larger proportions of the phenotypic variance than univariate association testing methods.

Finally, we demonstrate the full potential of the **easyGWASCore** framework by conducting a comprehensive study in the model organism *Arabidopsis thaliana*. Here, we investigate the effect of non-additive genetic variance on hybrid phenotypes in *Arabidopsis thaliana* and characterise the contribution of dominance to heterosis — that is the phenotypic superiority of progeny of a cross relative to their genetically distinct parents — as a potential source of missing heritability. For this purpose, we utilise the **easyGWASCore** framework to conduct different GWASs using a univariate method, as well as our novel network guided multi-locus approach. Subsequently we use the visualisation and annotation pipeline to investigate significantly associated regions in more detail. Our results suggest that non-additive effects might be an important source of information that could help to explain parts of the missing heritability.

In summary, the **easyGWASCore** framework and **easyGWAS** cloud service are two novel approaches that help to explain more of the missing heritability, while simultaneously simplifying the process of conducting, analysing and managing such studies.

Zusammenfassung

Die jüngsten Fortschritte in der Sequenzierungstechnologie ermöglichen es erstmalig, komplette Genome ganzer Populationen in angemessener Zeit und kosteneffizient zu sequenzieren. Eine der primären Anwendungen dieser Daten ist es, die genetischen Ursachen von häufig auftretenden phänotypischen Merkmalen oder Krankheiten besser zu verstehen. Im Wesentlichen werden hierzu genomweite Assoziationsstudien (GWASs) verwendet, um damit Positionen im Genom zu finden, welche mit einem Phänotyp assoziiert sind. GWASs durchzuführen, ist jedoch ein herausforderndes Unterfangen. Zum Ersten können verschiedene Arten von versteckten Störfaktoren, wie beispielsweise Populationsstrukturen, umweltbedingte oder technische Einflüsse zu unechten Assoziationen führen. Zum Zweiten wurde in unterschiedlichen Studien nachgewiesen, dass assoziierte Positionen im Genom nur zum Teil die phänotypische Varianz erklären können. Dieses Phänomen wird oft als das Problem der fehlenden Heritabilität (Vererbbarkeit) bezeichnet. Eine Vielzahl an Tools wurde entwickelt, um diese Herausforderungen teilweise zu adressieren. Die große Vielfalt dieser Anwendungen führt jedoch zu einer stark fragmentierten Landschaft dieser Tools. Darüber hinaus besitzen die meisten dieser Tools kein einheitliches Datenformat und bieten keine unkomplizierten Lösungen an, um deren Ergebnisse zu visualisieren oder zu annotieren.

Das Ziel dieser Arbeit ist es, einen größeren Anteil der fehlenden Heritabilität zu erklären und gleichzeitig die Verwendung verschiedener Methoden zu vereinfachen, indem wir eine kombinierte Lösung zum Durchführen, Visualisieren und Annotieren von GWASs anbieten. Demzufolge haben wir **easyGWASCore**, ein kombiniertes Framework zum Durchführen von GWASs und Metaanalysen entwickelt. Unser Framework erleichtert die Verwendung von gängigen Methoden zum Testen von Assoziationen, indem eine gemeinsame Datenstruktur, eine Programmierschnittstelle und eine `Python` Kommandozeilenschnittstelle zur Verfügung gestellt wird. Zusätzlich bietet **easyGWASCore** eine integrierte Visualisierungs- und Annotationspipeline. Wir haben die Laufzeit des **easyGWASCore** Frameworks mit anderen etablierten Tools verglichen und fanden heraus, dass es mindestens so effizient ist wie diese einzelnen Software-Tools.

Als Nächstes haben wir die **easyGWASCore** Annotationspipeline mit Vorhersagen über die Pathogenität von Proteinen erweitert, um assoziierte Positionen im Genom zu priorisieren sowie potentielle kausale Positionen einzuengen. Jedoch gibt es eine große Anzahl solcher Anwendungen zur Pathogenitätsvorhersage und es ist nicht offensichtlich, welches dieser Tools am besten funktioniert. Wir haben demzufolge die Frage untersucht, ob es systematische Unterschiede in der Vorhersagequalität dieser Pathogenitätsvorhersage-Tools gibt. Wir haben herausgefunden, dass die Evaluierung durch zwei verschiedene Arten von Zirkularität gehindert wird und dass diese Arten der Zirkularität zu biologischen Missinterpretationen führen können. Folglich ist es wichtig, dass Wissenschaftler diese Arten der Zirkularität kennen, wenn Anwendungen zur Pathogenitätsvorhersage für weitere Experimente und Analysen verwendet werden.

Eine wachsende Anzahl an Stichproben und die Kombination der Ergebnisse mehre-

rer GWASs können dabei helfen, Teile der fehlenden Heritabilität zu erklären. Daher haben wir den Cloud- und Web-Dienst **easyGWAS** entwickelt, eine Plattform, um unkompliziert Daten und Ergebnisse von GWASs und Metaanalysen zu teilen und zu publizieren. Gleichzeitig vereinfacht **easyGWAS** die Verwendung des **easyGWASCore** Frameworks, indem es ein einfach zu verwendendes Schritt-für-Schritt-Verfahren anbietet, um verschiedene Arten von GWASs und Metaanalysen im Internetbrowser durchzuführen. Zusätzlich bietet **easyGWAS** dynamische Visualisierungs- und Annotationsfunktionen, um detailliertere Informationen über bestimmte Regionen zu erhalten.

Der gemeinsame Effekt von multiplen Positionen im Genom könnte ebenso dazu beitragen, Teile der fehlenden Heritabilität zu erklären. Jedoch sind Methoden, welche auf multiplikative Effekte zwischen mehreren Positionen im Genom ausgerichtet sind, oft nicht berechenbar für genomweite Untersuchungen. Des Weiteren sind Methoden, welche auf additive Effekte von mehreren Positionen im Genom ausgerichtet sind, oft schwer zu interpretieren. Wir haben daher eine neuartige Methode entwickelt, in welche wir bekanntes biologisches Vorwissen in Form von biologischen Netzwerken integrieren können, um dann multiple Positionen im Genom zu identifizieren, welche maximal mit einem Phänotypen assoziiert sind und innerhalb dieses Netzwerkes verbunden sind. Zusätzlich haben wir gezeigt, wie diese Methode für mehrere korrelierte Phänotypen erweitert werden kann. Beide Ansätze wurden in das **easyGWASCore** Framework integriert. Wir haben herausgefunden, dass beide Methoden verbesserte Fähigkeiten zeigen, genetische Marker zu entdecken sowie verbesserte Fähigkeiten, Teile der fehlenden Heritabilität zu erklären, indem größere Anteile der phänotypischen Varianz erklärt werden können als mit univariaten Methoden zur Assoziationssuche.

Letztendlich demonstrieren wir das gesamte Potential des **easyGWASCore** Framework anhand einer umfassenden Studie in dem Modellorganismus *Arabidopsis thaliana*. Hier haben wir den Effekt von nicht-additiver genetischer Varianz von Phänotypen in Hybriden *Arabidopsis thaliana* Individuen untersucht und den Beitrag von Dominanz auf Heterosis als eine mögliche Quelle fehlender Heritabilität charakterisiert. Heterosis ist die phänotypische Überlegenheit einer Kreuzung verglichen zu den genetisch unterschiedlichen Eltern. Aus diesem Zweck haben wir das **easyGWASCore** Framework verwendet, um verschiedene GWASs mit univariaten Methoden durchzuführen. Des Weiteren haben wir unseren neuartigen Ansatz zur netzwerkunterstützten Suche von multiplen Positionen im Genom verwendet. Anschließend wurde die Visualisierungs- und Annotationspipeline verwendet, um signifikant assoziierte Regionen im größeren Detail zu untersuchen. Unsere Ergebnisse deuten darauf hin, dass nicht-additive Effekte eine wichtige Quelle sind, um Teile der fehlenden Heritabilität zu erklären.

Zusammenfassend haben wir mit dem **easyGWASCore** Framework und dem Cloud basierten Dienst **easyGWAS** neuartige Ansätze entwickelt, welche dabei helfen, Teile der fehlenden Heritabilität zu erklären. Gleichzeitig haben wir den Prozess vereinfacht, solche Studien durchzuführen, zu analysieren und zu managen.

Acknowledgements

Above all, I would like to thank my supervisor Prof. Dr. Karsten Borgwardt for his excellent supervision and advice during my time as a PhD student in his lab on a professional, as well as on a personal level. Especially, I am greatly thankful for his untiring efforts, continuous support and availability during my PhD. In addition, I would like to thank him for his time and comments during the writing of this thesis.

The Machine Learning and Computational Biology Research Group, led by Prof. Dr. Karsten Borgwardt, was an interdisciplinary research group, affiliated with the Max Planck Institute for Developmental Biology and the Max Planck Institute for Intelligent Systems (now at ETH Zürich). Thus, I also could learn many important and state-of-the-art concepts from leading scientists in Biology and Machine Learning.

Herewith, I would like to thank Prof. Dr. Detlef Weigel from the Max Planck Institute for Developmental Biology, for his excellent support and the close collaborations on many highly interesting and cutting edge research projects. Due to this close collaboration and his advise I could broaden my knowledge in Biology which helped to improved my own research. Furthermore, I would like to thank Prof. Dr. Detlef Weigel for being the second reviewer of my thesis.

I also would like to thank Prof. Dr. Bernhard Schölkopf from the Max Planck Institute for Intelligent Systems for his advise and for hosting the `easyGWAS` web-application at his institute.

I would like to thank Prof. Dr. Oliver Kohlbacher and PD. Dr. Kay Nieselt for agreeing to be part of my PhD defence committee.

In addition, I would like to thank all my collaborators, co-authors and colleagues including, Prof. Dr. Karsten Borgwardt, Prof. Dr. Detlef Weigel, Prof. Dr. Bernhard Schölkopf, Prof. Dr. Mahito Sugiyama, Prof. Dr. Yoshinobu Kawahara, Prof. Dr. Jordan Smoller, Prof. Dr. Aasa Feragen, Prof. Dr. Mark Daly, Prof. Dr. Daniel MacArthur, Dr. Chloé-Agathe Azencott, Dr. Barbara Rakitsch, Dr. Damian Roqueiro, Dr. Daniel Koenig, Dr. Angela McGaughran, Dr. Christian Rödelsperger, Dr. Oliver Stegle, Dr. Christoph Lippert, Dr. Beth Rowan, Dr. Laramie Duncan, Dr. Dean Bodenham, Danelle Seymour, Felipe Llinares-López, Udo Gieraths, Stefan Kleeberger and many more.

Next, I would like to thank my former colleague and friend Dr. Chloé-Agathe Azencott for all our close collaborations on several research projects and her untiring efforts reading through all my manuscripts and especially for proofreading parts of my thesis. A special thank goes to my former colleague and close friend Dr. Barbara Rakitsch for all our professional and private discussions on various different research projects and topics, her untiring help in explaining me different concepts and methods, as well as proofreading parts of my thesis.

Furthermore, I would like to thank Dr. Damian Roqueiro for our great scientific discussions and his steady help with all administrative related duties in our group, as well as for his efforts reading and commenting on my thesis.

I am sincerely grateful to all who read parts of this thesis and commented on it, including Damian Roqueiro, Barbara Rakitsch, Chloé-Agathe Azencott, Felipe Llinares-López, Andrea Schuster, Dean Bodenham, Thomas Grimm and Gerhard Grimm.

During my studies and for many projects I was always supported by the administrative and technical team. In particular I would like to thank Sebastian Stark and Johannes Woerner for technical support, especially for setting up the servers for **easyGWAS**. I also would like to thank Hülya Wicher, Sabriana Rehbaum, Julia Braun and Jürgen Apfelbacher for their great administrative help. I also acknowledge the Max Planck Society (Max-Planck-Gesellschaft) for funding my PhD.

I cordially thank all my friends for their support and all the small welcome distractions that constantly motivated me to finish my PhD.

Finally, I want to thank my family, my parents Waltraud and Gerhard Grimm, my brothers, Thomas, Jakob and Christoph Grimm, as well as my beautiful and lovely girlfriend Andrea, for all their love and support. Especially, I would like to thank Andrea for constantly supporting and motivating me during writing up this thesis.

Contents

1	Introduction	1
1.1	Genome-Wide Association Studies	1
1.1.1	The Problem of the Missing Heritability	3
1.1.2	Population Stratification and Hidden Confounding	4
1.1.3	Data Sharing and Privacy	4
1.1.4	Algorithmic, Technical and Infrastructural Challenges	5
1.1.5	Representation and Annotation of Results	6
1.2	Objectives and Contributions of this Thesis	6
1.2.1	An Integrated Framework for GWASs and Meta-Analyses	7
1.2.2	An Extended Annotation Pipeline to Prioritise Associated Loci	7
1.2.3	A Cloud Service for GWASs and Meta-Analyses	8
1.2.4	Improving GWASs by Incorporating Biological Networks as Prior Knowledge	9
1.2.5	Case Study: Non-Additive Components of Genetic Variations in <i>Arabidopsis thaliana</i>	10
2	An Integrated Framework for Performing Genome-Wide Association Studies	13
2.1	Regression Based Methods for GWASs	14
2.1.1	Linear Regression	14
2.1.2	Logistic Regression	17
2.1.3	Linear Mixed Models	19
2.2	Hypothesis and Multiple Testing	22
2.2.1	Hypothesis Testing for Regression Methods	22
2.2.2	Multiple Hypothesis Testing	23
2.3	Meta-Analysis Methods for GWASs	27
2.3.1	Fisher’s Method	27
2.3.2	Stouffer’s Z	28
2.3.3	Fixed Effect Model for Meta-Analysis	28
2.3.4	Random Effect Model for Meta-Analysis	30

2.4	easyGWASCore: An Efficient C/C++ Framework for GWASs and Meta-Analyses	33
2.4.1	The Architecture and Design of easyGWASCore	33
2.4.2	The easyGWASCore Application Programming Interface	36
2.4.3	The Python Command Line Interface of easyGWASCore	40
2.4.4	Performance Analysis	44
2.5	Chapter Summary	45
3	Pathogenicity Prediction Scores as Additional Source for Annotation	47
3.1	A Comprehensive Analysis of Pathogenicity Prediction Tools	48
3.1.1	Experimental Settings	48
3.1.2	Results	50
3.1.3	Guidelines to Avoid Different Types of Circularity	58
3.2	Adding Pathogenicity Prediction Scores to easyGWASCore	59
3.3	Chapter Summary	62
4	A Cloud Service for Genome-Wide Association Studies	63
4.1	Architectural and Technical Details	64
4.2	Overview of the easyGWAS web-application	66
4.2.1	The easyGWAS Data Repository	66
4.2.2	The easyGWAS GWAS Centre	69
4.3	Publicly Available Data	77
4.4	Case Study in <i>Arabidopsis thaliana</i>	78
4.5	Chapter Summary	81
5	Network Guided Multi-Locus and Multi-Trait Association Mapping	83
5.1	SConES: Selecting Connected Explanatory SNPs	84
5.1.1	Method and Problem Formulation	84
5.1.2	Feature Selection with Graph Regularisation	87
5.1.3	Min-Cut Solution	88
5.1.4	Experimental Settings	90
5.1.5	Results	93
5.2	Multi-SConES	100
5.2.1	Multi-Task Formulation	100
5.2.2	Experimental Settings	102
5.2.3	Results	104
5.3	easyGWASCore Integration	108
5.3.1	Data Processing and Algorithmic Runtime Analysis of SConES	109
5.3.2	Runtime Comparison Between Different Implementations	109
5.3.3	Runtime Comparison of SConES Including a Grid-search	110
5.3.4	Runtime Comparison of Multi-SConES	111
5.4	Chapter Summary	111

6	Non-Additive Components of Genetic Variations in <i>Arabidopsis thaliana</i>	113
6.1	Data Generation and Preparation	114
6.1.1	Generation of Plant Material	114
6.1.2	Plant Phenotyping	116
6.1.3	F_1 Genotype Data Generation for GWASs	116
6.1.4	Phenotype Data Preparation for GWASs	118
6.2	Methods and Experimental Settings	119
6.2.1	Heritability Estimation Based on Family Data	119
6.2.2	Genome-Wide Association Mapping	120
6.2.3	GWAS Visualisations and Annotations	121
6.2.4	Estimation of Variance Explained	122
6.2.5	Power Analysis	122
6.3	Results	123
6.3.1	Phenotypic Analysis Based on Family Data	123
6.3.2	Association Mapping of Phenotypic Components	124
6.3.3	Analysis of Variance Explained	126
6.3.4	Simulation of Phenotypes and Power-Analysis	127
6.4	Chapter Summary	128
7	Conclusions and Outlook	131
A	Nomenclature	139
B	Performance Evaluation Statistics	141
C	General GWAS related terminology	143
C.1	Minor Allele Frequency	143
C.2	Genotype Encoding	143
D	easyGWASCore API and Python Command Line Interface Overview	145
D.1	The Application Programming Interface	145
D.2	The easyGWASCore Command Line Python Interface	150
	List of Figures	153
	List of Tables	159
	Bibliography	161

CHAPTER 1

Introduction

1.1 Genome-Wide Association Studies

The sequencing of medium- to large-sized genomes was an expensive and cumbersome enterprise in the late 20th and the early 21st centuries. One of the most popular technologies at this time was the so called Sanger sequencing. For the sequencing of a whole mammal genome, such as the first complete human reference genome [*International Human Genome Sequencing Consortium et al.*, 2004], large sequencing centres and hundreds of scientists were needed. Hence, there was a high demand for inexpensive high-throughput sequencing technologies. Great advances in this field led to the development of new sequencing machines, also referred to as Next Generation Sequencing (NGS) machines, and enabled researchers to cheaply generate billions of short sequences (*reads*) within days [*Metzker*, 2010]. However, this vast amount of new data led to computational problems and bottlenecks. Efficient alignment and mapping algorithms were needed to reconstruct the original sequence from all these short reads, as well as sophisticated methods to detect arbitrary types of structural variations (SVs). Indeed, several achievements over the last years in the development of novel alignment and mapping algorithms [*Buchfink et al.*, 2014; *Li and Durbin*, 2009; *Li et al.*, 2009c; *Ossowski et al.*, 2008], as well as advances in algorithms to identify SVs such as Single Nucleotide Polymorphisms (SNPs) [*DePristo et al.*, 2011; *Li et al.*, 2009a,b; *McKenna et al.*, 2010; *Ossowski et al.*, 2008], Insertions and Deletions (InDels) [*DePristo et al.*, 2011; **Grimm et al.**, 2013; *Lee et al.*, 2008; *Medvedev et al.*, 2009; *Tuzun et al.*, 2005; *Ye et al.*, 2009] or other types of variations, such as tandem duplications and Copy Number Variations (CNVs), laid the foundation for the systematic analysis of this enormous amount of sequencing data. For the first time it was possible to sequence and analyse whole populations of individuals in a cost-effective manner and in a reasonable amount of time, such as done in the initial phase of the 1000 Human Genomes project [*1000 Genomes Project Consortium et al.*, 2010] or in the 1001 Genomes project in *Arabidopsis thaliana* [*Cao et al.*, 2011]. The recent advances in NGS technologies and the latest sequencing efforts led to unprecedented detailed maps of structural variations at

a genome-wide level [1000 Genomes Project Consortium *et al.*, 2010; Cao *et al.*, 2011]. Understanding the genetic basis and mechanisms that lead to heritable variations has been a central aim for geneticists for already more than a century. Thus, these latest sequencing efforts enabled researchers to better understand the genetic basis of common traits and diseases by investigating whether any of these SVs are associated with a certain trait at a genome-wide level [Burton *et al.*, 2007; McCarthy *et al.*, 2008]. For this purpose, Genome-Wide Association Studies (GWASs) and are used as an integral tool to better understand and investigate the genetic basis of common traits [McCarthy *et al.*, 2008]. Usually, SNPs are used as genetic markers for GWASs. In general an association is a correlation between the allelic and the phenotypic differences of a cohort of independent individuals. In Figure 1.1 we give a toy example of a GWAS in which we search for SNPs that are highly correlated with a binary phenotype of plant flowers: colour yellow vs. colour blue. The term *phenotype* is quite general and can



Figure 1.1: Illustration of a genome-wide association study: Simple illustration of a GWAS using a binary plant phenotype (plant flowers yellow vs. plant flowers blue) and three SNPs. The green SNP is associated with the phenotype whereas the others are not.

be either an apparent characteristic (e.g. the height of a plant or human eye colour) or any quantifiable characteristic, such as having a disease or being a non-responder to a certain drug. Thus, phenotypes can be binary/dichotomous (e.g. in case-control studies), categorical (e.g. different treatment groups) or continuous measurements (e.g. height of a person or plant).

Due to biological events, such as recombination, genetic drifts, selection or mutation, a non-random correlation between alleles at different loci in close proximity is created — this non-random correlation is also referred to as linkage disequilibrium (LD) [Hartl *et al.*, 1997; Reich *et al.*, 2001; Visscher *et al.*, 2012]. Thus, the likelihood that loci are inherited together — and thus linked to each other — decreases with their physical distance [Visscher *et al.*, 2012]. This leads to an important point: an association between a genetic marker and a phenotype is not necessarily a causal relationship. This is because, SNPs that are used for association testing might only be linked to causal variants — if there are any causal variants at all. In Figure 1.2, we illustrate an example with one causal variant that is not sequenced but that is still linked to an

associated SNP. Some studies exploit this property of LD to create well-design SNP



Figure 1.2: Illustration of an association with a non-causal SNP: Here, the causal SNP (red) was not sequenced. However, an indirect association with a linked SNP (blue) can be observed.

chip arrays (e.g. Affymetrix[®] SNP array) to tag SNPs that are in functional regions or in close proximity to the causal one (e.g. upstream and downstream of a gene) [Atwell *et al.*, 2010; Burton *et al.*, 2007].

Until 2013, approximately 2,000 robust associations have been identified for more than 300 complex traits and phenotypes in human GWASs [Manolio, 2013], including novel associations for traits like human height [Visscher, 2008; Yang *et al.*, 2010] or diseases such as type 2 diabetes [Scott *et al.*, 2007], migraine [Chasman *et al.*, 2011; Freilinger *et al.*, 2012], chronic obstructive pulmonary disease (COPD) [Pillai *et al.*, 2009], Crohn’s disease [Rioux *et al.*, 2007] or schizophrenia [Schizophrenia Psychiatric Genome-Wide Association Study (GWAS) Consortium *et al.*, 2011]. GWASs have also been successfully applied to other species, such as *Arabidopsis thaliana* [Atwell *et al.*, 2010; Filiault and Maloof, 2012; Meijón *et al.*, 2014], *Oryza sativa indica* [Zhao *et al.*, 2011] or *Drosophila melanogaster* [Mackay *et al.*, 2012].

1.1.1 The Problem of the Missing Heritability

Despite the apparent success of these GWASs, it has been shown that more than 80% of all identified variants are found in non-coding regions and that many of these identified variants often fail to explain much of the phenotypic variability — the latter phenomenon is often referred to as the problem of “missing heritability” [Manolio *et al.*, 2009]. A prominent example is the quantitative trait “human height”. As reported in Visscher [2008], variations in human height within a population have an estimated empirical heritability of approximately 80%. Although more than 50 loci associated with human height have been detected in several studies [Gudbjartsson *et al.*, 2008; Lettre *et al.*, 2008; Weedon *et al.*, 2008], they together account only for approximately 5% of the phenotypic variance [Manolio *et al.*, 2009; Visscher, 2008]. Many theories have been suggested to explain parts of this missing heritability. One suggestion is to also include variants with weak or small effect sizes that could not be detected to be significantly associated with a standard univariate test — that is the test of a single loci at a time [Manolio *et al.*, 2009]. Indeed, Yang *et al.* [2010] found that 45% of the phenotypic variance in human height can be explained when including all commonly available SNPs. Hence, small effect sizes could lead to a lack of statistical power or the lack to replicate certain GWASs in an independent population [Seng and Seng, 2008].

The same is true for variations that can only be found in a minority of individuals, so-called rare variations. Furthermore, investigating effects of multiple genetic markers simultaneously, by considering additive or interactive effects, might also contribute to explain parts of this missing heritability [Marchini *et al.*, 2005]. Searching for new strategies and developing new methods to explain more of this missing heritability is therefore of utmost importance and has been evolved to an active research area over the past 10 years.

1.1.2 Population Stratification and Hidden Confounding

An additional challenge in modern GWASs are hidden confounding factors, such as population stratification, environmental or technical influences [Buettner *et al.*, 2015; Listgarten *et al.*, 2010; Novembre *et al.*, 2008; Price *et al.*, 2006]. For example, a general assumption in traditional genome-wide association studies is that the phenotypes are independently distributed across cohorts [Flint and Eskin, 2012]. However, this independence criterion is violated due to complex genetic relationships, which means that related individuals will have more similar phenotypes than more distant individuals [Flint and Eskin, 2012]. In other words, a set of SNPs might have a similar structure as the phenotype. This relationship leads to structured data which in turn might lead to spurious and false interpretations of these associations. Thus, it is important to properly account for relatedness in structured data and other types of hidden confounding. Mixed models are a prominent class of methods that can be used to account for population stratification in univariate GWASs [Kang *et al.*, 2008, 2010; Lippert *et al.*, 2011; Listgarten *et al.*, 2013]. To also address the missing heritability problem, multi-locus models have been proposed to investigate multiple genetic markers simultaneously, while accounting for population structure [Lippert *et al.*, 2013; Rakitsch *et al.*, 2013b; Segura *et al.*, 2012]. In addition, mixed models can also be used to investigate *pleiotropic* effects, that is the effect of a single genetic marker or gene to different correlated phenotypes (multi-trait studies), while at the same time correcting for structured residuals [Korte *et al.*, 2012; Rakitsch *et al.*, 2013a; Solovieff *et al.*, 2013].

1.1.3 Data Sharing and Privacy

The apparent success to infer surnames from anonymised genetic data [Gymrek *et al.*, 2013] led to many discussions about data privacy and about how to share genetic data with other scientists and labs — if sharing is an option at all. Because of concerns about privacy, researchers tend to be overzealous to protect the data and only share summary statistics (e.g. p-values or effect estimates) of the GWASs they conducted. Large genetic consortia consisting of many different labs in various countries have been created to study different types of diseases. These consortia often use a technique called meta-analysis to combine summary results from several independent GWASs conducted at different nodes of the consortia. One advantage of these consortia is that combining the results from several independent GWASs leads to studies with larger sample sizes. Nevertheless, these vast growing amount of samples will lead to a big

computational burden in the near future and thus there will be a need to re-engineer even basic algorithms. A second advantage is that GWASs can be performed at the individual nodes without the need to share the raw genetics data with others. However, becoming part of an existing consortia is a difficult and cumbersome process and there is no guarantee to succeed with this process at all.

It has been shown that meta-analysis is a powerful technique to (i) increase statistical power, to (ii) reduce the number of false positive associations [Evangelou and Ioannidis, 2013] and to (iii) detect novel associations that could help to explain more of the missing heritability [Evangelou and Ioannidis, 2013; Franke *et al.*, 2010; Nalls *et al.*, 2014; Neale *et al.*, 2010; Pharoah *et al.*, 2013; Ripke *et al.*, 2013; Stahl *et al.*, 2010]. Therefore, it would be of great value to have a way to conduct GWASs on different datasets without the need to grant full access rights to the raw genetic data.

1.1.4 Algorithmic, Technical and Infrastructural Challenges

More and more individuals can be sequenced in less time due to vast advances in the development of NGS technologies and the cheaper costs of sequencing. This excessive growth of datasets, as well as the steady increase in the number of GWASs in humans and other organisms over the last few years [Atwell *et al.*, 2010; Filiault and Maloof, 2012; Mackay *et al.*, 2012; Manolio, 2013; Meijón *et al.*, 2014; Zhao *et al.*, 2011], led to many algorithmic, technical and data management challenges. Datasets with several thousand individuals and millions of markers need efficient algorithms even for simple tasks, such as searching for univariate associations. This problem becomes even more intense when searching for loci that jointly contribute to the phenotypic variance. Improving algorithms to scale to large datasets is currently an active research field. Some algorithms for medium-sized datasets (hundreds to thousands of samples and millions of SNPs) have already been proposed by several researchers [Kang *et al.*, 2010; Lippert *et al.*, 2011; Rakitsch *et al.*, 2013a,b]. However, most of these algorithms will not scale to hundreds of thousands of individuals.

An additional important point is that most of the available algorithms neither share a common data input or output format nor an easy way to access publicly available data. Although this seems to be a minor problem, it has extensive impacts on the correctness, productivity and management of large GWAS projects. First, publicly available data sources, especially for non-human species, are often scattered over different websites. For human data, ethic approvals and detailed research proposals are usually needed. Thus, collecting or getting access to those datasets is often a cumbersome and time-consuming task and requires a high degree of organisation. Secondly, converting these datasets between various data formats for different algorithms could easily lead to errors and thus to wrong biological interpretations. Also, technical problems emerge when reliably storing, backing up, handling or sharing large volumes of data and results. Thus, advanced technical background knowledge in server architectures, server configurations and data storage are a tremendous advantage to successfully organise and lead large GWAS projects.

1.1.5 Representation and Annotation of Results

Eventually, visualising, annotating and interpreting results from GWASs is an additional tedious and labour-intensive step. Most available tools mainly focus on the association mapping part but leave the users with massive and confusing result files. However, the keys for a successful GWAS are the annotation and interpretation of the results, as well as clear and intuitive visualisations. The number of tools to visualise result files from mapping algorithms is limited. For example, to create basic visualisations for output files from PLINK [Purcell *et al.*, 2007] — a popular collection of algorithms for genome-wide association analyses — the tool Haploview [Barrett *et al.*, 2005] or custom Python, R or Matlab scripts could be used. Unfortunately, these visualisations are static and do not allow the user to dynamically interact with them, such as zooming into interesting regions, dynamically changing the multiple hypothesis correction method or retrieving additional information about regions of interest.

Also, retrieving automatic annotations for significantly associated hits is linked to labour-intensive extra steps. Third party tools, such as snpEFF [Cingolani *et al.*, 2012] or SIFT 4G¹ [Vaser *et al.*, 2015], could be used to retrieve additional annotations for regions of interest. The tool snpEFF [Cingolani *et al.*, 2012] annotates SNPs based on their genomic position and predicts a potential effect of a given variant, including whether this variant is located within a gene or if the variants leads to an amino acid change. In addition, SIFT 4G [Vaser *et al.*, 2015] predicts whether the change of an amino acid might lead to a potential damaging or pathogenic effect of the protein. These additional information might help to better interpret results and to narrow down interesting hits or regions for further biological investigation.

1.2 Objectives and Contributions of this Thesis

As set out above, GWASs are a highly complex field with many challenges that have to be addressed and solved. Conducting a whole GWAS from the beginning till the end requires a variety of different steps that have to be combined similarly to the pieces of a puzzle (Figure 1.3). The aim of this thesis is to contribute to these different *puzzle* pieces by developing methods and tools that help to explain parts of the missing heritability. These different *puzzle* pieces can also be combined in an integral tool and framework to facilitate the process of conducting and managing GWASs by bringing together the storage, handling and sharing of data with the analysis, representation and annotation. In the following sections we will give a brief overview about the individual chapters of this thesis and our published manuscripts, as well as the individual contributions.

¹<http://siftdb.org>

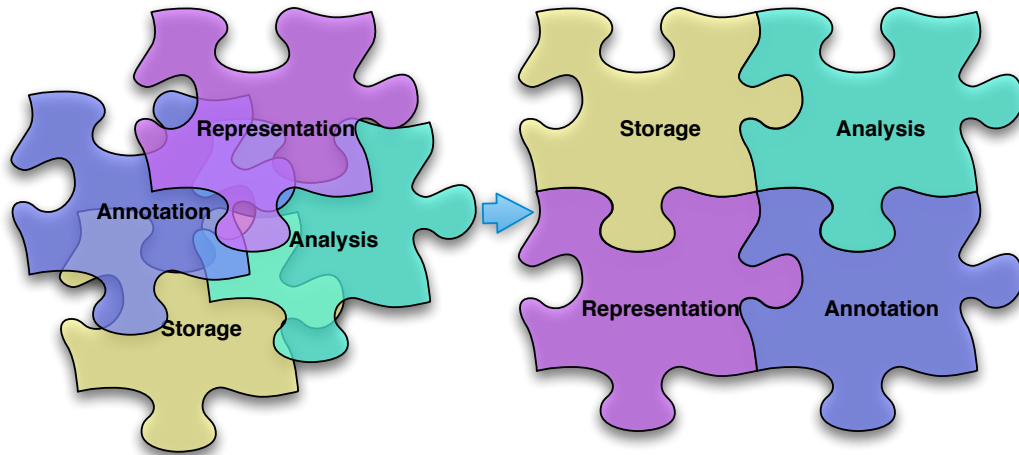


Figure 1.3: Combing the different puzzle pieces of a GWAS: To successfully perform a complete GWAS many different pieces have to be combined. The objective of this thesis is to contribute to different methods to explain larger parts of the missing heritability while facilitating the process of conducting a GWAS.

1.2.1 An Integrated Framework for GWASs and Meta-Analyses

In this thesis we aim to explain more of the missing heritability, while simultaneously facilitating the usage of different methods for GWASs and meta-analyses. However, due to the fragmented and diverse landscape of these tools, a first objective of this thesis is to develop an integrated framework and Application Programming Interface (API) consisting of different popular algorithms for performing GWASs and meta-analysis. The framework should simultaneously simplify the process of data handling and management. The framework is called `easyGWASCore` and is developed in C/C++ with Python interfaces.

In Chapter 2 we therefore introduce and summarise popular regression based models for GWASs. Next we introduce a technique to obtain a measure of statistical significance between two regression models and discuss several multiple hypothesis correction methods. In addition, we give a brief overview about four popular meta-analysis models to combine the results from several conducted GWASs. Eventually, we describe the `easyGWASCore` framework and analyse the performance by comparing it to established and widely used tools.

Publications and Individual Contributions: Chapter 2 is based on the following unpublished work:

- **Dominik G Grimm** and Karsten M Borgwardt. A C/C++ Framework with Python Interfaces for Genome-Wide Association Studies. *Unpublished*, 2015

Dominik Grimm developed the framework, API, performed the experiments, analysed the data and wrote the text.

1.2.2 An Extended Annotation Pipeline to Prioritise Associated Loci

The number of associated candidate loci and regions found by different methods can be large. Annotating and interpreting these loci is cumbersome and often not possible

without additional validations and extensive biological experiments. Therefore, it is important to prioritise associated loci before further biological investigation. Different *in silico* tools can be used to retrieve more knowledge about these variants, such as the effect of a given SNP. SNPs that lead to an amino acid change, so called missense variants, could be used to predict whether this variant leads to a damaging or pathogenic effect on the protein. However, due to the wealth of such pathogenicity prediction tools, an important practical question to answer is which of these tools generalise best, that is, correctly predicts the pathogenic character of a given variant.

In Chapter 3, we comprehensively evaluate a selection of ten pathogenicity prediction tools. Eventually, we enriched the **easyGWASCore** annotation pipeline with pathogenicity prediction scores to prioritise loci in a certain region for further biological investigation and to narrow down potential causal loci.

Publications and Individual Contributions: Parts of the introduction, methods and results in Chapter 3 are based on the following publication:

- **Dominik G Grimm**, Chloé-Agathe Azencott, Fabian Aicheler, Udo Gieraths, Daniel G MacArthur, Kaitlin E Samocha, David N Cooper, Peter D Stenson, Mark J Daly, Jordan W Smoller, Laramie E Duncan, and Karsten M Borgwardt. The Evaluation of Tools Used to Predict the Impact of Missense Variants Is Hindered by Two Types of Circularity. *Human mutation*, 36(5):513–523, 2015

Dominik Grimm, Chloé-Agathe Azencott, Jordan Smoller, Laramie Duncan and Karsten Borgwardt conceived the study. Dominik Grimm implemented the analysis pipeline. Dominik Grimm performed the data preprocessing with contributions from Fabian Aicheler and Udo Gieraths. Dominik Grimm performed the experiments and created the figures. Dominik Grimm, Chloé-Agathe Azencott, Laramie Duncan and Karsten Borgwardt analysed the data. Kaitlin Samocha, Mark Daly, Daniel MacArthur, David Cooper and Peter Stenson provided data as well as feedback to biological annotations. Dominik Grimm, Chloé-Agathe Azencott, Laramie Duncan and Karsten Borgwardt wrote the paper with contributions from all authors.

1.2.3 A Cloud Service for GWASs and Meta-Analyses

In Chapter 4 we introduce **easyGWAS**, a cloud- and web-service for performing, visualising and annotating genome-wide association and meta-studies. **easyGWAS** is a platform to share data and results from GWASs or to make them publicly available in a straightforward manner. At the same time **easyGWAS** facilitates the usage of the **easyGWASCore** framework through easy-to-use graphical step-by-step procedures to conduct GWASs and meta-analysis in a web-browser. The web-application provides dynamic visualisation and annotation functions to gain deeper insights about specific regions of interest. As a whole, **easyGWAS** should serve the community through easy data access, validation, production, reproduction and sharing of GWASs.

In the first section of Chapter 4 we describe the technical details of **easyGWAS**. The

second section gives an overview about its different functions and views, as well as an detailed description of the graphical step-by-step procedure to create new GWASs or meta-studies. Finally, we apply **easyGWAS** to conduct a case-study in *Arabidopsis thaliana*.

The web application is accessible at: <https://easygwas.tuebingen.mpg.de>

Publications and Individual Contributions: A journal publication of this work is in preparation. Parts of this chapter are based on the following preprint:

- **Dominik G Grimm**, Bastian Greshake, Stefan Kleeberger, Christoph Lippert, Oliver Stegle, Bernhard Schölkopf, Detlef Weigel, and Karsten M Borgwardt. **easyGWAS: An integrated interspecies platform for performing genome-wide association studies.** *arXiv preprint arXiv:1212.4788*, 2012

Dominik Grimm and Karsten Borgwardt designed the study. Dominik Grimm developed and implemented the functionality and methods of the web-application with help from Stefan Kleeberger and Bastian Greshake. Dominik Grimm performed the experiments and analysed the data. Dominik Grimm set up the server. Christoph Lippert and Oliver Stegle provided statistical feedback. Detlef Weigel and Bernhard Schölkopf provided biological and methodological advice throughout the project. Detlef Weigel provided relevant data. Bernhard Schölkopf provided infrastructural support for hosting the web-application. Dominik Grimm and Karsten Borgwardt wrote the preprint with input from all authors.

1.2.4 Improving GWASs by Incorporating Biological Networks as Prior Knowledge

The joint effect of multiple loci could also help to explain parts of the missing heritability. Multi-locus methods that focus on multiplicative effects are often unfeasible to compute for a genome-wide setting whereas methods that focus on additive effects are often hard to interpret. Including prior biological knowledge could help to better interpret the results. In Chapter 5 we develop a novel method, called **SConES**, to efficiently discover sets of genetic markers that are maximally associated with a phenotype while being connected in an underlying biological network (e.g. a protein-protein interaction network). In addition, we extend this multi-locus mapping approach to also take into account multiple correlated traits and networks. Furthermore, we evaluate both methods on simulated and real world data. Both methods are integrated into the **easyGWASCore** framework to facilitate the usage of these algorithms. Eventually, we compare the performance of both implementations in **easyGWASCore** to different implementations in **Matlab** and **R**.

Publications and Individual Contributions: Parts of the introduction, method and result sections in Chapter 5 are based on the following publications:

- Chloé-Agathe Azencott, **Dominik Grimm**, Mahito Sugiyama, Yoshinobu Kawahara, and Karsten M Borgwardt. Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics*, 29(13):i171–i179, 2013

Chloé-Agathe Azencott, Dominik Grimm, Yoshinobu Kawahara and Karsten Borgwardt conceived the study. Chloé-Agathe Azencott and Dominik Grimm implemented the `Matlab` version of this method with contributions from Mahito Sugiyama. Dominik Grimm implemented the `C/C++` version for the `easyGWASCore` framework. Dominik Grimm and Chloé-Agathe Azencott performed the experiments. Chloé-Agathe Azencott and Dominik Grimm analysed the data. Chloé-Agathe Azencott, Dominik Grimm and Karsten Borgwardt wrote the manuscript with contributions from all authors.

- Mahito Sugiyama, Chloé-Agathe Azencott, **Dominik Grimm**, Yoshinobu Kawahara, and Karsten Borgwardt. Multi-task feature selection on multiple networks via maximum flows. In *Proc. of the 2014 SIAM Int'l Conf. on Data Mining (SDM'14)*, pages 199–207, 2014

Mahito Sugiyama, Chloé-Agathe Azencott, Dominik Grimm, Yoshinobu Kawahara and Karsten Borgwardt conceived the study. Mahito Sugiyama implemented the `R` version of the code. Dominik Grimm implemented the `C/C++` version for the `easyGWASCore` framework. Mahito Sugiyama performed the experiments with contributions from Chloé-Agathe Azencott and Dominik Grimm. Mahito Sugiyama analysed the data. Mahito Sugiyama, Chloé-Agathe Azencott, Dominik Grimm and Karsten Borgwardt wrote the manuscript with contributions from all authors.

1.2.5 Case Study: Non-Additive Components of Genetic Variations in *Arabidopsis thaliana*

Finally, we utilise the `easyGWASCore` framework and demonstrate its full potential by conducting a novel study in the model organism *Arabidopsis thaliana*. Here, we investigate the effect of non-additive genetic variance on hybrid phenotypes in *Arabidopsis thaliana* and characterise the contribution of dominance to heterosis — that is the phenotypic superiority or inferiority of progeny of a hybrid cross relative to their genetically distinct parents — as a potential source of missing heritability. Combining a non-standard genotype encoding with a linear mixed model we are able to identify a number of genomic positions which significantly contribute to non-additive genetic variance. We find that these significantly associated loci account for a large fraction of the total genetic variance. In addition, we show that we can increase the fraction of explained phenotypic variance with a small set of detected loci using the network guided multi-locus mapping approach `SConES`. Eventually, we use the `easyGWASCore` visualisation and annotation pipeline to gain additional information about the pathogenicity status of associated missense variants and its genes.

Publications and Individual Contributions: Parts in Chapter 6 are based on the following work (publication in preparation):

- Danelle K. Seymour, Chae Eunyoung, **Dominik G. Grimm**, Carmen M. Pizzaro, François Vasseur, Barbara Rakitsch, Karsten M. Borgwardt, Daniel Koenig, and Detlef Weigel. The genetic architecture of non-additive hybrid phenotypes in *A. thaliana*. *In Preparation*, 2015

Danelle Seymour, Daniel Koenig, Eunyoung Chae, Dominik Grimm, Karsten Borgwardt and Detlef Weigel designed the research. Danelle Seymour, Eunyoung Chae, Carmen Pizzaro and François Vasseur performed the biological experiments, including plant crossing, growing and phenotyping. Dominik Grimm and Barbara Rakitsch performed the genome-wide association experiments. Dominik Grimm, Danelle Seymour and Barbara Rakitsch performed the data analysis. Danelle Seymour, Daniel Koenig and Detlef Weigel wrote the paper with help and contributions from all authors.

CHAPTER 2

An Integrated Framework for Performing Genome-Wide Association Studies

Many tools and algorithms for performing GWASs and meta-analyses have been developed over the last few years. While some of these tools are collections of several different algorithms for GWASs [Aulchenko *et al.*, 2007; Purcell *et al.*, 2007; Yang *et al.*, 2011] or meta-analyses [Aulchenko *et al.*, 2007; Mägi and Morris, 2010; Purcell *et al.*, 2007; Willer *et al.*, 2010], others only implement an algorithm or method tailored to a certain task [Azencott *et al.*, 2013; Bulik-Sullivan *et al.*, 2014; Kang *et al.*, 2010; Lippert *et al.*, 2011; Llinares-López *et al.*, 2015a; Loh *et al.*, 2015; Rakitsch *et al.*, 2013b; Sugiyama *et al.*, 2014]. Various different data input and output formats, as well as the large fragmentation of these tools make them unnecessarily difficult to use. In addition, it is a cumbersome process to analyse and annotate GWAS results, since most of these tools often miss basic functionality to visualise or annotate their own output.

The objective of this chapter is to develop an integrated framework with a collection of popular methods and algorithms for performing GWASs and meta-analyses, as well as serving the community at large with easy to use data handling methods, visualisations and annotations of results. The framework comes with a common `C/C++` Application Programming Interface (API) and a `Python` interface which facilitates the integration, comparison and development of novel algorithms and pipelines. Furthermore, we will utilise this API to develop an easy to use command line interface in `Python`. The command line tool will offer an intuitive interface for performing GWASs with different algorithms, as well as the visualisation and annotation of these results. In essence, the API will be used throughout this thesis for the development of novel algorithms, as well as for being a resource for future developments beyond this thesis.

In the first half of this chapter we will review different regression models for GWASs and their statistical inference. We will then give a brief introduction about hypothesis testing for regression based models and about multiple hypothesis correction methods. In addition, we will summarise popular meta-analysis methods for GWASs. In the second half of this chapter we will describe the `easyGWASCore` framework, its architecture, APIs and command line interface. We will demonstrate the capabilities of the

API and the command line interface on various different examples. Eventually, we will analyse the performance of the `easyGWASCore` framework by comparing it to well established state-of-the-art tools.

2.1 Regression Based Methods for GWASs

Regression based methods are an important class of models that are commonly used in the field of GWASs to find associations between a single phenotype \mathbf{y} and a single genetic marker \mathbf{g} . A genetic marker can be a simple point mutation, such as a Single Nucleotide Polymorphisms (SNP), but also a more complex structural variation, such as a Copy Number Variation (CNV). In this section we will describe various regression models for GWASs and its corresponding statistical inference procedure to estimate unknown parameters in those models. For a more in depth explanation of general regression-based concepts we suggest the following literature [*Fahrmeir et al.*, 2013; *Hastie et al.*, 2009a].

2.1.1 Linear Regression

The basic idea in a linear regression model is to find a linear mapping between a quantitative or continuous phenotype \mathbf{y}_c and a genetic marker \mathbf{g} :

$$\begin{aligned}\mathbf{y}_c &= \beta_0 + \beta_1 \mathbf{g} + \epsilon, \\ \epsilon &\sim \mathcal{N}(0, \sigma^2 I),\end{aligned}\tag{2.1}$$

where β_0 is the weight of the intercept, β_1 the parametric weight of the genetic marker \mathbf{g} and ϵ the additive random error or noise term which we assume to be normally distributed. The phenotype \mathbf{y}_c is a n -dimensional vector of phenotypic observations $\mathbf{y}_c = (y_{c1}, y_{c2}, \dots, y_{cn})^\top \in \mathbb{R}^n$, where n is the number of samples. The genetic marker $\mathbf{g} = (a_1, a_2, \dots, a_n)^\top$ contains the encoded allele information a_i for sample i . An overview about the different allele encodings can be found in the Appendix C.2. Within the classical linear regression framework we assume that the noise ϵ is additive and follows approximately a Gaussian distribution with zero mean $\mathbb{E}[\epsilon] = 0$. Further, we assume that the noise is constant (homoscedastic) and uncorrelated leading to the covariance matrix $\text{Cov}[\epsilon] = \sigma^2 I$. Equation 2.1 can be rewritten in matrix notation:

$$\mathbf{y}_c \sim \mathcal{N}(\mathbf{G}\boldsymbol{\beta}, \sigma^2 I),\tag{2.2}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1)^\top \in \mathbb{R}^2$ and $\mathbf{G} \in \mathbb{R}^{n \times 2}$ is a matrix containing the intercept (a vector of ones $\mathbf{1}$) and a single genetic marker \mathbf{g} , that is $\mathbf{G} = (\mathbf{1}, \mathbf{g})$.

The parameters $\boldsymbol{\beta}$ and σ^2 are unknown and have to be estimated. For this purpose, several different techniques can be used. In the following we will describe two commonly used methods. First we will give a brief overview about the method of least squares and then we introduce the maximum likelihood estimator.

Method of Least Squares

The method of least squares is one of the most commonly used techniques for finding a linear fit of the parametric weight β to minimise the sum of the squared training error [Fahrmeir et al., 2013; Hastie et al., 2009a]:

$$\begin{aligned} \text{LS}(\beta) &= \sum_{i=1}^n (y_{ci} - \beta_0 - \beta_1 g_i)^2 \\ &= (\mathbf{y}_c - \mathbf{G}\beta)^\top (\mathbf{y}_c - \mathbf{G}\beta). \end{aligned} \quad (2.3)$$

To find the global minimum we have to compute the gradient of Equation 2.3:

$$\nabla_{\beta} \text{LS}(\beta) = -2(\mathbf{G}^\top \mathbf{y}_c - \mathbf{G}^\top \mathbf{G}\beta). \quad (2.4)$$

Under the assumption that the matrix $\mathbf{G}^\top \mathbf{G}$ is positive definite we can derive a closed form equation for the least squares estimator $\hat{\beta}_{LS}$ by setting Equation 2.4 to 0:

$$\begin{aligned} -2(\mathbf{G}^\top \mathbf{y}_c - \mathbf{G}^\top \mathbf{G}\beta) &= 0 \\ \Leftrightarrow -\mathbf{G}^\top \mathbf{y}_c + \mathbf{G}^\top \mathbf{G}\beta &= 0 \\ \Leftrightarrow \mathbf{G}^\top \mathbf{G}\beta &= \mathbf{G}^\top \mathbf{y}_c \\ \Leftrightarrow \hat{\beta}_{LS} &= (\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{y}_c. \end{aligned} \quad (2.5)$$

Figure 2.1 illustrates the method of least squares. Note that the least squares estimator $\hat{\beta}_{LS}$ in Equation 2.5 neglects any assumption about the distribution of the noise ϵ . To estimate the parameters considering the distribution of the noise term we can use a maximum likelihood estimator.

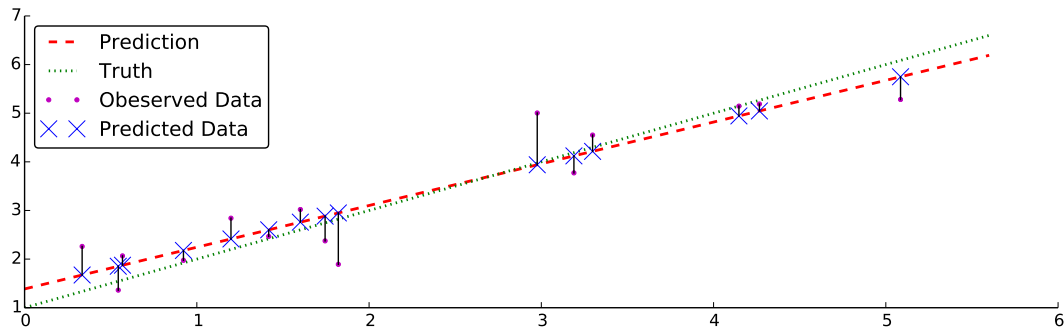


Figure 2.1: Illustration of the least squared estimator. Here, we try to minimise the sum of the squared training error, that is the sum of the distances between the observed data points (magenta points) and the predicted data points (blue crosses).

Maximum Likelihood Estimator

Estimation of the weight vector β : As written in Equation 2.2 we assume additive Gaussian distributed noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ with zero mean and constant variance σ^2 . To estimate the unknown parameters β and σ^2 from Equation 2.2 we can derive the

following likelihood function [Fahrmeir et al., 2013]:

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma^2) &= \mathcal{N}(\mathbf{y}_c | \mathbf{G}\boldsymbol{\beta}, \sigma^2 I) \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp^{-\frac{1}{2\sigma^2}(\mathbf{y}_c - \mathbf{G}\boldsymbol{\beta})^\top(\mathbf{y}_c - \mathbf{G}\boldsymbol{\beta})}. \end{aligned} \quad (2.6)$$

By applying the logarithmic function we can simplify the likelihood function from Equation 2.6:

$$\begin{aligned} nll(\boldsymbol{\beta}, \sigma^2) &= -\frac{1}{2}n \log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y}_c - \mathbf{G}\boldsymbol{\beta})^\top(\mathbf{y}_c - \mathbf{G}\boldsymbol{\beta}) \\ &= -\frac{1}{2}n \log(2\pi) - \frac{1}{2}n \log(\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y}_c^\top \mathbf{y}_c - 2\mathbf{G}^\top \boldsymbol{\beta}^\top \mathbf{y}_c + \boldsymbol{\beta}^\top \mathbf{G}^\top \mathbf{G} \boldsymbol{\beta}). \end{aligned} \quad (2.7)$$

To infer the maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{ML}$ we set the gradient to zero and solve for the parameter $\boldsymbol{\beta}$:

$$\begin{aligned} \frac{\partial nll(\boldsymbol{\beta}, \sigma^2)}{\partial \boldsymbol{\beta}} &= -\frac{1}{2\sigma^2}(-2\mathbf{G}^\top \mathbf{y}_c + 2\mathbf{G}^\top \mathbf{G} \boldsymbol{\beta}) = \frac{1}{\sigma^2}(\mathbf{G}^\top \mathbf{y}_c - \mathbf{G}^\top \mathbf{G} \boldsymbol{\beta}) \\ \text{Set to 0: } &\frac{1}{\sigma^2}(\mathbf{G}^\top \mathbf{y}_c - \mathbf{G}^\top \mathbf{G} \boldsymbol{\beta}) = 0 \\ &\Leftrightarrow \mathbf{G}^\top \mathbf{G} \boldsymbol{\beta} = \mathbf{G}^\top \mathbf{y}_c \\ &\Leftrightarrow \hat{\boldsymbol{\beta}}_{ML} = (\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{y}_c. \end{aligned} \quad (2.8)$$

Note that the found solution is the global minimum of the negative log-likelihood function (Equation 2.7), since the function is convex. As we can see in Equation 2.8 the maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{ML}$ is equal to the least square estimator $\hat{\boldsymbol{\beta}}_{LS}$ in Equation 2.5. Hence, minimising the sum of the squared training error is equal to maximising the likelihood or minimising the negative log-likelihood, respectively.

Based on these estimates we are able to create a predictor to approximate the target variable \mathbf{y}_c :

$$\hat{\mathbf{y}}_c = \mathbf{G}\hat{\boldsymbol{\beta}}_{LS} = \mathbf{G}\hat{\boldsymbol{\beta}}_{ML} = \mathbf{G} \underbrace{(\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{y}_c}_{\hat{\boldsymbol{\beta}}_{LS} = \hat{\boldsymbol{\beta}}_{ML}}. \quad (2.9)$$

The residuals, that is the difference between the predicted values in $\hat{\mathbf{y}}_c$ and the true phenotypic observations in \mathbf{y}_c , can be computed as follows:

$$\hat{\mathbf{r}} = \mathbf{y}_c - \hat{\mathbf{y}}_c. \quad (2.10)$$

Estimation of the noise variance σ^2 : To get an estimation of the variance parameter we have to differentiate the negative log-likelihood function from Equation 2.7 with respect to σ^2 . Replacing $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}_{LS}$ or $\hat{\boldsymbol{\beta}}_{ML}$ and setting the gradient to zero we

get [Fahrmeir et al., 2013]:

$$\begin{aligned}
\frac{\partial nll(\boldsymbol{\beta}, \sigma^2)}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{LS})^\top (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{LS}) \\
&= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML})^\top (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML}), \\
\text{Set to 0: } &-\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML})^\top (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML}) = 0 \\
&\Leftrightarrow \frac{1}{2\sigma^4} (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML})^\top (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML}) = \frac{n}{2\sigma^2} \\
&\Leftrightarrow \frac{1}{\sigma^2} (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML})^\top (\mathbf{y}_c - \mathbf{G}\hat{\boldsymbol{\beta}}_{ML}) = n \\
&\Leftrightarrow \frac{1}{\sigma^2} (\mathbf{y}_c - \mathbf{G} \underbrace{(\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{y}_c}_{\hat{\boldsymbol{\beta}}_{ML}})^\top (\mathbf{y}_c - \mathbf{G} \underbrace{(\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{y}_c}_{\hat{\boldsymbol{\beta}}_{ML}}) = n \\
&\Leftrightarrow \hat{\sigma}_{ML}^2 = \frac{1}{n} (\mathbf{y}_c - \underbrace{\mathbf{G}(\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{y}_c}_{\hat{\mathbf{y}}_c})^\top (\mathbf{y}_c - \underbrace{\mathbf{G}(\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{y}_c}_{\hat{\mathbf{y}}_c}) \\
&\Leftrightarrow \hat{\sigma}_{ML}^2 = \frac{1}{n} \underbrace{(\mathbf{y}_c - \hat{\mathbf{y}}_c)^\top}_{\hat{\mathbf{r}}} \underbrace{(\mathbf{y}_c - \hat{\mathbf{y}}_c)}_{\hat{\mathbf{r}}} \\
&\Leftrightarrow \hat{\sigma}_{ML}^2 = \frac{1}{n} \hat{\mathbf{r}}^\top \hat{\mathbf{r}}. \tag{2.11}
\end{aligned}$$

We now retrieved an estimator to approximate the noise variance $\hat{\sigma}_{ML}^2$. However, this estimator is biased as shown in Fahrmeir et al. [2013]. The unbiased restricted maximum likelihood estimator (REML) is defined as:

$$\hat{\sigma}_{REML}^2 = \frac{1}{n-p} \hat{\mathbf{r}}^\top \hat{\mathbf{r}}, \tag{2.12}$$

where p is the rank of matrix \mathbf{G} . A full proof can be found in Fahrmeir et al. [2013].

2.1.2 Logistic Regression

Linear regression is appropriate if the phenotype is continuous and approximately Gaussian distributed. However, in some cases the phenotype is binary and approximately drawn from $\mathbf{y}_{bi} \sim \mathcal{B}(p_i)$, where \mathcal{B} is the Bernoulli distribution and p_i is the conditional probability of $y_{bi} = 1$ given the allele a_i [Fahrmeir et al., 2013; Hastie et al., 2009a]:

$$p_i = P(y_{bi} = 1 | a_i). \tag{2.13}$$

One example of a GWAS with a binary phenotype is a case-control study for which we investigate a group of patients having a specific disease (cases) and a control group without the disease (controls). Here we try to identify genetic markers that are significantly associated with patients having the disease. For such experiments, the classical linear regression model is not well suited. The first reason is, that a linear regression allows for $\hat{\mathbf{y}}_{\mathbf{b}}$ values that are larger than one and smaller than zero. This is not valid since we are modelling a probability $P(y_{bi} = 1 | a_i)$. The second reason is that we assume constant (homoscedastic) noise for the variance parameter σ^2 . However,

if a phenotypic observation y_{bi} is drawn from a Bernoulli distribution the variance is defined as:

$$\text{Var}(y_{bi}) = p_i(1 - p_i). \quad (2.14)$$

Thus, the variance σ^2 depends on the different alleles a_i and therefore cannot be assumed to be constant (homoscedastic).

For that reason we model the conditional probability that a set of individuals belongs to the label 1 (cases) given a genetic marker $\mathbf{G} = (\mathbf{1}, \mathbf{g})$ and the parametric weights $\boldsymbol{\beta} = (\beta_0, \beta_1)^\top$ as follows:

$$P(\mathbf{y}_{\mathbf{b}} = \mathbf{1} | \mathbf{G}; \boldsymbol{\beta}) = h(\mathbf{G}\boldsymbol{\beta}), \quad (2.15)$$

where h is the logistic response function, that is:

$$h(\mathbf{G}\boldsymbol{\beta}) = \frac{e^{\mathbf{G}\boldsymbol{\beta}}}{1 + e^{\mathbf{G}\boldsymbol{\beta}}} = \frac{1}{1 + e^{-\mathbf{G}\boldsymbol{\beta}}}. \quad (2.16)$$

Maximum Likelihood Estimator

Because the phenotype $\mathbf{y}_{\mathbf{b}}$ is binary and thus drawn from a Bernoulli distribution $\mathbf{y}_{\mathbf{b}} \sim \mathcal{B}(p)$ we can derive the following maximum likelihood function [Fahrmeir et al., 2013; Hastie et al., 2009a]:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n p_i^{y_{bi}} (1 - p_i)^{1 - y_{bi}}. \quad (2.17)$$

Again we can simplify Equation 2.18 by taking the logarithmic function:

$$\begin{aligned} ll(\boldsymbol{\beta}) &= \sum_{i=1}^n \log L_i(\boldsymbol{\beta}) \\ &= \sum_{i=1}^n [y_{bi} \log(p_i) - y_{bi} \log(1 - p_i) + \log(1 - p_i)] \\ &= \sum_{i=1}^n \left[y_{bi} \log \left(\frac{p_i}{1 - p_i} \right) + \log(1 - p_i) \right] \\ &= \sum_{i=1}^n \left[y_{bi} \log \left(\frac{h(\mathbf{G}_i\boldsymbol{\beta})}{1 - h(\mathbf{G}_i\boldsymbol{\beta})} \right) + \log(1 - h(\mathbf{G}_i\boldsymbol{\beta})) \right] \\ &= \sum_{i=1}^n \left[y_{bi} \log \left(\frac{\frac{1}{1 + \exp(-\mathbf{G}_i\boldsymbol{\beta})}}{1 - \frac{1}{1 + \exp(-\mathbf{G}_i\boldsymbol{\beta})}} \right) + \log \left(1 - \frac{1}{1 + \exp(-\mathbf{G}_i\boldsymbol{\beta})} \right) \right] \\ &= \sum_{i=1}^n \left[y_{bi} \mathbf{G}_i\boldsymbol{\beta} - \log(1 + e^{\mathbf{G}_i\boldsymbol{\beta}}) \right], \end{aligned} \quad (2.18)$$

where $\mathbf{G}_i = (1, a_i)$, containing the constant intercept 1 and the encoded allelic information a_i for sample i .

To estimate the unknown parameter $\boldsymbol{\beta}$ in this logistic regression model we have to differentiate the log-likelihood function from Equation 2.18 with respect to $\boldsymbol{\beta}$ and set it to zero:

$$\begin{aligned} s(\boldsymbol{\beta}) &= \frac{\partial l(\boldsymbol{\beta})}{\partial(\boldsymbol{\beta})} = \sum_{i=1}^n \frac{\partial l_i(\boldsymbol{\beta})}{\partial(\boldsymbol{\beta})} = \sum_{i=1}^n \mathbf{G}_i \left[y_{bi} - \underbrace{\frac{1}{1 + e^{-\mathbf{G}_i \boldsymbol{\beta}}}}_{p_i} \right] \\ &= \sum_{i=1}^n \mathbf{G}_i (y_{bi} - p_i) = \mathbf{G}^\top (\mathbf{y}_b - \mathbf{p}) = 0. \end{aligned} \quad (2.19)$$

This function is often referred to as the *score*-function s [Fahrmeir et al., 2013; Hastie et al., 2009a]. To solve the non-linear score function in Equation 2.19 we can use the iterative Newton-Raphson method [Ypma, 1995]. For this purpose, we have to determine the Hessian matrix by computing the second derivative of Equation 2.18 [Fahrmeir et al., 2013; Hastie et al., 2009a]:

$$H(\boldsymbol{\beta}) = \frac{\partial^2 l(\boldsymbol{\beta})}{\partial(\boldsymbol{\beta})\partial(\boldsymbol{\beta}^\top)} = - \sum_{i=1}^n \left[\mathbf{G}_i \mathbf{G}_i^\top p_i (1 - p_i) \right] = -\mathbf{G}^\top \mathbf{W} \mathbf{G}, \quad (2.20)$$

where \mathbf{W} is a $n \times n$ diagonal matrix and the i th diagonal element of matrix \mathbf{W} is $p_i(1 - p_i)$. The Newton-Raphson method is an iterative method for solving a non-linear equation numerically [Ypma, 1995]. Because each step depends on the previous step an initial starting condition has to be chosen. For this purpose, we initialise the parameter $\boldsymbol{\beta}$ with 0. The k th update step for the Newton-Raphson method is defined as:

$$\begin{aligned} \boldsymbol{\beta}^{(k+1)} &= \boldsymbol{\beta}^{(k)} - H(\boldsymbol{\beta}^{(k)})^{-1} s(\boldsymbol{\beta}^{(k)}) \\ &= \boldsymbol{\beta}^{(k)} + (\mathbf{G}^\top \mathbf{W} \mathbf{G})^{-1} \mathbf{G}^\top (\mathbf{y}_b - \mathbf{p}). \end{aligned} \quad (2.21)$$

The update procedure can be truncated if $\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\|_2 < \rho$, where $\|\cdot\|_2$ is the l_2 -norm and ρ is a tolerance threshold, e.g. $1e^{-16}$.

2.1.3 Linear Mixed Models

Linear mixed models (LMMs) are commonly used to account for fixed and random effects at the same time. In the field of GWASs, linear mixed models are often used to account for hidden confounding, such as population stratification [Kang et al., 2008, 2010; Lippert et al., 2011; Listgarten et al., 2012, 2013; Yu et al., 2006; Zhang et al., 2010c]. Since population structure cannot be directly observed, we can treat it as a random effect. Given a population of n samples we can write the linear mixed model

as follows:

$$\mathbf{y} = \underbrace{\mathbf{G}\boldsymbol{\beta}}_{\text{fixed}} + \underbrace{\mathbf{u}}_{\text{random}} + \underbrace{\epsilon}_{\text{noise}}, \quad (2.22)$$

where \mathbf{y} is the phenotype of size n and \mathbf{G} is the fixed effects matrix including the intercept and the genetic marker, that is $\mathbf{G} = (\mathbf{1}, \mathbf{g})$. The parameter $\boldsymbol{\beta} = (\beta_0, \beta_1)^\top$ contains the weights for the intercept β_0 and the genetic marker β_1 . Similar to a classical linear regression, the noise term ϵ is assumed to be additive, homoscedastic and uncorrelated and follows approximately a Gaussian distribution with zero mean and constant, uncorrelated variance $\epsilon \sim \mathcal{N}(0, \sigma_e^2 \mathbf{I})$. The term \mathbf{u} represents the random effect vector $\mathbf{u} \sim \mathcal{N}(0, \sigma_g^2 \mathbf{K})$, where σ_g^2 is the genetic variance. The covariance matrix is a $n \times n$ kinship matrix \mathbf{K} , measuring the genetic similarity between all n samples. Thus, the kinship matrix can be used to account for population structure, family structure and cryptic relatedness within a population of n samples [Kang *et al.*, 2008, 2010; Lippert *et al.*, 2011; Listgarten *et al.*, 2012, 2013; Zhang *et al.*, 2010c]. There are various different kinship matrices, such as the IBS or Balding-Nichols [Balding and Nichols, 1995] matrix. A commonly used kinship matrix is the so called realized relationship kernel (RRK) [Hayes *et al.*, 2009]:

$$\mathbf{K} = \frac{1}{n} \mathbf{M} \mathbf{M}^\top, \quad (2.23)$$

where $\mathbf{M} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m)$ is a matrix of size $n \times m$ with m being the total number of available genetic markers. We assume that \mathbf{M} is normalized with zero mean and unit variance.

Maximum Likelihood Estimator

Similar to the classical linear and logistic regression models we have to estimate the unknown parameters, in this case $\boldsymbol{\beta}$, the genetic variance σ_g^2 , as well as the noise variance σ_e^2 . Again, this can be done by using maximum likelihood inference techniques. The likelihood function is defined as follows:

$$L(\boldsymbol{\beta}, \sigma_g^2, \sigma_e^2) = \mathcal{N}(\mathbf{y} | \mathbf{G}\boldsymbol{\beta}; \sigma_g^2 \mathbf{K} + \sigma_e^2 \mathbf{I}). \quad (2.24)$$

We can simplify the maximum likelihood formulation from Equation 2.24 by taking the logarithmic function and introducing the term $\delta = \sigma_e^2 / \sigma_g^2$, that is the ratio of the noise variance σ_e^2 and the genetic variance σ_g^2 [Kang *et al.*, 2008; Welham and Thompson, 1997]:

$$\begin{aligned} ll(\boldsymbol{\beta}, \sigma_g^2, \sigma_e^2) &= \log \mathcal{N}(\mathbf{y} | \mathbf{G}\boldsymbol{\beta}; \sigma_g^2 \mathbf{K} + \sigma_e^2 \mathbf{I}) \\ \Leftrightarrow ll(\boldsymbol{\beta}, \sigma_g^2, \delta) &= \log \mathcal{N}(\mathbf{y} | \mathbf{G}\boldsymbol{\beta}; \sigma_g^2 (\mathbf{K} + \delta \mathbf{I})). \end{aligned} \quad (2.25)$$

With these simplifications we can estimate the weight parameter $\hat{\boldsymbol{\beta}}_{ML}$ and the genetic variance parameter $\hat{\sigma}_{g(ML)}^2$ in closed form. Thus, we only have to solve the optimization problem for the ratio parameter δ . The optimization problem can be solved

numerically using a root finding method such as Newton-Raphson [Ypma, 1995] or Brent's method [Brent, 1971].

Estimation of the weight vector β : To retrieve an estimation for the parameter $\hat{\beta}_{ML}$ we differentiate the log-likelihood function from Equation 2.25 by setting the gradient to zero and solving for the parameter β :

$$\begin{aligned} \frac{\partial l(\beta, \sigma_g^2, \delta)}{\partial \beta} &= -\frac{1}{2\sigma_g^2} (\mathbf{y} - \mathbf{G}\beta)^\top (\mathbf{K} + \delta\mathbf{I})^{-1} (\mathbf{y} - \mathbf{G}\beta) \\ &= \frac{1}{\sigma_g^2} \left(\mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{G}\beta - \mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{y} \right), \\ \text{Set to Zero: } \frac{1}{\sigma_g^2} \left(\mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{G}\beta - \mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{y} \right) &= 0 \\ \Leftrightarrow \mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{G}\beta &= \mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{y} \\ \Leftrightarrow \hat{\beta}_{ML} &= \left(\mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{G} \right)^{-1} \mathbf{G}^\top (\mathbf{K} + \delta\mathbf{I})^{-1} \mathbf{y}. \end{aligned} \quad (2.26)$$

Estimation of the genetic variance σ_g^2 : Similar to the latter estimations we can derive a maximum likelihood estimator for the genetic variance by differentiating the gradient of Equation 2.25 with respect to σ_g^2 . Replacing β with the maximum likelihood estimator $\hat{\beta}_{ML}$ (Equation 2.26) and setting the gradient to zero we get:

$$\begin{aligned} \frac{\partial l(\beta, \sigma_g^2, \delta)}{\partial \sigma_g^2} &= -\frac{n}{2\sigma_g^2} + \frac{1}{2\sigma_g^4} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML})^\top (\mathbf{K} + \delta\mathbf{I})^{-1} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML}), \\ \text{Set to Zero: } -\frac{n}{2\sigma_g^2} + \frac{1}{2\sigma_g^4} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML})^\top (\mathbf{K} + \delta\mathbf{I})^{-1} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML}) &= 0 \\ \Leftrightarrow \frac{1}{\sigma_g^2} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML})^\top (\mathbf{K} + \delta\mathbf{I})^{-1} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML}) &= n \\ \Leftrightarrow \hat{\sigma}_{g(ML)}^2 &= \frac{1}{n} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML})^\top (\mathbf{K} + \delta\mathbf{I})^{-1} (\mathbf{y} - \mathbf{G}\hat{\beta}_{ML}). \end{aligned} \quad (2.27)$$

Estimation of the ratio parameter δ : After we estimated the unknown parameters $\hat{\beta}_{ML}$ and $\hat{\sigma}_{g(ML)}^2$ we can write the log-likelihood function from Equation 2.25 as a function only of δ [Kang *et al.*, 2008]:

$$l(\hat{\beta}_{ML}, \hat{\sigma}_{g(ML)}^2, \delta) = l(\delta). \quad (2.28)$$

Using Newton-Raphson or Brent's method we can solve this optimization problem numerically with respect to the parameter δ .

Since we are testing several thousands of SNPs we have to compute $(\mathbf{K} + \delta\mathbf{I})^{-1}$ for every single SNP. This operation is cubic in n , that is $\mathcal{O}(n^3)$. This leads to a computational bottleneck if we investigate hundreds of thousands of SNPs. Kang *et al.* [2008, 2010] and Lippert *et al.* [2011] proposed several techniques to efficiently speed up linear mixed models. In the following we briefly introduce factored spectrally transformed linear mixed models (FaSTLMM) by Lippert *et al.* [2011].

Factored Spectrally Transformed Linear Mixed Models (FaSTLMM): As proposed by *Lippert et al.* [2011] we can replace the kinship matrix \mathbf{K} with its spectral decomposition (often referred to as eigendecomposition), that is:

$$\mathbf{K} = \mathbf{U}\mathbf{S}\mathbf{U}^\top, \quad (2.29)$$

where \mathbf{U} is a $n \times n$ matrix whose i th column is the i th eigenvector of the kinship matrix \mathbf{K} . The matrix \mathbf{S} is a diagonal matrix whose i th diagonal element is the i th eigenvalue of the i th eigenvector. Using this trick we now can modify Equation 2.25 [*Lippert et al.*, 2011]:

$$\begin{aligned} l(\boldsymbol{\beta}, \sigma_g^2, \delta) &= \log \mathcal{N}(\mathbf{y} | \mathbf{G}\boldsymbol{\beta}; \sigma_g^2 (\mathbf{K} + \delta \mathbf{I})) \\ &= \log \mathcal{N}(\mathbf{y} | \mathbf{G}\boldsymbol{\beta}; \sigma_g^2 (\mathbf{U}\mathbf{S}\mathbf{U}^\top + \delta \mathbf{I})) \\ &= \log \mathcal{N}((\mathbf{U}^\top \mathbf{y}) | (\mathbf{U}^\top \mathbf{G})\boldsymbol{\beta}; \sigma_g^2 (\mathbf{S} + \delta \mathbf{I})). \end{aligned} \quad (2.30)$$

Since the kinship matrix \mathbf{K} is calculated using all SNPs or a subset of SNPs [*Listgarten et al.*, 2012, 2013], the factorial decomposition can be computed once for all SNPs we like to test, which is a cubic operation $\mathcal{O}(n^3)$. After the decomposition is computed we are able to efficiently estimate the unknown parameters $\hat{\boldsymbol{\beta}}_{ML}$, $\hat{\sigma}_{g(ML)}^2$ and $\hat{\delta}_{ML}$ for every single SNP (a detailed derivation using the factorial decomposition can be found in the Supplementary Material of *Lippert et al.* [2011]).

2.2 Hypothesis and Multiple Testing

2.2.1 Hypothesis Testing for Regression Methods

In the last sections we introduced various regression models and showed how we can estimate the unknown parameters. However, to answer the question of whether a given genetic marker \mathbf{g} is significantly associated with a specific phenotype \mathbf{y} , we have to perform a statistical hypothesis test. For this purpose, we have to compare the null hypothesis \mathcal{H}_0 that the genetic marker \mathbf{g} has no effect on the phenotype \mathbf{y} with the alternative hypothesis \mathcal{H}_1 that this marker is associated with this phenotype. In other words, if we assume that the null hypothesis \mathcal{H}_0 is true, the estimated weight $\hat{\beta}_1$ of the genetic marker \mathbf{g} would be zero. Thus, $\hat{\beta}_1$ has to be different from zero such that the alternative hypothesis \mathcal{H}_1 could be true. The weights in a regression model are often denoted as the *effect estimates*, as well. We can summarise the hypothesis we like to test as follows:

$$\mathcal{H}_0 : \beta_1 = 0 \text{ against } \mathcal{H}_1 : \beta_1 \neq 0. \quad (2.31)$$

For hypothesis testing several techniques can be used. A widely used test is the likelihood ratio test (LRT). The likelihood ratio is computed between the likelihood function of the alternative model and the null model. By applying the logarithmic function we

can write:

$$\begin{aligned} LR &= 2 \log \left(\frac{\max_{\mathcal{H}_1} L_1}{\max_{\mathcal{H}_0} L_0} \right) = 2(l_1 - l_0), \\ LR &\sim \chi_k^2, \end{aligned} \quad (2.32)$$

where l_0 is the log-likelihood function of the null model and l_1 is the log-likelihood of the alternative model. The number of degrees of freedom k is dependent on the difference of parameters between the null and alternative model [Wilks, 1938]. In GWASs we usually test if a single genetic marker has an effect on the phenotype. Thus, the null distribution of this likelihood-ratio statistic is approximately χ_1^2 distributed with one degree of freedom [Fahrmeir *et al.*, 2013].

In the following example we perform a LRT, using a linear regression, to test whether a single SNP \mathbf{g} is significantly associated with a given phenotype \mathbf{y} . For this purpose, we estimate the unknown parameters for the null model ($\beta_1 = 0$) and the alternative model ($\beta_1 \neq 0$) and calculate the LR, as follows:

$$\begin{aligned} LR &= 2 \log \left(\frac{\max_{\mathcal{H}_1} L_1}{\max_{\mathcal{H}_0} L_0} \right) = 2 \log \left(\frac{\mathcal{N}(\mathbf{y} | \hat{\beta}_0(\mathcal{H}_1) + \hat{\beta}_1(\mathcal{H}_1) \mathbf{g}; \hat{\sigma}_{\mathcal{H}_1}^2 \mathbf{I})}{\mathcal{N}(\mathbf{y} | \hat{\beta}_0(\mathcal{H}_0); \hat{\sigma}_{\mathcal{H}_0}^2 \mathbf{I})} \right) \\ &= 2 \left(\log \left(\mathcal{N}(\mathbf{y} | \hat{\beta}_0(\mathcal{H}_1) + \hat{\beta}_1(\mathcal{H}_1) \mathbf{g}; \hat{\sigma}_{\mathcal{H}_1}^2 \mathbf{I}) \right) - \log \left(\mathcal{N}(\mathbf{y} | \hat{\beta}_0(\mathcal{H}_0); \hat{\sigma}_{\mathcal{H}_0}^2 \mathbf{I}) \right) \right). \end{aligned} \quad (2.33)$$

We then can use the complementary cumulative distribution function (also referred to as survival function) of the χ^2 distribution to compute the p-value for a given LR . The LRT can be easily adapted to all regression models, discussed in this thesis, by simply modifying the likelihood terms in Equation 2.33.

We can reject the null hypothesis and call a statistical test significant, if the p-value is below a predefined significance threshold α . A commonly used significance cut-off is 5% ($\alpha = 0.05$), this means that the null hypothesis is rejected in at most 5% of the cases when it is actually true. Thus, a falsely rejected null hypothesis is also called a *false positive* (FP) or a *type-1 error*.

2.2.2 Multiple Hypothesis Testing

When testing thousands to millions of hypothesis simultaneously, which is a common setting in GWASs [Bush and Moore, 2012; Johnson *et al.*, 2010], we are confronted with the so called multiple hypothesis testing problem. For example, let us consider a typical GWAS in which we are testing more than 500,000 SNPs. If we assume a significance threshold of 5% ($\alpha = 0.05$), we would expect that 25,000 SNPs are deemed significant just due to random chance. Within this family of m tests we can compute the probability of making at least one type-1 error (or a false positive (FP)):

$$P(V \geq 1) = 1 - (1 - \alpha)^m, \quad (2.34)$$

where V is the number of type-1 errors. This probability is called the *family-wise error rate* (FWER). As we can see in Figure 2.2, the probability of making at least one type-1 error converges quickly to 1, with an increasing number of tests.

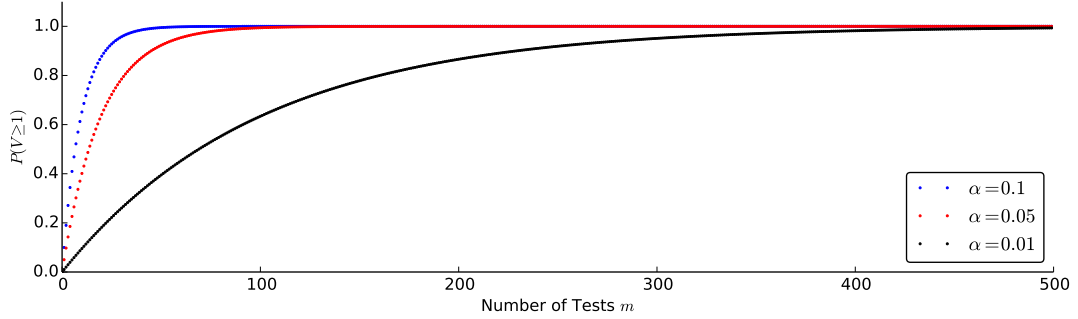


Figure 2.2: Probability of making at least one type-1 error. Here, we show the probability of making at least one type-1 error when testing m multiple hypothesis with respect to three different significance thresholds α .

The Bonferroni method [Abdi, 2007] is a widely used approach to control the FWER. It simply rejects the null hypothesis $\mathcal{H}_{0(i)}$ for test i if the p-value pv_i is less or equal to α/m . However, the Bonferroni correction tends to be too conservative. We can modify Equation 2.34 with the Bonferroni corrected significance cut-off:

$$P_B(V \geq 1) = 1 - \left(1 - \left(\frac{\alpha}{m}\right)\right)^m. \quad (2.35)$$

The limit of Equation 2.35, as the number of tests approaches infinity, is:

$$\lim_{m \rightarrow \infty} \left(1 - \left(1 - \left(\frac{\alpha}{m}\right)\right)^m\right) = 1 - e^{-\alpha}. \quad (2.36)$$

Equation 2.36 is always smaller than α , that is:

$$1 - e^{-\alpha} < \alpha \quad \forall \alpha \in \mathbb{R}^+. \quad (2.37)$$

If we assume a significance threshold of 5%, the limit of the probability of making at least one type-1 error is:

$$\lim_{m \rightarrow \infty} \left(1 - \left(1 - \left(\frac{0.05}{m}\right)\right)^m\right) = 0.0487706. \quad (2.38)$$

Equation 2.36 and 2.38 shows that the Bonferroni methods tends to be always smaller than α and hence too conservative, which might lead to a higher probability of *false negatives* (FN). A FN is often called a *type-2 error*, that is the acceptance of the null-hypothesis when in fact the alternative hypothesis is the true. A slightly more powerful approach to control the FWER was introduced by Holm [1979]. The Holm-Bonferroni method is a sequential step-down adjustment of each p-value. For this purpose, we first have to sort all p-values in decreasing order:

$$pv_1 \leq pv_2 \leq pv_3 \leq \dots \leq pv_m. \quad (2.39)$$

The adjusted Holm-Bonferroni p-value $\hat{p}v_i$, for a given significance threshold α , is computed, as follows:

$$\hat{p}v_i = \min \{(m - i + 1)pv_i, 1\}. \quad (2.40)$$

These two methods are a small selection of approaches to control the FWER, other approaches are described by *Šidák* [1967], *Hochberg* [1988], *Hommel* [1989], *Hochberg and Benjamini* [1990], *Tarone* [1990] and *Hommel and Krummenauer* [1998].

As shown before, controlling the FWER reduces the number of type-1 errors but at the same time increases the probability of making type-2 errors. An alternative strategy, introduced by *Benjamini and Hochberg* [1995], controls the *false discovery rate* (FDR). The FDR is the expected proportion of type-1 errors V among the set of all rejected hypothesis R [*Benjamini and Hochberg*, 1995]:

$$\begin{aligned} \text{FDR} &= E \left[\frac{V}{R} \right] \text{ where } \frac{V}{R} = 0 \text{ if } V = R = 0 \\ &= E \left[\frac{V}{R} | R > 0 \right] P(R > 0). \end{aligned} \quad (2.41)$$

In the case when all hypothesis are true the FWER = FDR, since $V = R$ that leads to FWER = $P(V \geq 1) = E \left[\frac{V}{R} \right] = \text{FDR}$. As shown in *Benjamini and Hochberg* [1995], controlling the FDR is less conservative than controlling the FWER. Consequently, controlling the FWER also includes controlling the FDR. On the other hand, controlling the FDR does not necessarily control the FWER. However, controlling the FDR can be more powerful than controlling the FWER [*Benjamini and Hochberg*, 1995], since it decreases the probability of making a type-2 error (false negative) (see Figure 2.3).



Figure 2.3: FWER vs. FDR vs. no correction. The further left a method the higher the probability of making more type-2 errors, whereas the further right the more type-1 errors.

The first step to control the FDR after *Benjamini and Hochberg* [1995], is to order the unadjusted p-values in decreasing order as in Equation 2.39. Next, we have to find the largest $k \in \mathbb{N}^+$, given a FDR significance cut-off α , such that:

$$pv_{(k)} \leq \alpha \frac{k}{m}, \quad (2.42)$$

where m is the total number of tests. We then can reject all k null hypothesis $\mathcal{H}_{0(i)}$, $i = 1, \dots, k$ and declare them as significant. To correct the raw p-value $pv_{(i)}$ by computing an adjusted *Benjamini & Hochberg* FDR p-value $\hat{p}v_{(i)}$, we have to follow a linear step-

up adjustment for the sorted list of p-values:

$$\hat{pV}_{(i)} = \begin{cases} pV_{(m)} & \text{if } i = m, \\ \min \left\{ \frac{m}{i} pV_{(i)}, \hat{pV}_{(i+1)} \right\} & \text{for } i = m - 1, \dots, 1. \end{cases} \quad (2.43)$$

Note that this method only guarantees to control the FDR if the p-values $pV_{(i)}$ are independent and uniformly distributed under their null hypothesis $\mathcal{H}_{0(i)}$. For genome-wide association studies this case is rarely true, since genetic markers are in linkage disequilibrium (LD), that is the correlation between genetic markers within a specific regions on the chromosome.

Benjamini and Yekutieli [2001] proposed an extension to control the FDR under the assumption of dependencies (dependent FDR). For this purpose, we have to extend Equation 2.42 with the term $c = \sum_{i=1}^m \frac{1}{m}$:

$$pV_{(k)} \leq \alpha \frac{k}{m} \frac{1}{c}. \quad (2.44)$$

Consequently, we have to adjust the step-up procedure in Equation 2.43 to compute the adjusted p-values after *Benjamini, Hochberg & Yekutieli*:

$$\hat{pV}_{(i)} = \begin{cases} c pV_{(m)} & \text{if } i = m, \\ \min \left\{ c \frac{m}{i} pV_{(i)}, \hat{pV}_{(i+1)} \right\} & \text{for } i = m - 1, \dots, 1. \end{cases} \quad (2.45)$$

However, this method is more conservative than the original *Benjamini & Hochberg* procedure. Importantly, quite often we do not know the exact form of the dependency structure between the random variables. Because of that, making any kind of assumptions between the dependencies of variables can have unforeseeable consequences [*Ewens and Grant*, 2005].

An alternative definition of the FDR, the *positive false discovery rate* (pFDR) [*Storey*, 2002; *Storey and Tibshirani*, 2003], is defined as:

$$\text{pFDR} = E \left[\frac{V}{R} | R > 0 \right]. \quad (2.46)$$

The pFDR is the rate that at least one test is positive (or the rate that discoveries are false), whereas the FDR is the rate that false discoveries occur. Storey argued that the extra term $P(R > 0)$ in Equation 2.41 might lead to ambiguous interpretations of results [*Storey*, 2003, 2011; *Zaykin et al.*, 2000].

Storey and Tibshirani [2003] introduced the *q-value* as a method to control the pFDR in the field of genome-wide studies and defined the q-value as follows:

“The q-value of a particular feature in a genome-wide data set is the expected proportion of false positives incurred when calling that feature significant.”

To estimate q-values based on a set of pre-computed p-values, we first have to sort all p-values in decreasing order, as in Equation (2.39). The next step involves the estimation of $\hat{\pi}_0$, that is the proportion of genetic markers that are truly null [Storey and Tibshirani, 2003]:

$$\hat{\pi}_0(\lambda) = \frac{\#\{pv_i > \lambda; i = 1, \dots, m\}}{m(1 - \lambda)}, \quad (2.47)$$

where $\#(\cdot)$ is the number of all p-values pv_i that are larger than the tuning parameter $\lambda \in \{0, 0.01, 0.02, \dots, 0.95\}$. We then fit a natural cubic spline \hat{f} with 3 degrees of freedom on Equation 2.47 and set the estimate of $\hat{\pi}_0 = \hat{f}(1)$. With this estimate of $\hat{\pi}_0$ we can compute all q-value estimates in a step down procedure $\hat{q}(pv_{(i)}), i = (m, m - 1, m - 2, \dots, 1)$ as follows:

$$\hat{q}(pv_{(i)}) = \begin{cases} \hat{\pi}_0 pv_{(m)} & \text{if } i = m, \\ \min\left(\frac{\hat{\pi}_0 m pv_{(i)}}{i}, \hat{q}(pv_{(i+1)})\right) & \text{if } i < m. \end{cases} \quad (2.48)$$

2.3 Meta-Analysis Methods for GWASs

Meta-analysis is a technique to combine the results from several already conducted GWASs and is a powerful technique to increase the statistical power to detect genetic associations of complex diseases or phenotypes [Evangelou and Ioannidis, 2013]. A variety of different meta-analysis methods exist. In this section, we give a brief overview about commonly used frequentist methods for meta-analysis in the field of GWASs. Further, we highlight their advantages and disadvantages. All methods are integrated into our common `easyGWAScore` framework.

2.3.1 Fisher's Method

Fisher's method is one of the earliest meta-analysis methods that combines p-values from different studies with the same null-hypothesis [Fisher, 1934; Mosteller and Fisher, 1948]. With Fisher's method we test the null-hypothesis that the effect size is zero in all studies. The alternative hypothesis is true if at least one of the individual null hypothesis can be rejected [Borenstein et al., 2011]. In the field of GWASs we can use this method to combine the p-values of different studies. Fisher's method can be used for all one-sided p-values. The test statistic follows a χ^2 distribution under the global null hypothesis with $2k$ degrees of freedom and is defined as follows:

$$T_{\text{Fisher}} = -2 \sum_{j=1}^k \ln(pv_{ij}), \quad T_{\text{Fisher}} \sim \chi_{2k}^2, \quad (2.49)$$

where k is the number of GWA studies and pv_{ij} is the p-value of the i th genetic marker of j th study. Fisher's method is easy to use and requires only limit information from each study. However, all studies are weighted equally, as we can see in Equation 2.49. This is suboptimal for GWASs with different sample sizes. A second disadvantage

of Fisher's method is that it does not consider the direction of the effect. Thus, associations in different studies with opposite effect directions might strengthen the effect rather than contradicting each other.

2.3.2 Stouffer's Z

Stouffer's method is closely related to Fisher's method but is based on Z-scores rather than p-values [Stouffer *et al.*, 1949]. Stouffer's method for the genetic marker i is defined as:

$$Z_i^{\text{Stouffer}} = \frac{\sum_{j=1}^k (Z_{ij})}{\sqrt{k}}, \quad (2.50)$$

where k is the total number of studies and Z_{ij} is the i th Z-score of study j . The Z-score Z_{ij} can be derived from one-tailed p-values:

$$Z_{ij} = \phi^{-1}(1 - \text{pv}_{ij}), \quad (2.51)$$

where ϕ is the standard normal cumulative distribution function and pv_{ij} is the one-tailed p-value of the i th genetic marker of study j . Similar to Fisher's method we can test the null-hypothesis that the effect for the i th genetic marker is zero in all studies. For this purpose, we can compute one-sided and two-sided p-values for the Z_i^{Stouffer} -test statistic in Equation 2.50:

$$\text{One-sided p-value: } \text{pv}_i = 1 - \phi(|Z_i^{\text{Stouffer}}|), \quad (2.52)$$

$$\text{Two-sided p-value: } \text{pv}_i = 2(1 - \phi(|Z_i^{\text{Stouffer}}|)). \quad (2.53)$$

Stouffer's method can easily be extended by a weighting term w_{ij} to penalise studies differently, that is:

$$Z_i^{\text{Stouffer}} = \frac{\sum_{j=1}^k w_{ij} Z_{ij}}{\sqrt{\sum_{j=1}^k w_{ij}^2}}, \quad (2.54)$$

where w_{ij} is the weight of the i th genetic marker for study j . For GWASs w_{ij} is in general $\sqrt{n_{ij}}$, where n_{ij} is the number of samples/individuals for the i th genetic marker of study j .

A second advantage of Stouffer's method is that it is easy to introduce the direction of the effect, as well. This can be achieved by multiplying the Z-score in Equation 2.51 with the sign of the effect β_{ij} of the i th genetic marker of study j :

$$Z_{ij} = \phi^{-1}(1 - \text{pv}_{ij}) \text{sign}(\beta_{ij}). \quad (2.55)$$

2.3.3 Fixed Effect Model for Meta-Analysis

An alternative strategy to combine p-values or Z-scores is to combine effect sizes. Combining effect sizes is more powerful than combining p-values or Z-scores [Borenstein *et al.*, 2010, 2011]. However, it requires that the computations are standardised across

all studies, e.g. that they are transformed with the same methods or measured on the same scale. However, for large GWAS studies this is not always possible. In that case, one has to use one of the latter methods. Otherwise, Fixed Effect Models (FEM) can be used to combine the effect sizes for different studies. Here we assume that all k GWA studies share a common true effect θ_i for the genetic marker i . For example, we can think of combining different GWASs for a common phenotype that were measured in the same lab, by the same person and under exactly the same environmental conditions. As in *Borenstein et al.* [2010, 2011] we illustrate the shared common true effect with a triangle and the individual (within-study) true effect for study j with a circle (Figure 2.4). In practice the common true effect size θ_i for the genetic marker i is unknown due to its within-study noise (or sampling error). If we assume an infinite number of samples the sampling error (within-study noise) would be the zero. Thus, the observed effect for each study would be the same as the true effect. However, the observed effect Y_{ij} for the genetic marker i and study j (illustrated as squares in Figure 2.4) deviates from the true effect and is defined as the sum of the common true effect θ_i and its within-study noise ϵ_{ij} , that is:

$$Y_{ij} = \theta_i + \epsilon_{ij}, \quad (2.56)$$

where i is the i th genetic marker of study j . The within-study variance of the genetic marker i for study j is denoted as σ_{ij}^2 as illustrated in Figure 2.4.

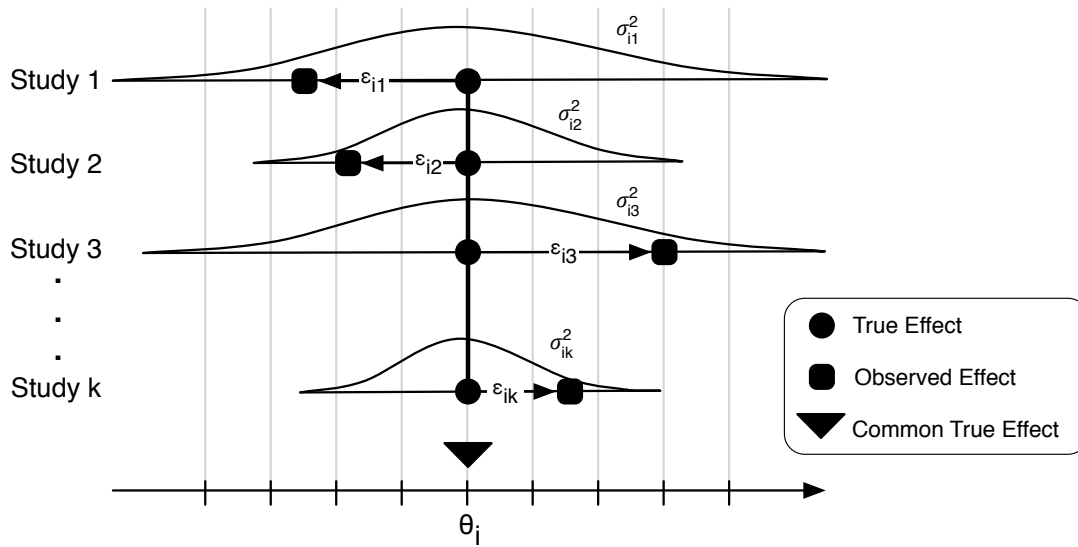


Figure 2.4: Fixed effect model. The Fixed Effect Model assumes a common true effect of the genetic marker i across all studies k . The circle represents the true effect for study j . The square is the observed effect for study j and ϵ_{ij} is the random noise (sampling error) of the genetic marker i for study j . σ_{ij}^2 is the variance of the genetic marker i for study j . The triangle is the estimated common true effect across all k studies.

Estimation of the common true effect θ_i : To estimate the unknown common true effect $\hat{\theta}_i$ we compute a weighted mean of all observed effect sizes Y_{ij} , that is:

$$\hat{\theta}_i = \frac{\sum_{j=1}^k w_{ij} Y_{ij}}{\sum_{j=1}^k w_{ij}}, \quad (2.57)$$

where w_{ij} is the weight assigned to the genetic marker i in study j . For the weighting we use the inverse of the within-study variance σ_{ij}^2 , that is:

$$w_{ij} = \frac{1}{\sigma_{ij}^2}. \quad (2.58)$$

Hence, studies with a more precise within-study variance are weighted stronger than studies with less precise within-study variances. Thus, the inverse of the variance is approximately proportional to the sample size [Borenstein *et al.*, 2010, 2011]. The estimated variance $\hat{\sigma}_{\hat{\theta}_i}^2$ of the estimated common true effect $\hat{\theta}_i$ can be computed as follows:

$$\hat{\sigma}_{\hat{\theta}_i}^2 = \frac{1}{\sum_{j=1}^k w_{ij}}. \quad (2.59)$$

Hence, the estimated standard error $\hat{\sigma}_{\hat{\theta}_i}$ is:

$$\hat{\sigma}_{\hat{\theta}_i} = \sqrt{\hat{\sigma}_{\hat{\theta}_i}^2} = \sqrt{\frac{1}{\sum_{j=1}^k w_{ij}}}. \quad (2.60)$$

With the estimates $\hat{\theta}_i$ and $\hat{\sigma}_{\hat{\theta}_i}$ we easily can derive the 95% upper and lower confidence interval estimates under the assumption of normality for the estimated common true effect:

$$\text{Upper Limit}_{\hat{\theta}_i, 95\%} = \hat{\theta}_i + 1.96\hat{\sigma}_{\hat{\theta}_i}, \quad (2.61)$$

$$\text{Lower Limit}_{\hat{\theta}_i, 95\%} = \hat{\theta}_i - 1.96\hat{\sigma}_{\hat{\theta}_i}. \quad (2.62)$$

Hypothesis testing: With the estimated parameters we can perform a statistical test to test the null hypothesis that the estimated common true effect $\hat{\theta}_i$ is zero. For this purpose, we compute a Z-Value, that is:

$$Z_i^{\hat{\theta}_i} = \frac{\hat{\theta}_i}{\hat{\sigma}_{\hat{\theta}_i}}. \quad (2.63)$$

Similar to other Z-Value based method we easily can derive one-sided and two-sided p-values as follows:

$$\text{One-sided p-value: } \text{pv}_i = 1 - \phi(|Z_i^{\hat{\theta}_i}|), \quad (2.64)$$

$$\text{Two-sided p-value: } \text{pv}_i = 2(1 - \phi(|Z_i^{\hat{\theta}_i}|)), \quad (2.65)$$

where ϕ is the standard normal cumulative distribution function and pv_i is the p-value of the i th genetic marker.

2.3.4 Random Effect Model for Meta-Analysis

For the Random Effect Model (REM) we do not make the assumption that all studies share exactly the same effect size. Here, we assume that the true effect μ_i for the

genetic marker i is approximately Gaussian distributed and is the mean among all individual true study effects θ_{ij} , where j is the j th study of all k studies [Borenstein et al., 2010, 2011]. For example, it is unrealistic to assume that all experiments are conducted exactly under the same conditions and that no variation is going on between the individual studies. Most often it is the case that we like to combine GWASs computed on a common phenotype that was measured in different labs, by different scientists and approximately under the same environmental conditions. Thus, each individual true effect θ_{ij} of study j is an additive combination by the mean μ_i of all studies and the between-study variation τ_{ij} (also between-study noise or study heterogeneity) [Borenstein et al., 2010, 2011]. In Figure 2.5 we illustrated the basic concepts of a REM. Further, we define the observed effect size Y_{ij} in a REM as an

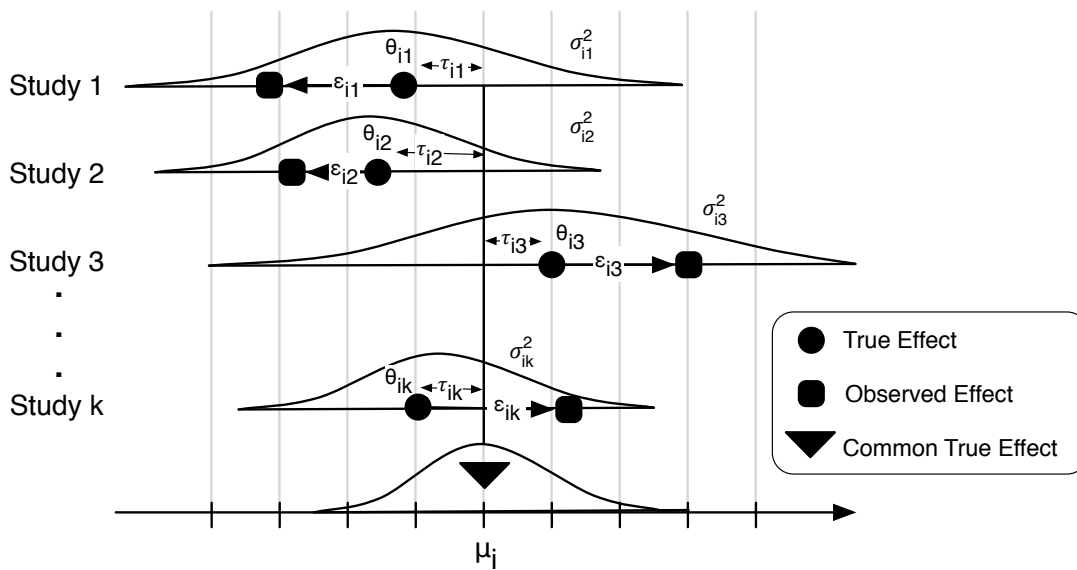


Figure 2.5: Random effect model. The Random Effect Model assumes a within-study and between-study variance for genetic marker i across all studies k . The circle represents the true effect θ_{ij} for study j . The square is the observed effect for study j , ϵ_{ij} is the random noise (sampling error) and τ_{ij} the true variation in effect sizes of the genetic marker i for study j . σ_{ij}^2 is the variance of the genetic marker i for study j . The triangle is the estimated mean of the population effect across all k studies.

additive contribution of the mean of all true effects μ_i , the within-study noise ϵ_{ij} and the between-study variation τ_{ij} [Borenstein et al., 2010, 2011], as follows:

$$Y_{ij} = \theta_{ij} + \epsilon_{ij} = \underbrace{\mu_i}_{\text{mean true effect size}} + \underbrace{\tau_{ij}}_{\text{between-study noise}} + \underbrace{\epsilon_{ij}}_{\text{within-study noise}}. \quad (2.66)$$

Estimation of the overall population between-studies variance T^2 : To estimate the between-studies variance \hat{T}^2 across a whole population of studies we can use the method after DerSimonian and Laird [1986], as described in Borenstein et al. [2010, 2011], that is:

$$\hat{T}_i^2 = \frac{Q_i - df}{C_i}, \quad (2.67)$$

where Q_i is the total variance, df the expected variance if all studies share the same true effect (degrees of freedom) and C_i is a scaling factor. Q_i is defined as follows:

$$Q_i = \sum_{j=1}^k (w_{ij} Y_{ij}^2) - \frac{\left(\sum_{j=1}^k w_{ij} Y_{ij} \right)^2}{\sum_{j=1}^k w_{ij}}, \quad (2.68)$$

where w_{ij} is the inverse within-study variance as defined in Equation 2.58. The expected variance is defined as:

$$df = k - 1, \quad (2.69)$$

where k is the total number of studies. Thus, if the total variance Q is equal to the expected variance df , no between-study variance can be observed. The scaling factor C is computed as follows:

$$C_i = \sum_{j=1}^k w_{ij} - \frac{\sum_{j=1}^k w_{ij}^2}{\sum_{j=1}^k w_{ij}}. \quad (2.70)$$

Estimation of the mean effect size μ_i : In a REM we estimate the mean of the true effect $\hat{\mu}_i$, by computing a weighted mean of all individual observed effect sizes Y_{ij} , that is:

$$\hat{\mu}_i = \frac{\sum_{j=1}^k \widetilde{w}_{ij} Y_{ij}}{\sum_{j=1}^k \widetilde{w}_{ij}}, \quad (2.71)$$

where \widetilde{w}_{ij} is the additive inverse of the within-study variance σ_{ij}^2 and the estimated between-studies variance \hat{T}^2 , that is:

$$\widetilde{w}_{ij} = \frac{1}{\sigma_{ij}^2 + \hat{T}_i^2}. \quad (2.72)$$

The estimated variance $\hat{\sigma}_{\hat{\mu}_i}^2$ and the estimated standard error $\hat{\sigma}_{\hat{\mu}_i}$ of the estimated mean of the true effect $\hat{\mu}_i$ can be computed as follows:

$$\hat{\sigma}_{\hat{\mu}_i}^2 = \frac{1}{\sum_{j=1}^k \widetilde{w}_{ij}}, \quad (2.73)$$

$$\hat{\sigma}_{\hat{\mu}_i} = \sqrt{\hat{\sigma}_{\hat{\mu}_i}^2} = \sqrt{\frac{1}{\sum_{j=1}^k \widetilde{w}_{ij}}}. \quad (2.74)$$

With these estimates we easily can derive the 95% upper and lower confidence interval estimates under the assumption of normality:

$$\text{Upper Limit}_{\hat{\mu}_i, 95\%} = \hat{\mu}_i + 1.96 \hat{\sigma}_{\hat{\mu}_i}, \quad (2.75)$$

$$\text{Lower Limit}_{\hat{\mu}_i, 95\%} = \hat{\mu}_i - 1.96 \hat{\sigma}_{\hat{\mu}_i}. \quad (2.76)$$

Hypothesis testing: Similarly to the FEM, we can perform a statistical test to test the null hypothesis that the mean of the estimated effect $\hat{\mu}_i$ is zero. As in Equation 2.63 we compute the Z-Value $Z_i^{\hat{\mu}_i}$ as the ratio of the estimated mean effect $\hat{\mu}_i$ to the

estimated standard error $\hat{\sigma}_{\hat{\mu}_i}$. The one-sided and two-sided p-values are then computed as follows:

$$\text{One-sided p-value: } pv_i = 1 - \phi(|Z_i^{\hat{\mu}_i}|), \quad (2.77)$$

$$\text{Two-sided p-value: } pv_i = 2(1 - \phi(|Z_i^{\hat{\mu}_i}|)), \quad (2.78)$$

where ϕ is the standard normal cumulative distribution function and pv_i is the p-value of the i th genetic marker.

2.4 easyGWASCore: An Efficient C/C++ Framework for GWASs and Meta-Analyses

In the following subsections we will describe **easyGWASCore**, a C/C++ framework with **Python** interfaces, that integrates the previously introduced regression and meta-analysis methods and offers a common data pre- and post-processing pipeline. First, we will characterise the architecture of the **easyGWASCore** framework, followed with a brief explanation of the application programming interface and a demonstration of its flexibility on several examples. Second, we will introduce the **Python** command line interface by conducting an example GWAS on *Arabidopsis thaliana*, including the visualisation and annotation of the results. Finally, we will analyse the performance of our framework and compare it to well established state-of-the-art tools.

2.4.1 The Architecture and Design of easyGWASCore

We used the high level programming language C/C++ to implement the main algorithms and methods of **easyGWASCore**. We used **SWIG**¹ a simplified wrapper and interface generator to create **Python** interfaces for our C/C++ algorithms and methods. We structured the framework into three main abstraction layers, as illustrated in Figure 2.6. Here, a layer represents a collection of at least one module, whereas a module represents a collection of at least on class object. **Layer 1** is an assembly of different *core* modules, such as a general library of basic algorithms and helper methods that are required by many other methods and algorithms. Thus far, the **easyGWASCore** framework contains five main core modules, providing a collection of basic statistical classes (e.g. different distribution functions), input/output (*IO*) management classes (e.g. log file writers, file progress class), basic helper classes (e.g. string helper class, math helper class, cross-validation class), kernel function classes (e.g. realised relationship kernel) and an optimisation class (e.g. root finding algorithms), as illustrated in Figure 2.6. The second layer, **Layer 2**, is a collection of generic and widely used modules for various algorithms and methods. As illustrated in Figure 2.6, the framework contains a *regression* and a *meta* module. These two modules are an assembly of commonly used implementations of different regression models (e.g. linear, logistic or linear mixed model regression), as well as standard meta-analysis methods. Algorithms in **Layer 2**

¹<http://www.swig.org>

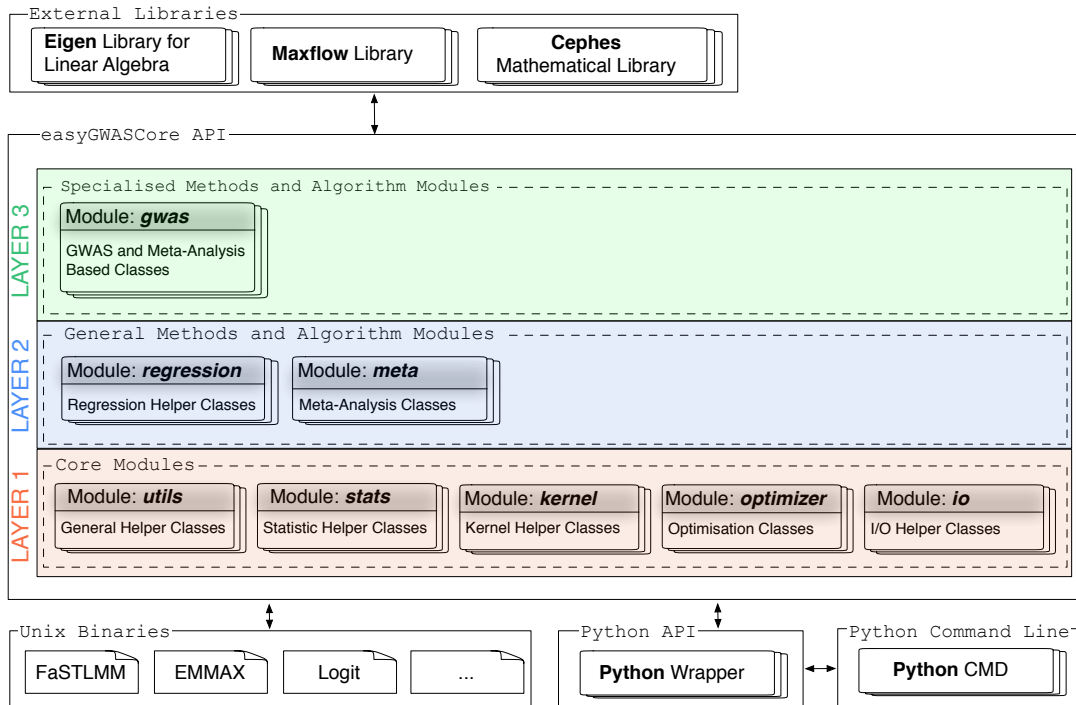


Figure 2.6: Layers and modules of easyGWASCore: easyGWASCore is structured into three main layers. The first layer contains core modules that are needed by a variety of algorithms and methods. The second layer contains modules that represent a general class of algorithms and methods that can be applied to solve many different problems. The last layer contains modules tailored to certain tasks that mostly use methods or algorithms from the first two layers.

are supposed to be as general as possible such that they can be easily re-used in as many applications and methods as possible. Methods from this layer make frequent use of modules from Layer 1.

The last layer, Layer 3, contains specialised algorithms and methods tailored to solve specific problems. Until now, the easyGWASCore framework contains a specialised module for performing GWASs and meta-analyses. Algorithms in this layer frequently use modules and classes from the first two layers. A detailed overview about all classes and function can be found in the Appendix D.1.

In addition, our framework uses different popular third party libraries. To perform any kind of matrix or vector arithmetic we linked the linear algebra library **Eigen** [Guennebaud et al., 2010]. This library also provides helpful numerical solvers (e.g. basic linear solvers, eigenvalue and eigenvector decomposition methods) that are frequently used in easyGWASCore. For many statistical test and methods, as well as for computing p-values different distribution functions are needed. Therefore, we created C/C++ interfaces to the **Ceph** mathematical library². This library offers different algorithms to precisely compute the **Cumulative Distribution Functions** (CDF) or **Survival Functions** (SF) of various statistical distributions. Furthermore, we included the **Maxflow** library [Boykov and Kolmogorov, 2004], that efficiently computes a max-flow/min-cut on arbitrary graphs using the Boykov-Kolmogorov algorithm. This library is used by our algorithm **SConES**, which we will introduce in Chapter 5.

²<http://www.moshier.net>

The use of Object Oriented Programming (OOP) paradigms and the modular structure of our C/C++ framework facilitates the development and integration of novel algorithms. For example we used class inheritance to create a common object structure for related classes and methods. In our framework all regression classes — `CLinearRegression`, `CLogisticRegression` and `CLinearMixedRegression` (see Appendix D.1) — inherit from the global parent class `CRegression`. Consequently, all regression models share a common set of methods. This also facilitates the re-use of existing code since existing algorithms can be easily modified by simply changing the type of regression model. In

Listing 2.1: C/C++ example of fitting a linear regression and logistic regression and printing the output of each model using a function that takes a pointer to the parent class of these models

```

1 #include <iostream>
2 //Include regression class
3 #include "CEasyGWAS/regression/CRegression.h"
4
5 /*Print some summary of the regression model to demonstrate that CRegression
6 *is the parent class of the CLinearRegression and CLogisticRegression class.*/
7 void printSummary(CRegression* regression) {
8     //Print results from regression model
9     regression->print();
10    //Print R2 measure
11    std::cout << "R2:\t" << regression->getRSquared() << std::endl;
12    //Print degrees of freedom
13    std::cout << "DF:\t" << regression->getDF() << std::endl;
14 }
15
16 int main() {
17     //Initialise a Random Matrix with 12 samples and 2 features
18     MatrixXd X = MatrixXd::Random(12,2);
19     //Initialise target vector with 12 samples
20     VectorXd y(12);
21     y << 1,2,3,4.5,5,6,7.5,8,9,10.5,11,12;
22     //Create a Linear Regression
23     CLinearRegression lm;
24     //Fit Linear Regression
25     lm.fit(y,X);
26     //Print something to illustrate inheritance
27     printSummary(&lm);
28
29     //Initialise binary target vector with 12 samples
30     VectorXd y_binary(12);
31     y << 0,0,0,0,1,0,1,1,1,0,1,1;
32     //Create a Logistic Regression
33     CLogisticRegression lg;
34     //Fit Logistic Regression
35     lg.fit(y,X);
36     //Print something to illustrate inheritance
37     printSummary(&lg);
38
39     return 0;
40 }

```

Listing 2.1 we give a basic example of how to run a linear and logistic regression using the `easyGWASCore` API. We therefore initialise and fit a linear regression in Lines 22-25 and logistic regression in Lines 32-35 (Listing 2.1). To demonstrate the inheritance of common methods from the parent class `CRegression`, we implement a simple function to print some parameters for the respective regression class. As function parameter we pass a pointer with type `CRegression`. Since this is the parent class of all regression based classes we can pass a reference to an object of type `CLinearRegression`, as well

as an object of type `CLogisticRegression`. The command line output for the example in Listing 2.1 is shown in Listing 2.2.

Listing 2.2: Output for example in Listing 2.1

```

1
2 > Output Linear Regression:
3
4
5 Linear Regression: y ~ 1 + x1 + x2
6   Estimated Parameters:
7       Betas      STD Betas
8   (Intercept) 6.86767    1.14805
9     x1        -1.43343    2.6427
10    x2         -1.43987    2.00449
11
12   LogLikelihood: -31.24
13   AIC:           68.4801
14   AICc:          71.4801
15   BIC:           69.9348
16   R2:            0.148831
17   DF:            3

```

```

> Output Logistic Regression:
Logistic Regression: y ~ 1 + x1 + x2
  Estimated Parameters:
      Betas      STD Betas
(Intercept) -0.117562  0.691814
x1           0.762114  1.58639
x2          -1.59986  1.22998
LogLikelihood: -7.3333
AIC:            20.6666
AICc:           23.6666
BIC:            22.1213
R2:             4.00972
DF:             3

```

2.4.2 The easyGWASCore Application Programming Interface

In this section we will describe the application programming interface (API) of the `easyGWASCore` framework. In the first sub-section we will give a brief and general overview about the `C/C++` and `Python` API and how those can be applied. In addition, we will give a concrete and fully functional example of how to perform a GWAS by exploiting the `C/C++` API. In the last sub-section we will demonstrate how the `easyGWASCore` API can be used to extend the framework with additional user-specific models.

General Overview About the `easyGWASCore` API

In the following we will demonstrate how to execute four different state-of-the-art GWAS algorithms using either the `C/C++` or `Python` API. For this purpose, let us assume that we are given an already pre-processed SNP matrix \mathbf{G} , a phenotype \mathbf{y} and a sample by sample kinship matrix \mathbf{K} . A side by side comparison between the `C/C++` and `Python` method calls can be found in Listing 2.3 and Listings 2.4, respectively. A complete list of all available algorithms and methods can be found in the Appendix D.1. The syntax — as illustrated — is nearly identical within and across these two different programming languages. The same is true for the meta-analysis methods. Let us assume we already conducted two GWASs I and II . For each of those two studies we are given a set of p-values \mathbf{pv}_j with their corresponding positions \mathbf{pos}_j , chromosomes \mathbf{chrom}_j and if available their corresponding estimated regression coefficients $\hat{\beta}_j$ and estimated standard errors of the regression coefficients $\hat{\beta}_{SE_j}$, where j is either study I or study II .

Listing 2.3: GWAS C/C++

```

1 //include methods for single trait GWAS
2 #include "CEasyGWAS/gwas/CSingleTraitGWAS.h"
3 //Linear Regression
4 CSingleTraitGWAS::LinearRegression lm;
5 lm.setGenotype(G);
6 lm.setPhenotype(y);
7 lm.test_associations();
8 lm.getPValues();
9
10 //Logistic Regression
11 CSingleTraitGWAS::LogisticRegression lr;
12 lr.setGenotype(G);
13 lr.setPhenotype(y);
14 lr.test_associations();
15 lr.getPValues();
16
17 //EMMAX
18 CSingleTraitGWAS::EMMAX emmax;
19 emmax.setGenotype(G);
20 emmax.setPhenotype(y);
21 emmax.setK(K);
22 emmax.test_associations();
23 emmax.getPValues();
24
25 //FaSTLMM
26 CSingleTraitGWAS::FaSTLMM fastlmm;
27 fastlmm.setGenotype(G);
28 fastlmm.setPhenotype(y);
29 fastlmm.setK(K);
30 fastlmm.test_associations();
31 fastlmm.getPValues();

```

Listing 2.4: GWAS Python

```

1 #Include easyGWASCore
2 import easyGWASCore as gwas_core
3 #Linear Regression
4 lm = gwas_core.LinearRegression()
5 lm.setGenotype(G)
6 lm.setPhenotype(y)
7 lm.test_associations()
8 lm.getPValues()
9
10 #Logistic Regression
11 lr = gwas_core.LogisticRegression()
12 lr.setGenotype(G)
13 lr.setPhenotype(y)
14 lr.test_associations()
15 lr.getPValues()
16
17 #EMMAX
18 emmax = gwas_core.EMMAX()
19 emmax.setGenotype(G)
20 emmax.setPhenotype(y)
21 emmax.setK(K)
22 emmax.test_associations()
23 emmax.getPValues()
24
25 #FaSTLMM
26 fastlmm = gwas_core.FaSTLMM()
27 fastlmm.setGenotype(G)
28 fastlmm.setPhenotype(y)
29 fastlmm.setK(K)
30 fastlmm.test_associations()
31 fastlmm.getPValues()

```

Listing 2.5: Meta-analysis C/C++

```

1 //include methods for meta-analysis
2 #include "CEasyGWAS/gwas/CMetaGWAS.h"
3 //Perform Fisher Method
4 CMetaGWAS meta;
5 //Add study I
6 meta.addPValuesStudy( $p_I$ ,  $chrom_I$ ,  $pos_I$ );
7 //Add study II
8 meta.addPValuesStudy( $p_{II}$ ,  $chrom_{II}$ ,  $pos_{II}$ );
9 //Compute
10 meta.performFisherMethod();
11 meta.getPValues();
12
13 //Perform Fixed Effect Model analysis
14 CMetaGWAS meta;
15 //Add study I
16 meta.addEffectSizeStudy( $chrom_I$ ,  $pos_I$ ,
17  $\hat{\beta}_I$ ,  $\hat{\beta}_{SEI}$ );
18 //Add study II
19 meta.addEffectSizeStudy( $chrom_{II}$ ,  $pos_{II}$ ,
20  $\hat{\beta}_{II}$ ,  $\hat{\beta}_{SEII}$ );
21 //Compute
22 meta.performFixedEffectModel();
23 meta.getPValues();

```

Listing 2.6: Meta-analysis Python

```

1 #Include easyGWASCore
2 import easyGWASCore as gwas_core
3 #Perform Fisher Method
4 meta = gwas_core.CMetaGWAS()
5 #Add study I
6 meta.addPValuesStudy( $p_I$ ,  $chrom_I$ ,  $pos_I$ )
7 #Add study II
8 meta.addPValuesStudy( $p_{II}$ ,  $chrom_{II}$ ,  $pos_{II}$ )
9 #Compute
10 meta.performFisherMethod()
11 meta.getPValues()
12
13 #Perform Fixed Effect Model analysis
14 meta = gwas_core.CMetaGWAS()
15 #Add study I
16 meta.addEffectSizeStudy( $chrom_I$ ,  $pos_I$ ,
17  $\hat{\beta}_I$ ,  $\hat{\beta}_{SEI}$ )
18 #Add study II
19 meta.addEffectSizeStudy( $chrom_{II}$ ,  $pos_{II}$ ,
20  $\hat{\beta}_{II}$ ,  $\hat{\beta}_{SEII}$ )
21 #Compute
22 meta.performFixedEffectModel()
23 meta.getPValues()

```

In the Listings 2.5 and 2.6 we give two examples for two different types of meta-analyses. In the first example we use Fisher's Method and for the second one a Fixed Effect Model. An overview about all other algorithms can be found in the Appendix D.1.

GWAS Example Using the C/C++ API

Here, we will illustrate how to create a standalone program to perform a complete GWAS using a linear regression model. For this purpose, we give a line by line explanation of how to parse the genotype and phenotype data, as well as how to perform the actual GWAS (see Listing 2.7).

Listing 2.7: C/C++ example for parsing, filtering and handling data, as well as performing a genome-wide association mapping using a linear regression

```

1 //Include PLINK parser class
2 #include "CEasyGWAS/io/CPLinkParser.h"
3
4 //Include GWAS data Input, Output class
5 #include "CEasyGWAS/io/CGWASDataIO.h"
6
7 //include methods for single trait GWAS
8 #include "CEasyGWAS/gwas/CSingleTraitGWAS.h"
9
10 //Main function for performing a linear regression including parsing
11 //and filtering genotype and phenotype data
12 int main() {
13     //Initialise data object for gwas
14     GWASData data;
15
16     //Read genotype PED file with the PLINK Parser and store data in data object
17     CPLinkParser::readPEDFile("genotype.ped",&data);
18
19     //Read genotype MAP file and store data in data object
20     CPLinkParser::readMAPFile("genotype.map",&data);
21
22     //Read phenotypes from file and store them in data object
23     CPLinkParser::readPhenotypeFile("phenotype.pheno",&data);
24
25     //GWAS data Helper to encode genotype data with a heterozygous encoding
26     CGWASDataHelper::encodeHeterozygousData(&data, GWASDataHelper::additive);
27
28     //Select phenotype 0 and remove samples with missing data
29     //and dump data in a new temporary /data object. This
30     //command is slicing and sorting the data accordingly
31     //such that samples match.
32     GWASData tmpData = CGWASDataHelper::removeSamples4MissingData(data, 0);
33
34     //Remove SNPs with a minor allele frequency less than 10%
35     CGWASDataHelper::filterSNPsByMAF(&tmpData, 0.1);
36
37     //Perform GWAS using a Linear Regression
38     CSingleTraitGWAS::LinearRegression gwas(tmpData.Y.col(0), tmpData.X);
39     gwas.test_associations();
40
41     ///Get results and save them in an output file
42     GWASResults results = gwas.getResults();
43     CGWASDataIO::writeSummaryOutput("output.txt", tmpData, results);
44
45     return 0;
46 }

```

The first step is to generate a GWAS data object by initialising the `GWASData` class (Line

14, Listing 2.7). Next we are able to load the genotype and phenotype data — stored in the popular PLINK [Purcell *et al.*, 2007] format — by using the input/output methods from the `CPlinkParser` class (Line 17-23, Listing 2.7). Note that the data loading methods take care of matching the samples from the genotype data to those of the phenotype data. After we successfully loaded the raw genotype data we need to encode this data using one of the commonly available genotype encodings (Appendix C.2). In this example we apply the standard additive genotype encoding where the major allele of a given SNP is encoded with 0, the heterozygous allele with 1 and the minor allele with 2 (Line 26, Listing 2.7). After loading and encoding the data we have to select one of the phenotypes that are stored in our data object and remove samples with missing phenotypic values. This can be achieved by using the `removeSamples4MissingData(.)` method. We therefore pass the current data object to this method and specify that the phenotype with id 0 should be selected. Thus, the phenotype data is stored in a filtered and sorted manner in a temporary data object (Line 32). This is necessary because the original data object might contain more than one phenotype and we do not want to modify the original data container. To remove SNPs with a minor allele frequency of less than 10% we use the `filterSNPsByMAF` method (Line 35). Finally, we can perform the actual genome-wide association scan by initialising a single trait linear regression using the `CSingleTraitGWAS` class (Line 38-39). Again, the interface is similar for all other single trait regression models. Thus it is straightforward to replace the linear regression with a logistic regression, `EMMAX` [Kang *et al.*, 2010] or `FaSTLMM` [Lippert *et al.*, 2011]. Finally, we retrieve the results by writing them to an output file (Line 42-43).

Building Novel Algorithms with the easyGWASCore API

In the previous sub-sections we showed the general structure of the API for performing GWASs and meta-analysis. However, often it is necessary to create more complex models that are not supported or implemented out of the box. In this section we demonstrate on a small example how to use the `easyGWASCore` API to create a more complex model to investigate epistatic effects, that is the multiplicative effect between two genetic markers. Let us assume we want to test whether the interaction effect between two genetic markers $\{\mathbf{g}_1, \mathbf{g}_2\}$ is significantly associated with a given phenotype \mathbf{y} . For this purpose, we first have to create the null model that only contains the intercept, that is:

$$\mathbf{y} = \beta_0, \quad (2.79)$$

and the alternative model that includes the interaction effect between those two markers, that is:

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{g}_1 \mathbf{g}_2. \quad (2.80)$$

Using the `easyGWASCore` API we first have to initialise and fit the null model as shown in Lines 13-15 (Listing 2.8). Next, we fit a second linear regression model including the interaction term $\mathbf{g}_1 \mathbf{g}_2$ (Lines 17-20). Finally we can conduct a log-likelihood ratio

test and compute a p-value using the survival function of the χ^2 distribution with one degree of freedom (Lines 23-24).

Listing 2.8: Example of how to use the API to create a model for a two-locus association test

```

1 #include "CEasyGWAS/regression/CRegression.h"
2 #include "CEasyGWAS/stats/CChi2.h"
3
4 int main() {
5     //Initialise a Random SNP with 12 samples
6     VectorXd g1 = VectorXd::Random(12);
7     //Initialise a Random SNP with 12 samples
8     VectorXd g2 = VectorXd::Random(12);
9     //Initialise a Random phenotype with 12 samples
10    VectorXd y = VectorXd::Random(12);
11
12    //Initialise linear regression for null model
13    CLinearRegression null_model(false);
14    //fit null model with intercept only
15    null_model.fit(y, VectorXd::Ones(y.rows()));
16
17    //Initialise linear regression for alternative model
18    CLinearRegression alt_model;
19    //fit alternative model with interaction effect between SNP g1 and SNP g2
20    alt_model.fit(y, g1.array()*g2.array());
21
22    //Compute p-value using LogLikelihood Ratio Test with 1 degree of freedom
23    float p_value = CChi2::sf(2*(alt_model.getLogLikelihood() -
24                                null_model.getLogLikelihood()),1);
25    return 0;
26 }

```

2.4.3 The Python Command Line Interface of easyGWASCore

Brief Overview About the Command Line Interface

In the previous section we described the API of the `easyGWASCore` framework. However, the API is mainly intended for developers and bioinformaticians to integrate the framework into their own pipelines and tools. Because of that, we developed an easy-to-use Python command line interface to conduct GWASs from the start to the end. The command line interface offers different data handling methods, algorithms for performing GWASs, as well as methods for visualising and annotating results. We structured the interface into three main parts: one for data handling related tasks (`data` command), a second one for GWASs related tasks (`gwas` command) and a third one for visualisation and annotation related tasks (`plot` command). The help function of the command line tool provides an overview of all these options (see Listing 2.9).

Efficient data loading and storing functionality, as well as storing the data in a structured and easy accessible format are crucial for big GWAS projects. We here use the Hierarchical Data Format 5 (HDF5). This data format was originally developed by the National Center for Supercomputing Applications and established as the standard data format by the NASA³. HDF5 is designed to store, organise and efficiently access huge amounts of data. The command line interface provides routines to convert between the popular PLINK input data files and our tool depended structured HDF5 files

³<https://www.hdfgroup.org/about/history.html>

Listing 2.9: Three main categories for different GWASs related tasks of the command line tool

```

1 $: python python/easygwascore.py -h
2   usage: easygwascore.py [-h] [-v] {plot,gwas,data} ...
3
4   easGWASCore: Performing, visualising and annotating GWASs
5
6   positional arguments:
7     {plot,gwas,data} Subcommands: Please specify the command and use the flag
8                       -h to print the parameters for the different subcommands
9     plot                Create different plots (e.g. Manhattan Plot, QQ-Plot, LD-
10                       Plot etc.! To list all options please use 'plot -h')
11     gwas                Perform a Genome-Wide Association Scan. To list all
12                       available options use 'gwas -h')
13     data                Data processing, converting and manipulation methods. To
14                       list all available options use 'data -h')
15
16   optional arguments:
17     -h, --help          show this help message and exit
18     -v, --version       show program's version number and exit

```

(e.g. `--plink2hdf5` and `--hdf5toplink`). All result files are stored in the HDF5 format as well, but can be exported into a human readable CSV format using the `--csv` parameter. To provide basic annotations for the top x associated hits the sub-command `--agene` can be used to query an available gene annotation file (e.g. in GFF format⁴). In addition, the linkage disequilibrium structure around significantly associated loci can be investigated by using the `-ld` sub-command. A complete list of all *data* handling related sub-commands can be found in the Appendix (Listing D.1).

For conducting the actual GWAS the `gwas` command has to be used. The command line tool supports a variety of different mapping algorithms ranging from a simple linear regression over more complex linear mixed models to permutation based tests (Appendix, Listing D.2). All algorithms are accessible via the `--algorithm` sub-command. Since most models assume Gaussian distributed residuals several options are provided to normalise phenotypes (`--transform {sqrt,log10,boxcox,zeroMean,unitVariance}`). In addition, different genotype encodings (e.g. additive, dominant or recessive encoding) and filtering options (e.g. minor allele frequency filter) are supported. A full list of all command line arguments for performing a GWAS can be found in the Appendix (Listing D.2).

For the the visualisation and annotation of the results the command `plot` has to be used. The sub-command `--manhattan` generates Manhattan plots. A Manhattan plot is a scatter plot, where the X-axis displays the genomic coordinates and the Y-axis the negative logarithm of the p-value for each genetic marker. An example of a Manhattan plot is illustrated in Figure 2.7 (left part). Quantile-Quantile plots (QQ-Plots, `--qqplot`) can be used to investigate whether the computed (observed) distribution of p-values follows approximately the expected distribution of p-values. Under the assumption that the null hypothesis is true, p-values are uniformly distributed. For GWASs we expect that only a few SNPs are significantly associated with the phenotype. Thus, we would expect that most of the computed p-values are uniformly distributed. For this purpose, the negative logarithm of the expected distribution of

⁴<http://www.ensembl.org/info/website/upload/gff.html>

p-values (X-axis) is plotted against the negative logarithm of the sorted list of observed p-values (Y-axis). If the two distributions of p-values is similar to each other then the points in the QQ-Plot will approximately lie on the line $x = y$. An inflation of p-values in the upper quantile of the plot could be an indicator for hidden confounders, such as population stratification. The genomic control (GC) value (or estimated $\hat{\lambda}$ inflation factor) [Devlin and Roeder, 1999] is reported for all QQ-plots to assesses the degree of inflated test statistics by measuring the deviation of the observed median test statistics from the expected one:

$$\hat{\lambda} = \frac{\text{median}(\text{test_statistics})}{\text{median}(\chi^2)} = \frac{\text{median}(\text{test_statistics})}{0.456}, \quad (2.81)$$

where $\hat{\lambda}$ values larger than one are an indicator of inflated test-statistics and values smaller than one are an indicator of deflated test-statistics. An example of a QQ-Plot and the estimated $\hat{\lambda}$ inflation factor is illustrated in Figure 2.7 (right part). The linkage disequilibrium structure of a focal SNP to all other SNPs in close proximity can be generated with the `--ldplot` sub-command. LD plots are a zoomed in version of a Manhattan plot. Different measurements of LD can be computed using the `--r2-measure` flag, including Excoffier-Slatkin [Excoffier and Slatkin, 1995], Roger-Huff [Rogers and Huff, 2009] or Pearson's correlation coefficient. We used the `covld` package from Alan Rogers⁵ to estimate LD [Rogers and Huff, 2009]. Optionally, LD plots can be enriched with gene annotations by using the `--sql_gene` flag. An example of a LD-Plot is illustrated in Figure 2.8. An overview of all plotting sub-commands can be found in the Appendix of this thesis (Listing D.3).

Example of Conducting a GWAS Using the Command Line Interface

In the following we will give an example of how to apply the Python command line interface for conducting a whole GWAS, including data processing, as well as the visualisation and annotation of results. We used genotype and phenotype data from the plant *Arabidopsis thaliana* [Atwell et al., 2010; Horton et al., 2012] and selected the defence related phenotype *avrPphB*. The data is stored in the popular PLINK [Purcell et al., 2007] format and has to be converted into the HDF5 format using the following command line argument:

```
$ python python/easygwascore.py data --plink2hdf5
                                --plink_data example/data/genotype
                                --plink_phenotype example/data/avrPphB.pheno
                                --hout example/data/AtPolyDB.hdf5
```

After converting the data we selected the FaSTLMM algorithm [Lippert et al., 2011] to conduct the GWAS using the standard additive genotype encoding and applied a minor allele frequency filter of 10%:

```
$ python python/easygwascore.py gwas --hdata example/data/AtPolyDB.hdf5
                                --out example/results/
                                --algorithm FaSTLMM
```

⁵<https://github.com/alanrogers/covld>

```
--maf 0.1
```

The computations are finished in approximately half a minute on a standard desktop machine. Next, we visualised the results by generating a basic Manhattan- and QQ-Plot with the following command line argument:

```
$ python python/easygwascore.py plot --manhattan
--qqplot
--estpv
--hfile example/results/
--out example/plots/
```

The Manhattan plot and the corresponding QQ-Plot are illustrated in Figure 2.7. Some

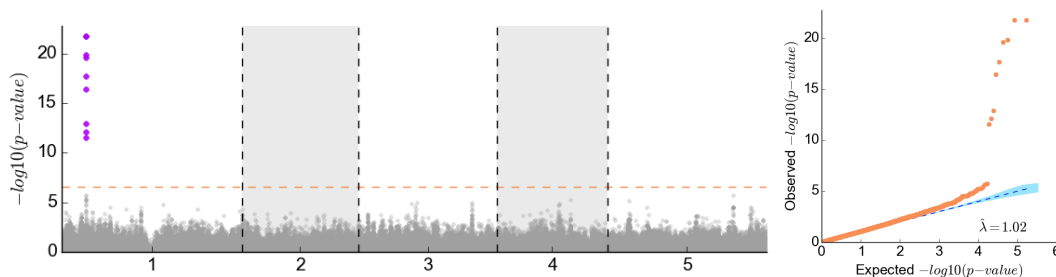


Figure 2.7: Manhattan plot and QQ-plot for the phenotype *avrPphB*: Purple points on the Manhattan Plot (left) indicate that these SNPs are significantly associated after correcting for multiple hypothesis using Bonferroni (red dashed line). The QQ-Plot (right) compares the observed distribution of p-values against the expected distributions using the negative logarithm of the p-values.

SNPs on chromosome 1 are significantly associated with the phenotype after correcting for multiple hypothesis using Bonferroni as illustrated in the Manhattan plot in Figure 2.7. Next, we generated a linkage disequilibrium plots for all significantly associated SNPs to gain more insights about this region. We are interested in the genes and the minor allele frequency of each SNP in this region. For this purpose, we first generated a SQL file from a given gene annotation file (GFF) with the following command:

```
$ python python/easygwascore.py data --gff2sql
--gfile example/data/TAIR10_genes.gff
--sqlout example/data/TAIR10.sql
```

In the second step we generated the LD plots and stored them using the PDF file format:

```
$ python python/easygwascore.py plot --ldplot --hfile example/results/
--hdata example/data/AtPolyDB.hdf5
--sql_gene example/data/TAIR10.sql
--out example/plots/
--maf 0.1
--ifformat pdf
```

One of the generate linkage disequilibrium plots is shown in Figure 2.8. The magenta SNP (Chr 1, Position 4146714) is the selected focal SNP. The strength of LD is illustrated in different colours, where red illustrates strong LD and blue weak LD. We found that all significantly associated SNPs in these region are common SNPs with a minor allele frequency between 0.25 and 0.5. In addition, we found that the focal SNP on chromosome 1 and position 4146714 is located in the gene AT1G12220.

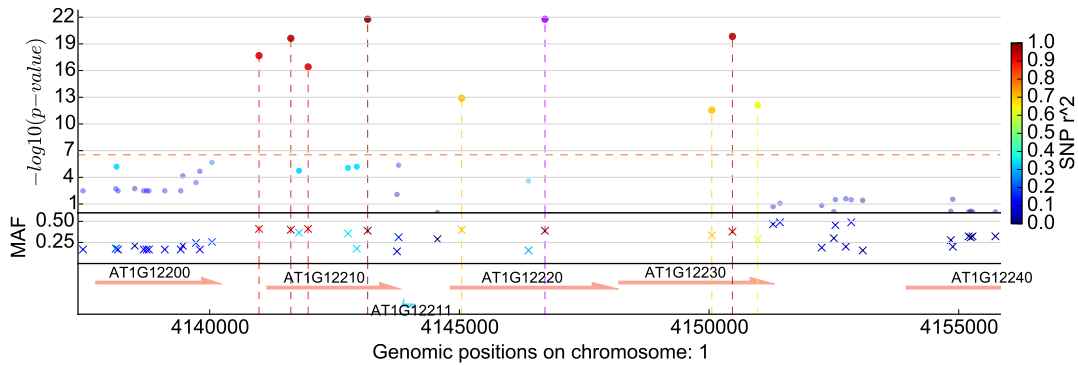


Figure 2.8: Linkage disequilibrium plot for SNP at position 4146714 on Chr 1: The magenta point is the focal SNP. LD is illustrated using different colours. The lower part of the plots gives information about the minor allele frequency of each SNP, as well as the genes in this region.

This gene is a well known resistance gene (R-gene) and is called *RESISTANCE TO PSEUDOMONAS SYRINGAE 5 (RPS5)*.

2.4.4 Performance Analysis

We analysed the performance of four popular GWA tools and methods, including Linear Regression (PLINK v1.0.7 [Purcell et al., 2007]), Logistic Regression (PLINK v1.0.7 [Purcell et al., 2007]), EMMAX [Kang et al., 2010] and FaSTLMM [Lippert et al., 2011], with those implemented in the *easyGWASCore* framework. For the analysis we used real genotype data from the 1001 genomes project in *Arabidopsis thaliana* and generated continuous and dichotomous random phenotypes. For all our experiments we varied the number of SNPs from ten thousand (10k) to five million (5M), as well as the number of samples from 100 to 500. For a fair comparison we used the same data format (PLINK) across all tools. Eventually, we reported the real CPU runtime in seconds over a single AMD Opteron CPU (2048 KB, 2600MHz) with 512GB of memory, running Ubuntu 12.04.5 LTS. In a first analysis we investigated how much of the total runtime is needed for the pre- and post-processing of the data and how much for running the actual algorithm. For this purpose, we executed all four algorithms implemented in the *easyGWASCore* framework and reported the runtime results in Figure 2.9. We observed that data processing takes a significant proportion of the total runtime. For linear regression, data handling is on average one magnitude slower than running the actual algorithm. However, for logistic regression the runtime of the algorithm is between one and two magnitudes slower than processing the data. This is mainly due to the fact, that a logistic regression cannot be solved in closed form. As discussed before, we are using custom implementation of the iterative Newton-Raphson procedure [Ypma, 1995]. However, different or more efficient techniques could be used for solving this optimisation problem, e.g. the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). Using linear mixed models on data with small sample sizes (between 100 and 250 samples) the actual algorithmic runtime is approximately on par with the data handling runtime. For more than 250 samples the runtime of the algorithm takes over.

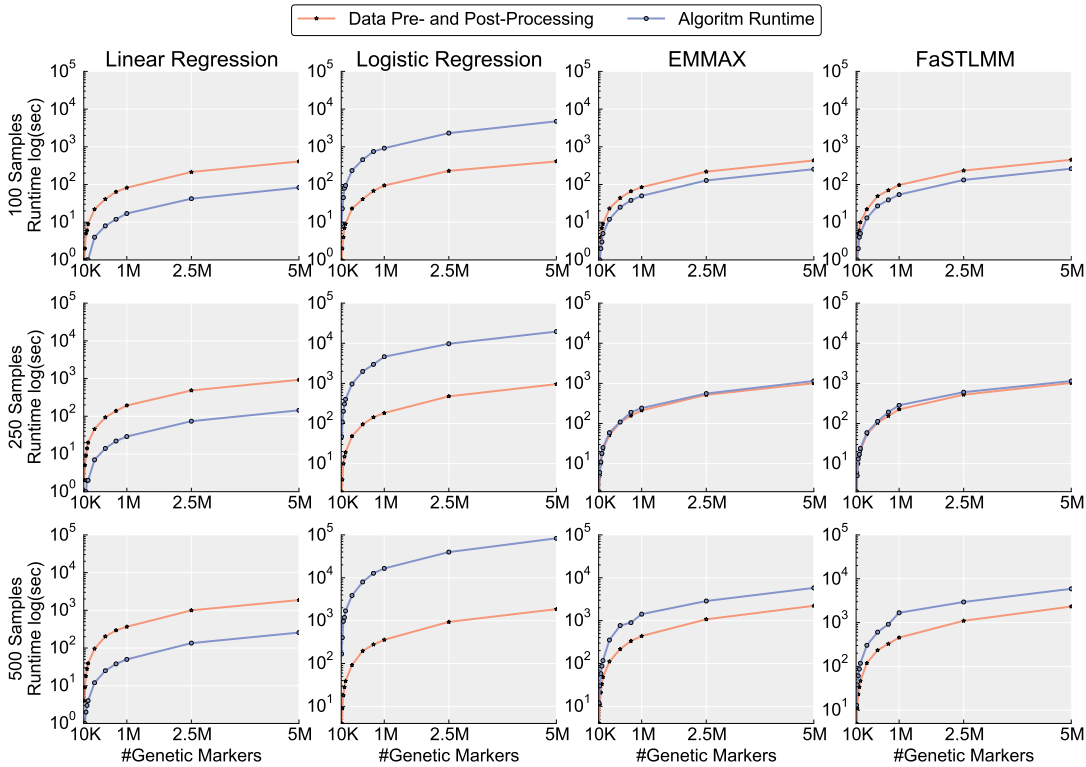


Figure 2.9: Performance analysis of data processing and algorithmic runtime: Performance analysis of four GWAS algorithms from `easyGWASCore` with respect to data handling and actual algorithmic runtime. The number of genetic markers as well as the number of samples were varied for each algorithm.

Next, we compared the performance of the `easyGWASCore` implementations to those from the individual tools PLINK v1.0.7 [Purcell et al., 2007], EMMAX [Kang et al., 2010] and FaSTLMM [Lippert et al., 2011]. For all algorithms and tools the total CPU time in seconds was measured; here, the runtime includes data processing, running the algorithm and writing the result files. When possible we always used the same parameters across all algorithms and tools, e.g. we used the same intervals and number of iterations for the Brent optimisation when comparing to FaSTLMM. All results are illustrated in Figure 2.10. Except for logistic regression all implementations in `easyGWASCore` were at least as efficient than the tools compared too. For FaSTLMM the `easyGWASCore` implementation was on average between 0.5 and 1 magnitude faster than the original one.

2.5 Chapter Summary

In this chapter we reviewed various important concepts, algorithms and method for genome-wide association and meta-studies. We described different regression based methods, the corresponding statistical inference procedures and introduced different concepts for multiple hypothesis testing. Further, we developed an integrated C/C++ framework with Python interfaces, called `easyGWASCore`. The `easyGWASCore` API will serve as a common basis for all algorithms we develop throughout this thesis, as well as for future developments. The `easyGWASCore` framework consists of two main

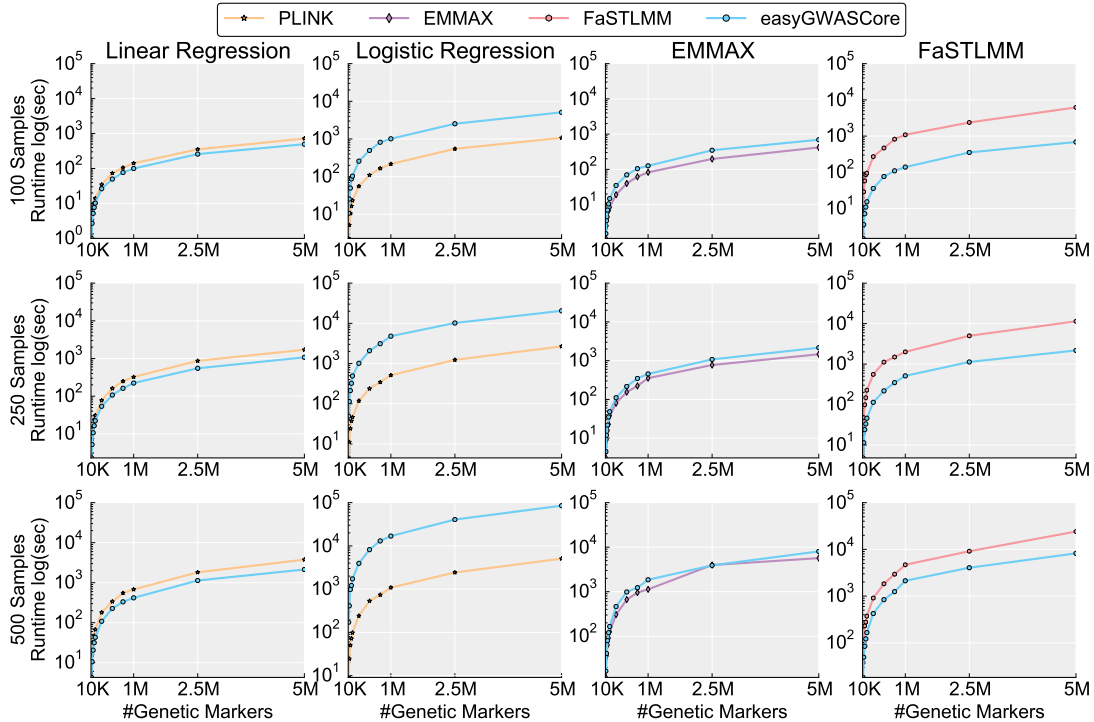


Figure 2.10: Runtime comparison between easyGWASCore and the individual tools: For each tool we measured the total runtime in seconds and compared it to the easyGWASCore implementation (blue). We varied the number of genetic markers as well as the number of samples.

parts, which can be divided into an application programming interface and a stand-alone command line tool. We showed on various examples how the API can be utilised to call different algorithms for GWASs and meta-analyses. In addition, we gave an example of how to exploit the API to develop a novel algorithm for a two-locus association mapping.

The Python command line interface is a command line tool to easily access different algorithms and analysis methods. We demonstrated its abilities by conducting an example GWAS in the model organism *Arabidopsis thaliana*, including visualisation and annotation of results. Finally, we analysed the performance of the easyGWASCore framework by comparing it to state-of-the-art tools. We found that easyGWASCore is at least as efficient as these popular tools.

One of the main advantages of our framework is the common data, visualisation and annotation pipeline. However, the annotation and interpretation of associated variants of GWASs is still a cumbersome task and often not possible without additional extensive biological experiments. In the next chapter we will investigate a class of *in silico* tools that can be used to narrow down the number of potential candidate variants by predicting whether a given missense variant might lead to a damaging or pathogenic effect on the protein. Due to the wealth of such pathogenicity prediction tools we will conduct a comprehensive comparison of ten widely used pathogenicity prediction tools and their ability to generalise to new unseen data. Eventually, we will extend the easyGWASCore annotation pipeline with pathogenicity prediction scores as an additional source of information.

CHAPTER 3

Pathogenicity Prediction Scores as Additional Source for Annotation

The annotation and interpretation of associated variants from GWASs is a cumbersome process and is often even impossible without additional extensive biological experiments. Reliable strategies that allow scientists to prioritise these variants for further investigations are therefore of high practical relevance. Different *in silico* tools, such as `snpEFF` [Cingolani *et al.*, 2012] or `Variant Effect Predictor (VEP)` [McLaren *et al.*, 2010], have been developed in recent years to annotate SNPs based on their genomic position. These tools predict potential effects of a given variant on genes (including different transcripts), proteins or regulatory regions. Variants that cause an amino-acid change might further lead to a harmful and damaging effect on the protein. Hence, several *in silico* tools have been developed that predict whether a given missense variant could lead to a potential pathogenic or damaging effect on the protein, such as `PolyPhen-2 (PP2)` [Adzhubei *et al.*, 2010], `MutationTaster-2 (MT2)` [Schwarz *et al.*, 2014], `MutationAssessor (MASS)` [Reva *et al.*, 2011], `SIFT` [Ng and Henikoff, 2003], `LRT` [Chun and Fay, 2009], `FatHMM weighted (FatHMM-W)` and `unweighted (FatHMM-U)` [Shihab *et al.*, 2013] or `Combined Annotation Dependent Depletion (CADD)` [Kircher *et al.*, 2014]. Sequence conservation scores, such as `phyloP` [Cooper and Shendure, 2011] or `GERP++` [Davydov *et al.*, 2010], are also often used for this task.

In this chapter we extend the `easyGWASCore` framework to also include pathogenicity prediction scores as an additional source for annotation. Due to the wealth of such pathogenicity prediction tools we first investigate the predictive performance of ten popular prediction tools in a comprehensive and systematic evaluation [Grimm *et al.*, 2015]. We demonstrate that a comparative evaluation of these tools is hindered by two types of circularity and encounter that the first type of circularity — type 1 circularity — is due to overlaps between datasets that were used for training and evaluation of these tools. Tools such as `PP2`, `MT2`, `MASS` and `CADD`, which require data to learn the parameters of their prediction model, run the risk of capturing idiosyncratic characteristics of their training data, leading to poor generalisation abilities when applied on new data. To prevent this phenomenon of overfitting [Hastie *et al.*, 2009b] it is imperative that tools are evaluated on data that were not used for the training of these tools

[Vihinen, 2013]. The second type of circularity — type 2 circularity — is closely linked to a statistical property of current variant databases. Often, all variants from the same gene are jointly labeled as being either pathogenic or neutral. As a consequence, classifiers that predict pathogenicity based on known information about specific variants in the same gene will achieve excellent results, while being unable to detect novel risk genes, for which no variants have been annotated before. Consequently, it will not be able to discriminate between pathogenic and neutral variants *within* the same gene. After this comprehensive analysis of *in silico* prediction tools and discussing their potential pitfalls, we demonstrate how to integrate pathogenicity prediction scores into the annotation and visualisation pipeline of the `easyGWASCore` framework.

3.1 A Comprehensive Analysis of Pathogenicity Prediction Tools

In this section we analyse the predictive performance of ten widely used pathogenicity prediction tools, including MT2 [Schwarz *et al.*, 2014], LRT [Chun and Fay, 2009], PP2 [Adzhubei *et al.*, 2010], SIFT [Ng and Henikoff, 2003], MASS [Reva *et al.*, 2011], FatHMM-W and FatHMM-U [Shihab *et al.*, 2013], CADD [Kircher *et al.*, 2014], phyloP [Cooper and Shendure, 2011] and GERP++ [Davydov *et al.*, 2010]. All these tools are commonly applied to predict whether a given missense variant might have a damaging effect on the protein. However, the original purposes these tools were designed for varies. The tools phyloP and GERP++ measure sequence conservation, while others, such as PP2, try to assess the impact of missense variants on protein structure or function. CADD in turn quantifies the overall pathogenic potential of a variant based on diverse types of genomic information. The tool SIFT is both a measure of sequence conservation, as well as a prediction tool whether or not protein function will be affected.

Given this wealth of different prediction tools and the fact that they are used for the same purposes, an important practical question to answer is whether one or several tools systematically outperform all others in terms of predictive performance. To address this question, we assess the predictive performance of these ten tools across five major public databases previously used to test these tools: *HumVar* [Adzhubei *et al.*, 2010], *ExoVar* [Li *et al.*, 2013], *VariBench* [Nair and Vihinen, 2013; Thusberg *et al.*, 2011], *predictSNP* [Bendl *et al.*, 2014] and the latest *SwissVar* (Dec. 2014) database [Mottaz *et al.*, 2010] (Table 3.1).

3.1.1 Experimental Settings

Data Preparation

Since some of these tools either require nucleotide substitutions or amino acid substitutions as input, we used the tool VEP [McLaren *et al.*, 2010] to convert all five benchmark datasets (Table 3.1) between both formats. Note that by contrast, analyses such as that of Thusberg *et al.* [2011] only assess tools that require amino-acid changes as in-

put. We also excluded all variants for which we could not determine an unambiguous

Datasets	Deleterious Variants (D)	Neutral Variants (N)	Total	Ratio (D:Total)	Tools potentially trained on data (fully or partly)	Removed variants overlapping with
<i>HumVar</i>	21,090	19,299	40,389	0.52	MT2, MASS, PP2, FatHMM-W	CADD training data
<i>ExoVar</i>	5,156	3,694	8,850	0.58	MT2, MASS, PP2, FatHMM-W	CADD training data
<i>VariBenchSelected</i>	4,309	5,957	10,266	0.42	MT2	CADD training data, <i>HumVar</i> , <i>ExoVar</i>
<i>predictSNPSelected</i>	10,000	6,098	16,098	0.62	MT2	CADD training data, <i>HumVar</i> , <i>ExoVar</i> , <i>VariBench</i>
<i>SwissVarSelected</i>	4,526	8,203	12,729	0.36	MT2	CADD training data, <i>HumVar</i> , <i>ExoVar</i> , <i>VariBench</i> , <i>predictSNP</i>

Table 3.1: Overview of all benchmark datasets: These preprocessed and filtered datasets are used to evaluate the performance of different prediction tools.

nucleotide or amino acid change. In addition, we systematically excluded all variants overlapping with the CADD [Kircher *et al.*, 2014] training data from all other data sets as the intersection of the training data from the tool CADD and that of all other data sets is small (fewer than a hundred variants). The *VariBench* dataset (benchmark database for variations) was created to address the problem of type 1 circularity [Nair and Vihinen, 2013; Thusberg *et al.*, 2011], that is the overlap between datasets that were used for training and evaluation of the models. However, while the pathogenic variants of this dataset were new, its neutral variants may have been present in the training data of other tools. *VariBench* has an overlap of approximately 50% with both *HumVar* and *ExoVar* (Figure 3.1). Importantly, these two datasets overlap with at least one of the training sets used to train the individual tools FatHMM-W, MT2, MASS and PP2.

We kept the non-overlapping variants of *VariBench* with *HumVar* and *ExoVar* to build an independent evaluation dataset, which we called *VariBenchSelected*. From the *predictSNP* benchmark dataset, we systematically excluded all variants that overlap with *HumVar*, *ExoVar* and *VariBench* and called the resulting dataset *predictSNPSelected*. In addition, we created a fifth dataset, *SwissVarSelected*. Here, we excluded from the latest *SwissVar* database (Dec. 2014) all variants overlapping with the other four datasets — *HumVar*, *ExoVar*, *VariBench* and *predictSNP*. Thus, *SwissVarSelected* should be the dataset containing the newest variants across all datasets. With one possible exception, none of the prediction tools or conservation scores we investigated in this manuscript were trained on *VariBenchSelected*, *predictSNPSelected* or *SwissVarSelected*. The exception is that some variants in the *Selected* datasets may overlap partially with variants used to train MutationTaster-2 (MT2) [Schwarz *et al.*, 2014] because MT2 was trained on private data (a large collection of disease variants from HGMD Professional [Stenson *et al.*, 2014]). Hence, the *Selected* datasets can be considered to be truly independent evaluation datasets, which are free of type 1 circularity.

Eventually, we obtained, for any given variant, scores and prediction labels for each tool directly from their respective web-servers or standalone tools. Since the pathogenicity score of a missense variant may depend on which transcript of the corresponding gene is considered, we standardised our analyses by examining the same transcript across all tools. Because of that, we chose the canonical transcript [Hubbard *et al.*, 2009].

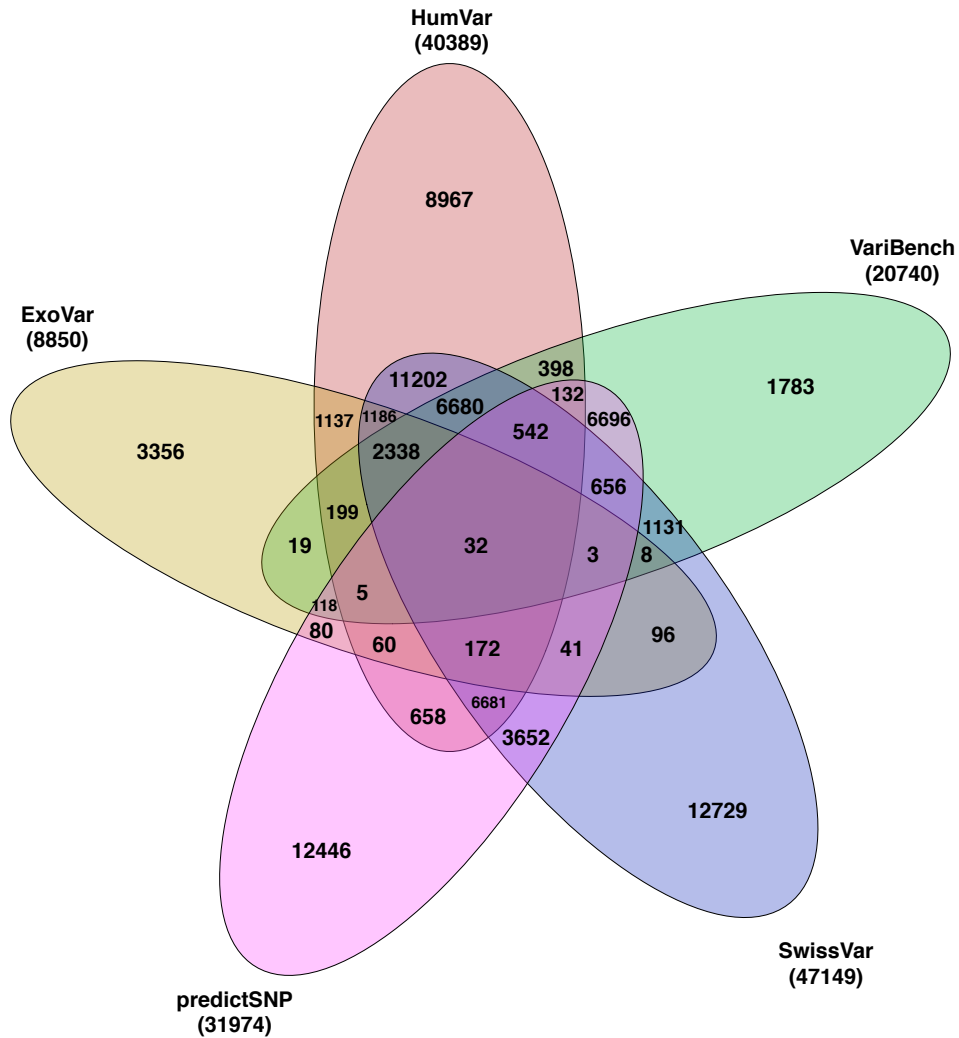


Figure 3.1: Venn-diagram showing the overlap between all five benchmark datasets: *VariBenchSelected* (10266 variants) is the part of *VariBench* not overlapping with *HumVar* nor *ExoVar*. *predictSNPSelected* (16098 variants) is the part of *predictSNP* not overlapping with *HumVar*, *ExoVar* nor *VariBench*. *SwissVarSelected* (12729 variants) is the part of *SwissVar* that does not overlap with *HumVar*, *ExoVar*, *VariBench*, nor *predictSNP*.

Performance Evaluation

To evaluate the performance of all the tools, we computed the area (AUC) under the receiver operating characteristic curve (ROC) (see Appendix B) using the predicted output of the variants and the true labels of the variants. Note that no cross-validation is needed here because the tools are already pre-trained. Thus, we are merely evaluating the generalisation abilities of all trained tools.

3.1.2 Results

Evaluation of Ten Pathogenicity Prediction Tools

First, we evaluated the performance of ten pathogenicity prediction tools and reported AUC values per tool and dataset in Figure 3.2. Hatched bars in Figure 3.2 indicate that the evaluation data were used in part or entirely to train the corresponding tool. Con-

sequently, these results may suffer from overfitting. On the two benchmarks *HumVar*

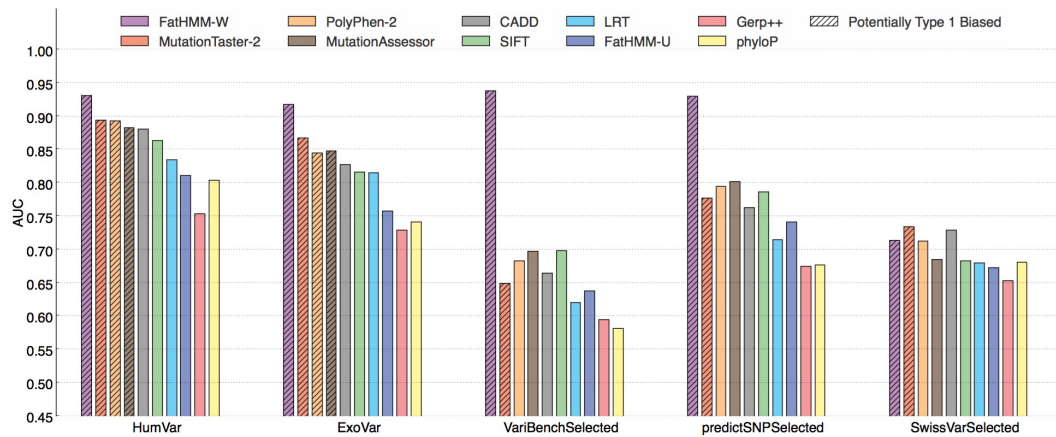


Figure 3.2: Predictive performance of 10 popular pathogenicity prediction tools over five datasets: Evaluation of the ten different pathogenicity prediction tools (by AUC) over five datasets. The hatched bars indicate potentially biased results, due to the overlap (or possible overlap) between the evaluation data and the data used (by tool developers) for training the prediction tool.

and *ExoVar*, the four best performing methods were fully or partly trained on these datasets (Figure 3.2). While MT2, PP2 and MASS outperformed CADD and SIFT on benchmarks that include some of their training data (*HumVar*, *ExoVar*), this was not the case on the independent *VariBenchSelected* and *predictSNPSelected* datasets. A potential explanation is that type 1 circularity — that is the overlap between training and evaluation sets — might lead to overly optimistic results on the first two datasets. We further observed that, across the first four datasets, *FatHMM-W* outperformed all other tools (Figure 3.2). However, *FatHMM-W* showed a severe drop in performance on the *SwissVarSelected* dataset. Finally, we observed across all datasets that trained predictors generally outperform untrained conservation scores. The superiority of *FatHMM-W*’s [Shihab et al., 2013] predictions on *VariBenchSelected* and *predictSNPSelected* and the severe drop in performance on *SwissVarSelected* made us investigate its underlying model to find the reason for its superior performance on all but one dataset.

Investigation of the Good Performance of *FatHMM-W*

In its unweighted version, *FatHMM-U* scores each variant by the log odds ratio of wild-type (P_w) to mutation amino acid (P_m), where the probabilities of observing each version of the amino acid are determined by an Hidden Markov Model based multiple-sequence alignment against UniRef90 sequences [Suzek et al., 2007]. The *FatHMM-U* score is then obtained as follows:

$$\text{FatHMM-U} = \ln \left(\frac{P_m(1 - P_m)^{-1}}{P_w(1 - P_w)^{-1}} \right) = \ln \frac{P_m(1 - P_w)}{P_w(1 - P_m)}. \quad (3.1)$$

Essentially, *FatHMM-U* assumes that the more conserved the position at which the mutation occurred, the more likely it is to be pathogenic.

The weighted version (*FatHMM-W*) also takes into account how tolerant to mutations

the sequence is. The tolerance to mutation of a sequence is evaluated using its relative frequency of known neutral (W_n) versus known pathogenic (W_d) variants in the relevant protein family, defined through SUPERFAMILY [Gough *et al.*, 2001] or Pfam [Sonnhammer *et al.*, 1997]. For this purpose, the **FatHMM-U** score from Equation 3.1 is weighted by the relative frequency of benign variants found in the UniProt database [Magrane *et al.*, 2011] and pathogenic variants from the non-public HGMD Professional database [Stenson *et al.*, 2014]. The updated score for the weighted version of **FatHMM-W** is defined as:

$$\text{FatHMM-W} = \ln \frac{(1 - P_w)(W_n + 1)}{(1 - P_m)(W_n + 1)}. \quad (3.2)$$

To further evaluate the role of this weighting scheme in the performance of **FatHMM-W**, we compared the original **FatHMM-W** to a regularised logistic regression [Lee *et al.*, 2006] over the weighting features ($\ln(W_n)$ and $\ln(W_d)$) in a 10-fold cross-validation on all *Selected* datasets. The use of these features alone was sufficient to achieve approximately the same predictive performance as **FatHMM-W** (Figure 3.3).

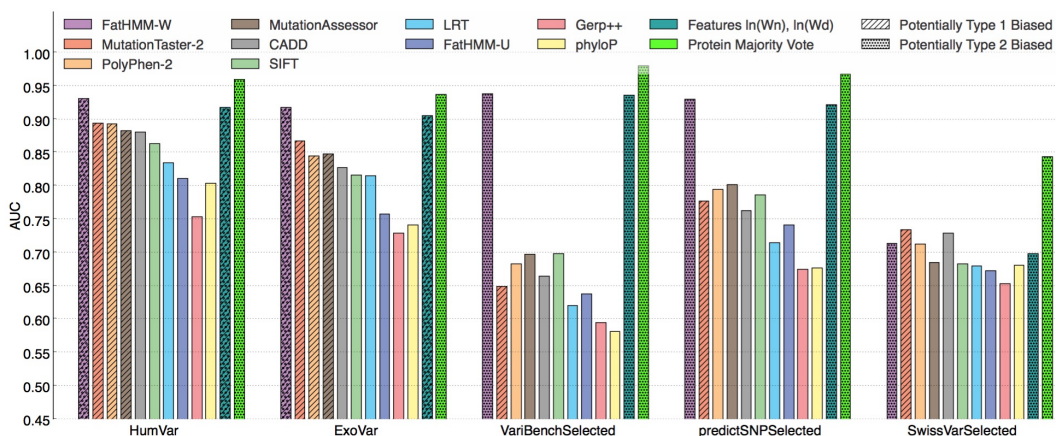


Figure 3.3: Evaluation of type 2 circularity: Evaluation of the ten different pathogenicity prediction tools (by AUC) over five datasets. The hatched bars indicate potentially biased results, due to the overlap (or possible overlap) between the evaluation data and the data used (by tool developers) for training the prediction tool. The dotted bars indicate that the tool is biased due to type 2 circularity.

Given that the ratio of neutral and pathogenic variants in the same protein family is the key feature used by **FatHMM-W**, we further analysed how an even simpler statistic — the fraction of pathogenic variants in the same protein — performs as a predictor. We refer to this predictor as a Protein Majority Vote (**MV**). For this purpose, we split each of the five evaluation datasets into ten subsets, and for each of the subsets, used the union of the nine other subsets as training data. Within that framework, we scored a variant by the pathogenic-to-neutral ratio, in the training data, of the protein that variant belongs to. If the protein did not appear in the training data, we assigned a score of 0.5. **MV** systematically outperformed **FatHMM-W** as shown in Figure 3.3. The pathogenicity of neighbouring variants within the same protein was therefore the best predictor of pathogenicity across these datasets. This strategy, while statistically effective on the currently existing databases, is not appropriate. Indeed it assigns the

same label to all variants in the same protein, based on information likely obtained at the protein-level (i.e. that it is associated with a disease), and cannot distinguish between pathogenic and neutral variants within the same protein.

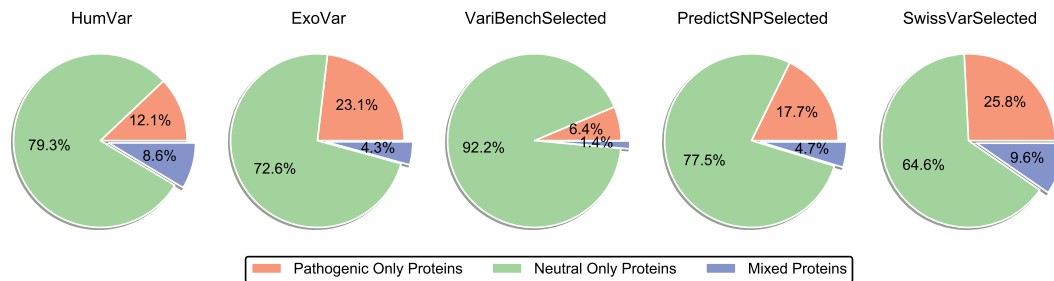
To better understand the outstanding performance of *FatHMM-W* and the *MV*, we examined the relative frequency of pathogenic variants across proteins in all our datasets. In the independent evaluation dataset *VariBenchSelected*, we found that more than 98% of all proteins (4,425 out of 4,490, Table 3.2) contain variants from a single

Datasets	“Pure” Pathogenic Proteins	Pathogenic Variants in “pure” Proteins	“Pure” Neutral Proteins	Neutral Variants in “pure” Proteins	Mixed Proteins	Variants in Mixed Proteins	Total Number of Proteins
<i>Hum Var</i>	1,277	10,484	8,400	17,140	911	12,765	10,588
<i>Exo Var</i>	891	4,336	2,794	3,478	165	1,036	3,850
<i>VariBenchSelected</i>	286	3,865	4,139	5,869	65	532	4,490
<i>predictSNPSelected</i>	855	7,090	3,738	5,649	228	3,359	4,821
<i>Swiss VarSelected</i>	1,444	2,749	3,614	6,568	549	3,412	5,598

Table 3.2: Protein categories and variants per category: Overview about the total number of proteins per dataset and the composition of these datasets.

class, i.e. either “pathogenic” or “neutral” (Figure 3.4a). For the remainder of this thesis, we shall refer to proteins with only one class of variant as “pure” proteins (divided in “pathogenic-only” proteins and “neutral-only” proteins). The existence of such “pure” proteins — while theoretically possible — should not be interpreted as a biological phenomenon. Rather, these designations are based on current knowledge, and are at least partially an artefact of how these particular datasets are populated. For the other datasets the fraction of mixed proteins was slightly larger (Figure 3.4a) but still relatively small compared to the other datasets.

(a) Protein perspective



(b) Variant perspective

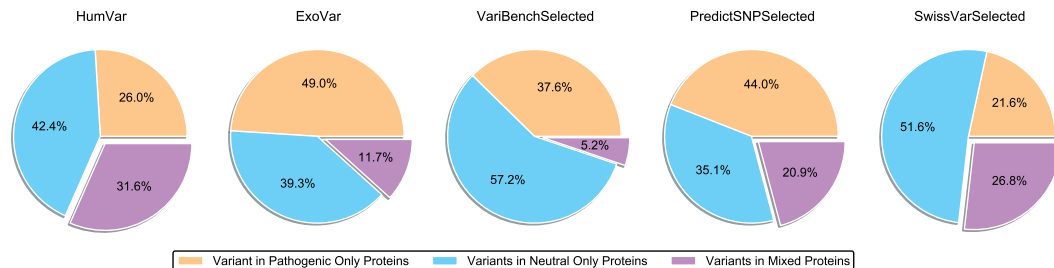


Figure 3.4: Dataset compositions: (a) Protein perspective: proportion of proteins containing only neutral variants (“neutral-only”), only pathogenic variants (“pathogenic-only”), and both types of variants (“mixed”). (b) Variant perspective: proportions, of variants in each of the three categories of proteins.

Nearly all (94.8%) variants in *VariBenchSelected* were located in pure proteins with

57.2% in neutral-only proteins and 37.6% in pathogenic-only proteins (Figure 3.4b). On such a dataset, excellent accuracies can be achieved by predicting the status of a variant based on the other variants in the same protein. This is the phenomenon we referred to as type 2 circularity. The remaining 5.2% of *VariBenchSelected* variants were located in “mixed” proteins (Figure 3.4b and Figure 3.5), which contained both pathogenic and neutral variants (pathogenic-to-neutral ratio in the open interval $]0.0, 1.0[$ in Figure 3.5). While the MV approach will necessarily misclassify some of these variants, it will still perform well on proteins containing primarily neutral or primarily pathogenic variants, and overall, only 0.7% of all variants in *VariBenchSelected* were in proteins containing an almost balanced ratio of pathogenic and neutral variants (pathogenic-to-neutral ratio in the interval $[0.4, 0.6]$ in Figure 3.5). Similar dataset compositions could be observed in the other three datasets *HumVar*, *ExoVar* and *predictSNPSelected* (Figure 3.4 and Figure 3.5). A striking property of *SwissVarSelected* was its much larger fraction of proteins with almost balanced pathogenic-to-neutral ratio: 6.5% of all variants (832 out of 12729) could be found in the most balanced category of mixed proteins $[0.4, 0.6]$ (Figure 3.5), compared to an average of 1.5% in the other four datasets.

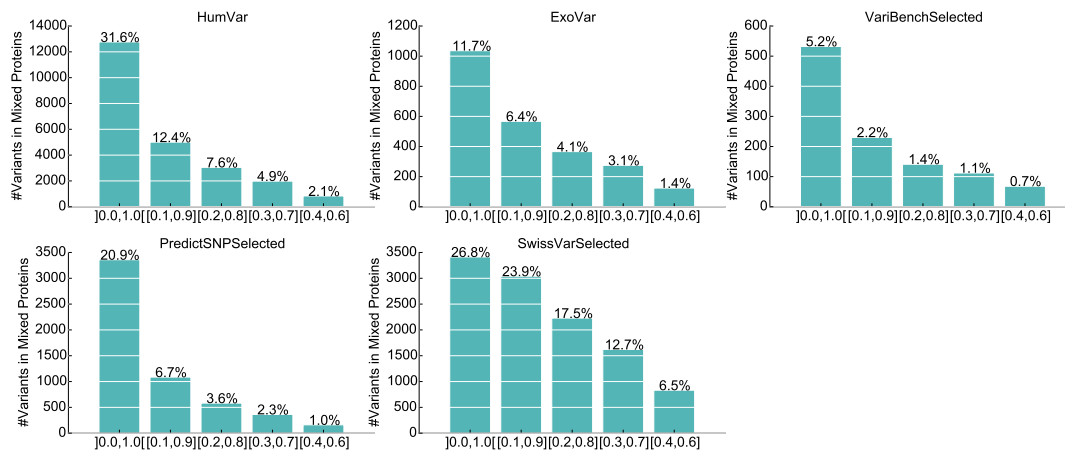


Figure 3.5: Fractions of variants for each dataset: Fractions of variants containing various ratios of pathogenic-to-neutral variants, binned into increasingly narrow bins, approaching balanced proteins. The open interval $]0.0, 1.0[$ contains all mixed proteins (as in Figure 3.4b).

To further understand *FatHMM-W*’s performance, we evaluated it separately on the mixed proteins. As shown in Figure 3.6, *FatHMM-W* performed well on pure proteins but lost much of its predictive power on the mixed proteins, as it is misled by its weighting scheme. On almost-balanced proteins, *FatHMM-W* was therefore outperformed by all other tools but *phyloP*. This may also be the first reason why *FatHMM-W* performed worse on *SwissVarSelected* than on all other datasets: *SwissVarSelected* contained many more variants in the most mixed categories, as shown in Figure 3.5. We observed that *PP-2* outperformed all other tools in the mixed categories for the datasets *predictSNPSelected* and *SwissVarSelected*. For the *VariBenchSelected* dataset no clear winner could be determined.

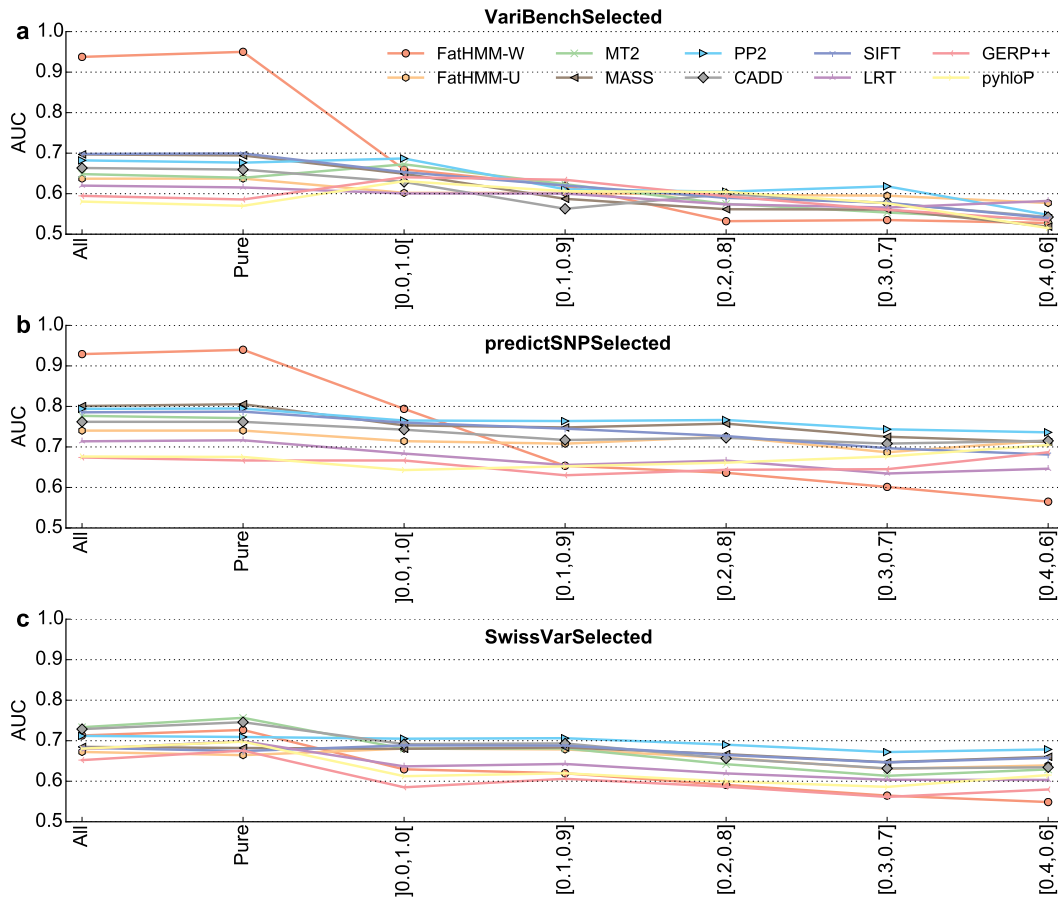


Figure 3.6: Performance of ten pathogenicity prediction tools according to protein pathogenic-to-neutral variant ratio: Evaluation of tool performance on subsets of *VariBenchSelected*, *predictSNPSelected* and *SwissVarSelected*, defined according to the relative proportions of pathogenic and neutral variants in the proteins they contain. “Pure” indicates variants belonging to proteins containing only one class of variant. [x, y] indicate variants belonging to mixed proteins, containing a ratio of pathogenic-to-neutral variants between x and y. [0,0, 1,0[therefore indicates all mixed proteins (the ratios of 0.0 and 1.0 being excluded by the reversed brackets). While *FatHMM-W* performs well or excellently on variants belonging to pure proteins (*VariBenchSelected* and *predictSNPSelected*), it performs poorly on those belonging to mixed proteins.

The second reason for the drop in performance was the presence of “new” proteins in *SwissVarSelected* that are unknown to the *FatHMM-W* weighting database. To show this, we used the *HumVar* and *ExoVar* datasets as a proxy for the training data among all our tools (*FatHMM*’s training data is not fully publicly available). We observed that $\sim 91\%$ of all pathogenic and $\sim 68\%$ of all neutral variants in *VariBenchSelected* were located in proteins that also occurred in *HumVar/ExoVar* (Figure 3.7). As *FatHMM-W* makes use of information from protein families, we computed pair-wise BLASTP [Camacho et al., 2009] alignments between all proteins in our *Selected* datasets and proteins in *HumVar/ExoVar*. Approximately 99% of all pathogenic variants in *VariBenchSelected* were located in proteins from *HumVar/ExoVar* or proteins with more than 70% sequence similarity to a protein in *HumVar/ExoVar*. Similar statistics could be observed for *predictSNPSelected* (Figure 3.7). However, for *SwissVarSelected* we observed that only $\sim 61\%$ of all pathogenic and $\sim 56\%$ of all neutral variants belong to proteins from

HumVar/ExoVar (Figure 3.7). Approximately 78% of all pathogenic and 77% of all neutral variants in *SwissVarSelected* were located in proteins from *HumVar/ExoVar* or in proteins with high sequence similarity (70% sequence similarity) to a protein from *HumVar/ExoVar* (Figure 3.7). Hence a significant proportion of *SwissVarSelected* variants could not be found in proteins from the proxy training dataset or proteins with high sequence similarity. All of these findings led to the conclusion that **FatHMM-W**'s good performance on *VariBenchSelected* and *predictSNPSelected* is largely due to type 2 circularity.

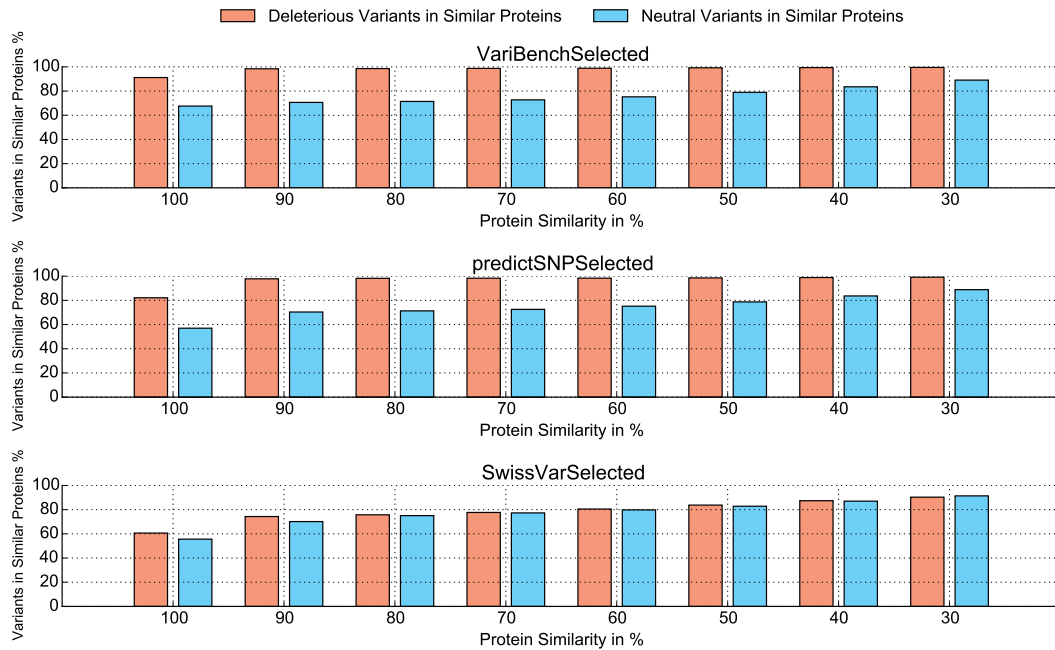


Figure 3.7: Variants from the *selected* datasets that are in identical or similar proteins in the proxy training datasets *HumVar/ExoVar*: Percentage of pathogenic and neutral variants that can be found in identical or similar proteins in *HumVar/ExoVar*. The x-axis shows different similarities between the proteins in the Selected dataset and the proteins in *HumVar/ExoVar*. The y-axis is the percentage of variants that can be found in identical or similar proteins.

Evaluation of Two Meta-Predictors

An additional set of tools we have not evaluated yet are so called *meta-predictors*, such as *Condel* [González-Pérez and López-Bigas, 2011] or *Logit* [Li et al., 2013]. These tools combine the scores of various pathogenicity prediction tools to boost their overall discriminative power. These tools are based on the expectation that individual predictors have complementary strengths because they rely on diverse types of information, such as sequence conservation or modifications at the protein level. However, the problems created by these different types of circularity could be exacerbated when combining several tools. One problem of these combined predictors could be that parts of their individual training data overlaps with those of one or more tools. In that case, tools that have been fitted to the data already will appear to perform better and may receive artificially inflated weights. The second problem could be that the data that is used

to assess the meta-predictor overlaps partly with those used to train the tools. Here, the tools themselves are biased toward performing well on the evaluation data, which can make their combination appear to perform better than it actually does.

In this subsection we evaluated two meta-predictors, `Condel` and `Logit`. Based on our previous findings about two types of circularity, we were interested in their performance when evaluated on datasets that avoid type 1 circularity, as well as in the effect of including the type 2 circularity-biased tool `FatHMM-W`. For this purpose, we compared the combination of the tools `PP2`, `MASS` and `SIFT` using the meta-predictor `Condel` and `Logit` with the combination of these three tools plus `FatHMM-W`. We referred to the meta-predictors that include `FatHMM-W` as `Condel+` and `Logit+`, respectively. Again, we evaluated the performance of these predictors on all five datasets including the *Selected* datasets, which are free of type 1 circularity (Figure 3.8). As already reported in

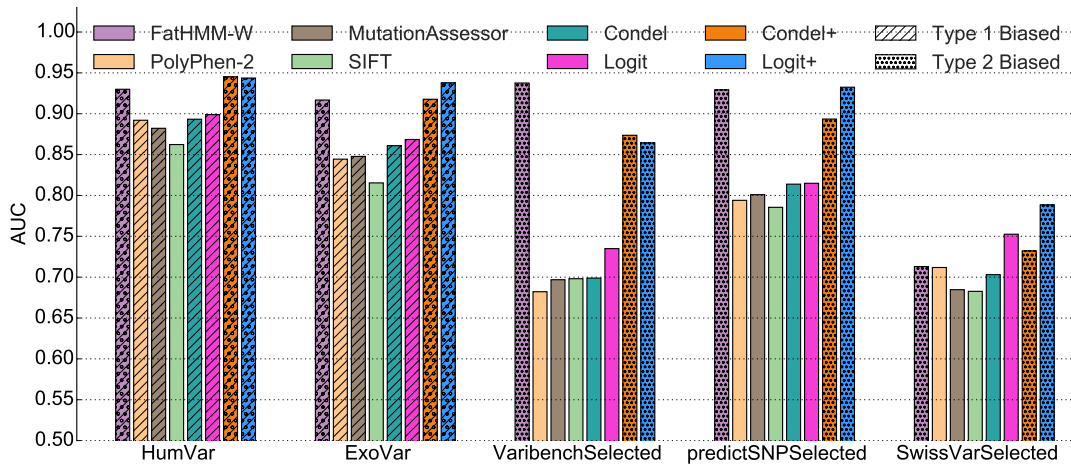


Figure 3.8: Comparison of the performance of two meta-predictors (`Logit` and `Condel`) and their component tools, across five datasets: Bar heights reflect AUC for each tool and tool combination. `Logit` and `Condel` are meta-predictors combining `MASS`, `PP2`, and `SIFT`. The “+” versions of `Logit` and `Condel` also include `FatHMM-W`. While effective in prediction, `FATHMM-W` (alone and in the `Logit+` and `Condel+` meta-predictors) is optimistically biased due to type 2 circularity.

Li et al. [2013], `Logit` outperformed all individual tools and `Condel` in terms of AUC. The performance of `Condel` on *VariBenchSelected* was on par with the performance of `SIFT` with an AUC=0.70. This showed that `Condel` is not necessarily superior to its individual tools on an unbiased dataset. In addition, we observed across all datasets that including `FatHMM-W` to either meta-predictor (`Condel+` and `Logit+`) led to a performance boost. However, these tools may be optimistically biased by type 2 circularity, given the inclusion of `FatHMM-W`. To show this, we again evaluated the performance of these tools and meta-predictors on the pure and mixed proteins on the *Selected* datasets (Figure 3.9). While `Logit` performed well on the pure proteins, `Condel` performed at least as well as `Logit` on variants in mixed proteins. Including `FatHMM-W`, however, led to a significant drop in performance for both `Logit+` and `Condel+` on all datasets but *SwissVarSelected*.

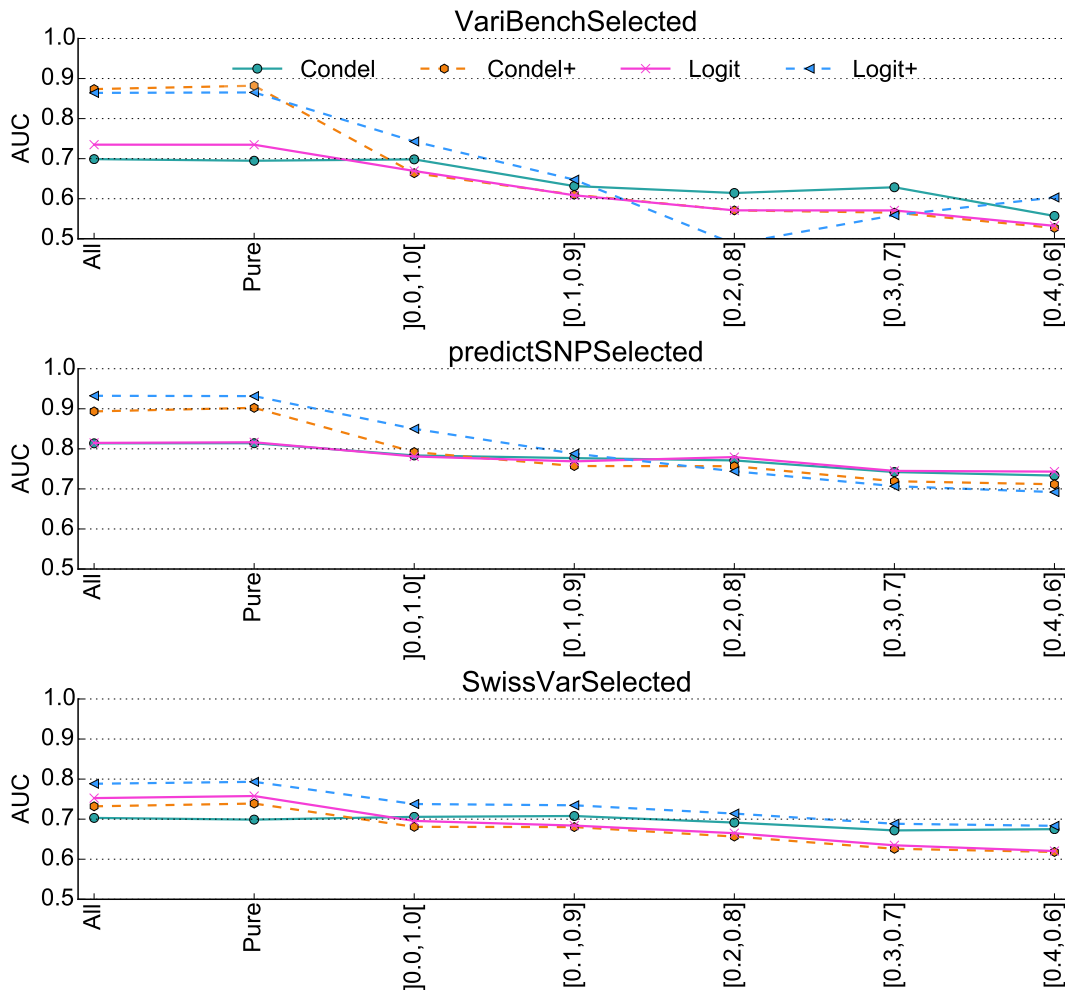


Figure 3.9: Performance according to protein pathogenic-to-neutral variant ratio: Evaluation of Condel, Condel+, Logit and Logit+ on subsets of *VariBenchSelected*, *predictSNPSelected* and *SwissVarSelected* defined according to the relative proportions of pathogenic and neutral variants in the proteins they contain.

3.1.3 Guidelines to Avoid Different Types of Circularity

We demonstrated that a fair evaluation of these different pathogenicity prediction tools is hindered by two types of circularity and that ignoring these effects could lead to overly optimistic assessments of tool performances. One severe consequence of this phenomenon is that it may hinder the discovery of novel disease risk genes, as these tools are widely used to choose variants for further functional investigation. Thus, it is important to be aware of these different types of circularity and to know how to avoid them in the future.

To avoid type 1 circularity, prediction tools should only be compared on benchmark datasets that do not overlap with any of the datasets used to train the tool. Although it is a well studied [Hastie *et al.*, 2009b; Vihinen, 2012, 2013] and trivial sounding problem, it is still a frequently made mistake. Also avoiding type 1 circularity is made more difficult by the fact that developers of such tools do not always share their training data, e.g. because they used private non-public datasets. Thus, we advocate to also publish the training data of such prediction tools.

A more rigorous strategy would be to retrain all available predictors on the same data,

in order to truly evaluate the predictor and not the quality of their training datasets. However, this is only possible if developers share their code, so that one can retrain their tool on other variants, or if the raw variant descriptors (variant features) — from which the tools derive their predictions — are made available. We investigated all ten tools and only for PP2 [Adzhubei *et al.*, 2010] was it straightforward to obtain these features.

To avoid type 2 circularity, it is imperative that future studies report prediction accuracy as function of the pathogenic-to-neutral ratio, as illustrated in Figure 3.6 and 3.9. An even better strategy would be to stratify training and test datasets such that variants from the same protein only occur in either the training or the test dataset. This way we completely remove the possibility to classify variants from the same protein, as suggested by Adzhubei *et al.* [2010]. Alternatively, one could develop different predictors for different pathogenic-to-neutral ratios in proteins.

Eventually, we have not mentioned another potential source of circularity. Let us consider the case for which variants are annotated by existing pathogenicity prediction tools and subsequently entered into a publicly available variant database. Here, it might be that tools that appear to perform well on “new” data, are in fact only recovering labels that they have given themselves. For that reason it is necessary to document the source of evidence that was used to assign a label to a variant when it was entered into a database.

3.2 Adding Pathogenicity Prediction Scores to easyGWAScore

In the last sections we conducted a comprehensive evaluation of various pathogenicity prediction tools and discussed different sources of circularity and its pitfalls. Nevertheless, these tools are still a great resource to narrow down certain variants for further biological investigations when knowing and avoiding these types of circularity as much as possible. That is why we extended the visualisation and annotation pipeline of **easyGWAScore** by highlighting missense variants, as well as their predicted pathogenicity status in the linkage disequilibrium plots. In these plots missense variants are illustrated with triangles, where an upper triangle (\triangle) represents a missense variants predicted to be pathogenic and a lower triangle (∇) a missense variant predicted to be benign. The colour of the triangles indicate the degree of LD to the focal (magenta) SNP. Using the command line argument `--pathogenicity_scores` one can pass a file containing pathogenicity prediction scores and labels for different variants. Listing 3.1 specifies the pathogenicity data input format. The data file has to be tab-separated. In addition, the file must contain the variant identifiers, matching the identifiers from the original PLINK files, as well as the predicted pathogenicity score, the predicted label (either BENIGN or DELETERIOUS) and the type of the variant (NONCODING, SYNONYMOUS or NONSYNONYMOUS). Optionally, the file can contain additional information, such as the amino acid change or the transcript id.

Listing 3.1: File format of the predicted pathogenicity scores file

1	Columns 1–6:	
2	-----	
3		
4	1. variant_identifier	This identifier must match the variant identifiers
5		from the PLINK files
6	2. chromosome	The chromosome on which the variant is located
7	3. position	The genomic position in bp
8	4. score	Predicted Pathogenicity Score
9	5. prediction	Pathogenicity Prediction, either BENIGN or
10		DELETERIOUS
11	6. variant_type	Type of the variant, e.g. NONCODING, SYNONYMOUS,
12		NONSYNONYMOUS
13		
14	Optional Columns 6–14:	
15	-----	
16		
17	6. reference_allele	The reference allele of the variant
18	7. alternative_allele	The alternative allele of the variant
19	8. transcript_id	Transcript identifier
20	9. gene_id	Gene identifier
21	10. gene_name	Gene name
22	11. region	Region of the variant, e.g. CDS, UTR_3, UTR_5
23	12. reference_amino	The reference amino acid
24	13. alternative_amino	The alternative amino acid
25	14. position_amino	The position of the amino acid change

To illustrate this annotation feature we conducted an example GWAS using the phenotype *YEL* in *Arabidopsis thaliana* [Atwell *et al.*, 2010]. First, we executed the FaSTLMM [Lippert *et al.*, 2011] algorithm using the easyGWASCore command line tool. No minor allele frequency filtering was applied for this example. Most pathogenicity prediction tools were design to predict the pathogenicity status for human variants. To retrieve pathogenicity prediction scores for *Arabidopsis thaliana* the tool SIFT4G¹ [Vaser *et al.*, 2015] can be used. SIFT4G is a more efficient version of the original SIFT [Ng and Henikoff, 2003] implementation by exploiting Graphical Processing Units (GPUs). Therefore, several databases for different species could be precomputed, including *Arabidopsis thaliana*. We then applied the SIFT4G annotation tool² to determine which SNPs in our dataset were missense variants and whether these variants had a predicted pathogenic effect. Next, we converted the SIFT4G output into the input format specified in Listing 3.1 and applied the easyGWASCore plotting commands to generate LD plots for all significantly associated hits by executing the following command line argument:

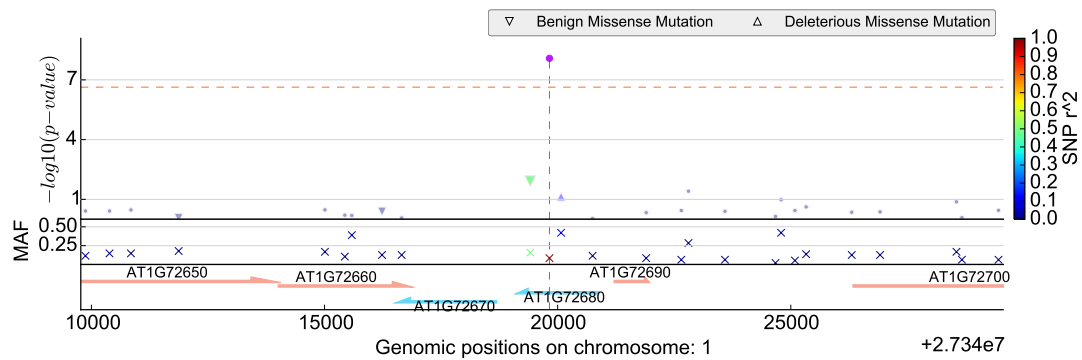
```
$ python python/easygwascore.py plot --ldplot --hfile example/results/
--hdata example/data/YEL.hdf5
--sql_gene example/data/TAIR10.sqlite
--out example/plots/
--ifformat pdf
--pathogenicity_scores example/data/pathogenicity_scores.tab
```

In Figure 3.10 we show three selected LD plots of this GWAS. In Figure 3.10 (a) the focal SNP is in LD to two missense variants. One of those missense variants is predicted to have a pathogenic effect. In Figure 3.10 (b) the focal SNP is a missense variant and is in close LD to other missense variants. In the last Figure 3.10 (c) the focal SNP is

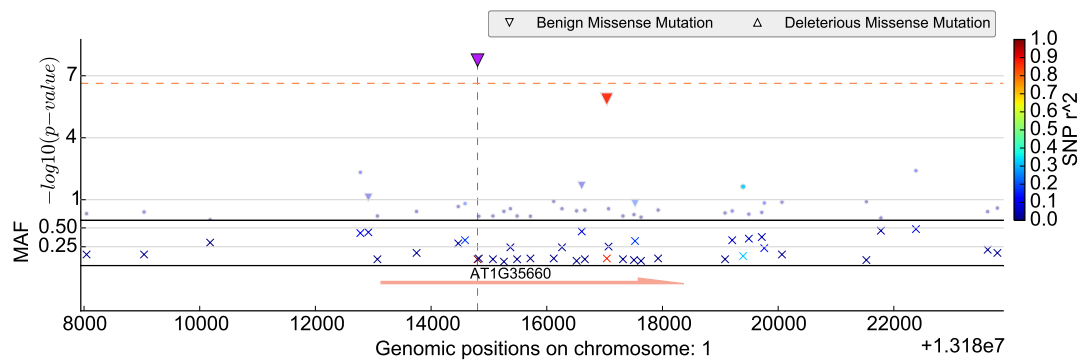
¹<http://sift-db.bii.a-star.edu.sg/AboutSIFT4G.html>

²http://sift-db.bii.a-star.edu.sg/SIFT4G_Annotator_v2.2.jar

- (a) Focal SNP is in close LD to two missense variants, where one of them is predicted to be pathogenic.



- (b) Focal SNP is a missense variant in close LD to other missense variants.



- (c) Focal SNP is a missense variant and predicted to be pathogenic

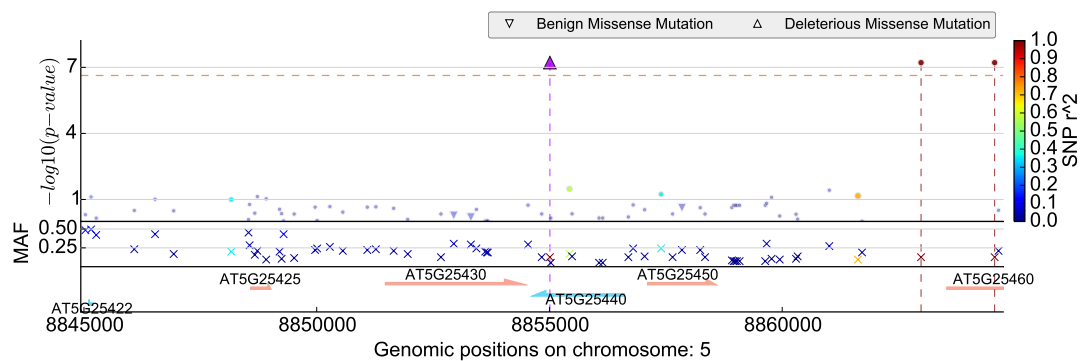


Figure 3.10: Linkage disequilibrium plots including pathogenicity predictions: LD plots for a selected number of significantly associated SNPs. LD plots are enriched with pathogenicity predictions. Upper triangles indicate missense variants predicted to be pathogenic, whereas lower triangles represent benign missense variants.

a missense variant with a predicted pathogenic effect.

3.3 Chapter Summary

In this chapter we conducted a comprehensive analysis of ten widely used pathogenicity prediction tools and two meta-predictors. We investigated whether there are systematic differences in the quality of their predictions when evaluated on five benchmark datasets. We found that the existence of two types of circularity hinder the evaluation of these tools and demonstrated that ignoring these effects could lead to overly optimistic assessments of tool performances. A severe consequence could be that it may hinder the discovery of novel disease risk genes, as these tools are widely used to choose variants for further functional investigation. In this chapter we provided several guidelines on how to avoid and discover these types of circularity. In addition, we proposed a new evaluation strategy to measure the performance of such tools in a fair and competitive way.

Eventually, we extended the **easyGWASCore** framework to also integrate pathogenicity predictions. For this purpose, we modified the linkage disequilibrium plots to also visualise missense variants and their predicted pathogenicity status. We demonstrated this new annotation functionality by performing an example GWAS using *Arabidopsis thaliana* and a flowering time related phenotype.

Although the **easyGWASCore** framework offers a common API and command line interface to conduct different GWASs, it still requires basic unix and command line knowledge. Also, the framework is missing an interface to easily access public GWAS data, as well as straightforward methods to share data and results with collaborators. In addition, the **easyGWASCore** framework offers different visualisation and annotation options, however they are all static. In the next chapter we will introduce a cloud service and web-application for GWASs and meta-analyses, called **easyGWAS**. This service will serve the community at large through easy data access, validation, production, reproduction and dynamic visualisations of GWASs, as well as with community features to share and publish data in a straightforward way.

CHAPTER 4

A Cloud Service for Genome-Wide Association Studies

In the latter chapters we introduced the `easyGWASCore` framework and extended the framework with an easy-to-use annotation pipeline. Furthermore, we diminished the fragmentation of different tools by creating a common data handling and processing pipeline. However, the user still requires either (i) a unix-based platform (e.g. Linux or Mac machine) and basic knowledge on how to use a Unix terminal in order to successfully use the framework or (ii) basic programming skills to utilise the `easyGWASCore` API. Also, plenty of independent steps are required to perform a number of GWASs, as well as a high degree of organisation and management, even when using `easyGWASCore` or PLINK [Purcell *et al.*, 2007]. Irrespective of the recent advances to speed up association mapping algorithms (e.g. [Lippert *et al.*, 2011, 2013; Rakitsch *et al.*, 2013a,b; Segura *et al.*, 2012]), performing large studies with several phenotypes, millions of SNPs and hundreds of samples might still be computationally demanding for a single desktop machine in terms of CPU power, availability of memory and storage. This is aggravated by the need to manage and store a plethora of different input and output files, which might become a non-trivial task for large GWAS projects. Especially, sharing data and result files from such projects is overly complicated if collaborators are in geographically different labs and locations. Eventually, it is imperative that results from GWASs can be visualised and annotated in a straightforward and dynamic way. To address some of these problems, web-applications have been developed at an attempt to simplify the process of performing GWASs, especially on publicly available data, such as *Arabidopsis thaliana* [Childs *et al.*, 2012; Seren *et al.*, 2012], *Drosophila melanogaster* [Mackay *et al.*, 2012] or mouse [Kirby *et al.*, 2010]. However, all these web-applications laid their focus on specific species with a fixed number of samples integrated. This implies that these web-applications do not allow the upload, management or analyses of novel datasets. In addition, sharing data or results with collaborators is not possible at all or only indirectly via an unique session key that had to be passed to the collaborators. Thus, it is overly complicated to work in a collaborative manner with others by using these tools.

In this chapter we introduce the cloud-service and web-application `easyGWAS` [Grimm

et al., 2012]. Therefore, we utilise the `easyGWASCore` API for the computation of GWASs and meta-analyses in the backend of the web-application. In addition, `easyGWAS` offers a straightforward way to upload new datasets for an arbitrary species, as well as methods to share data and results with collaborators. Furthermore, already computed GWASs and meta-analyses can be grouped into different projects and made publicly available to the research community. GWASs and meta-analyses can be performed either on publicly available data, on private data or on shared data. Since, data privacy is a central concern of `easyGWAS` we give users the opportunity to share their data with collaborators in such a way that the collaborators can perform GWASs on the data but will not have full access to the raw genotypes. In contrast to other web-applications, we also allow users to upload summary statistics of already computed GWASs for further investigation or the comparison to other GWASs stored in the `easyGWAS` data repository. The data repository facilitates the storage and management of data for different species (e.g. genotype, phenotype or covariate data) and represents the data in a structured and clear way. In addition, `easyGWAS` offers dynamic visualisation and annotation of results for all conducted GWASs and meta-analyses out of the box.

4.1 Architectural and Technical Details

The `easyGWAS` web-application is written in Python and builds upon the Django¹ and the `easyGWASCore` framework. Django is a web framework that provides a large collection of methods that help to develop versatile, secure, reliable and scalable web-applications. For example it frees the developer from the tedious task of writing complicated database drivers. Django is open-source and supported by the non-profit Django Software Foundation. The framework follows closely the Model View Controller (MVC) software architecture pattern. MVC tries to separate specific internal algorithmic patterns from pure information related representations. Thus, the *Model* comprises database specific routines, the *View* is the representation to the user and the *Controller* manages the communication between the *Model* and the *View*. For the actual web-design we used `Bootstrap`² a popular HTML5, CSS and JavaScript (JS) framework. `Bootstrap` helps to faster develop a responsive front-end for web-applications. Additionally, we used the `D3.js`³ JavaScript library to create dynamic visualisations, such as zoomable Manhattan plots.

Modern web-applications require the user to dynamically interact with the front-end without reloading the whole page for every single user action. We used a technique called Asynchronous JavaScript And XML (AJAX) to establish a background (asynchronous) communication between the client and the server. Thus, `easyGWAS` can dynamically display or update information in the clients web-browser without reloading or blocking the whole page. AJAX, however, is not an appropriate technique for tasks that might take seconds or even hours to finish. Many long running tasks, such

¹<https://www.djangoproject.com>

²<http://getbootstrap.com>

³<http://d3js.org>

as performing GWASs or parsing uploaded data files, could lead to a heavy load of the web-server (we used the open-source Apache HTTP server⁴). This might have severe consequences when the web-application is accessed by many users simultaneously. In extreme cases this could lead to a server crash, e.g. if more tasks are executed than the server can handle. Therefore, it is imperative to have a robust and scalable task management and scheduling system. For **easyGWAS** we used **Celery**⁵ an asynchronous task queue based on distributed message passing. Message passing is a technique to invoke a certain process or task even over distributed computer systems. We used the message passing server **RabbitMQ**⁶ to distribute tasks from the web-application to the **Celery** queues. Due to this message passing server, queues can be launched at different computing nodes. Thus, **easyGWAS** is highly scalable and can be quickly adjusted to the number of users by removing or adding new computing nodes. Various specialised queues for different types of tasks can be specified for **easyGWAS**. So far, five queues were deployed: (1) the *gwas* queue for performing different GWASs and meta-analyses, (2) the *gwas_permutation* queue for performing permutation based GWAS, (3) the *utils* queue for general data processing and manipulation tasks, (4) the *pdf* queue for exporting JavaScript visualisations into PDF files and (5) the *periodic* queue for any periodic tasks, such as periodic database cleaning tasks.

An additional challenge of performing GWASs in the browser is data storage and data management. This is because usual genotype datasets contain hundreds of samples and thousands to hundreds of thousands of genetic markers. Storing this huge amount of data in a SQL database such as **PostgreSQL** is not feasible, not even for a small number of users. Also, frequently storing, updating and querying user curated gene annotations with several thousands of genes leads rapidly to computational bottlenecks. Especially if several users store their data in the same database. We therefore developed a hybrid database model that is a mixture between a **PostgreSQL** database, as well as user-specific **SQLite** databases and **HDF5** files. The **PostgreSQL** database stores general information about the user, datasets or GWAS projects. Genotype, phenotype, covariate data and results are stored in **HDF5** files and linked to the user profiles. Each gene annotation file is stored in a separate **SQLite** file such that efficient queries for different annotation sets are ensured. A schematic overview of the **easyGWAS** architecture is illustrated in Figure 4.1.

Uploading large files through the web-browser, such as genotype data, is a challenging task. We therefore allow the users to link their personal Dropbox account with the **easyGWAS** web-application. Thus, a user can synchronise their genotype data with Dropbox in such a way that **easyGWAS** can fetch the data from the Dropbox account and integrates it into the user's data repository.

⁴<http://httpd.apache.org>

⁵<http://www.celeryproject.org>

⁶<http://www.rabbitmq.com>

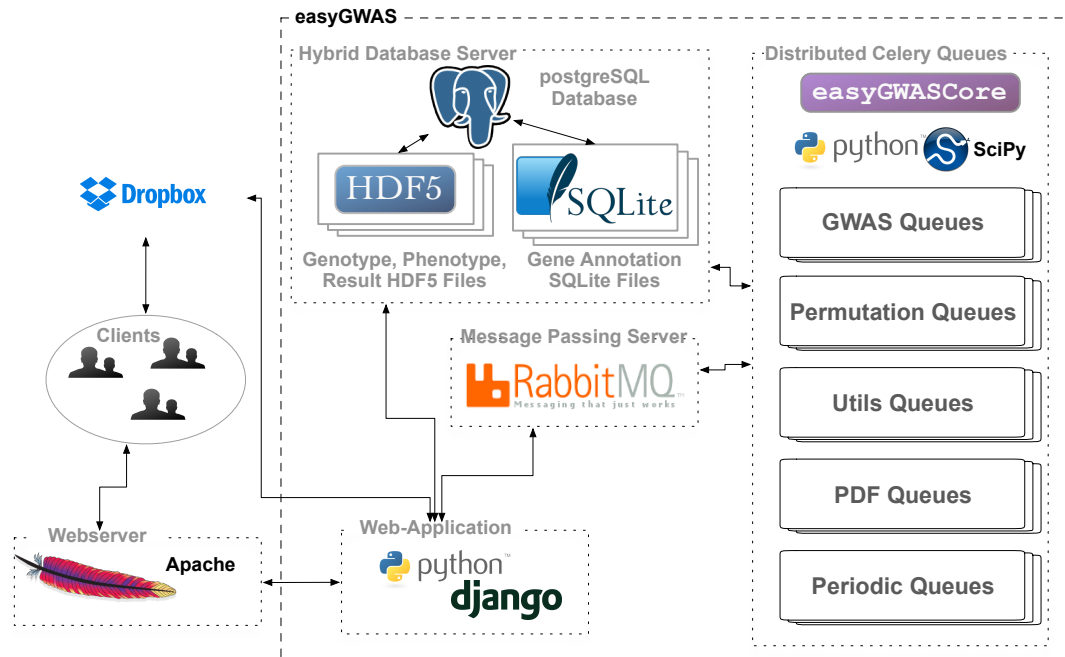


Figure 4.1: Schematics of the easyGWAS architecture: Illustration of the internal architecture of **easyGWAS** including the hybrid database model and different task queues. Communication between the web-application and queues is established via the RabbitMQ message passing server. Task queues can be distributed over different computing nodes. The hybrid database can be accessed from the web-application, as well as from the different task queues. Users can link their personal Dropbox account to **easyGWAS** to integrate large genotype datasets in **easyGWAS** (Dropbox and the Dropbox logo are trademarks of Dropbox, Inc.).

4.2 Overview of the easyGWAS web-application

The **easyGWAS** web-application consists of a publicly accessible area and a restricted private area. Users without a valid login credential are only allowed to access publicly available data and GWAS projects but are not allowed to perform GWASs or upload private data. Registered users, however, are allowed to upload, manage and analyse their private data, as well as work with shared and publicly available data. The web-application is structured into two main components, the *Data Repository* and the *GWAS Centre*. In the following sections we will give a brief overview of these two components and their functionality for registered and logged in users.

4.2.1 The easyGWAS Data Repository

The **easyGWAS** data repository includes different functions related to data integration, storage, management and representation. Again, it is structured into a publicly and privately accessible area (Figure 4.2). Publicly and freely available data will be displayed in the *Public Data* view, whereas user specific data is stored in a restricted and secure environment and can be accessed in the *Private Data* section. In addition, data that is shared with other collaborators will also be displayed in the *Private Data* section. The data of GWASs, which is organised in a hierarchical way, we refer to as a *Data Bag*. Each user can own and store several data bags. However, a single data bag can also be linked to several users (Figure 4.3) in the case the owner of the

The screenshot shows the 'easyGWAS' web application interface. At the top, there is a navigation bar with 'Home', 'GWAS Center', 'Public Data', and 'Private Data' tabs, and a 'Main Navigation Menu' button. Below this, the 'Public Data' section is active, showing a 'Data Repository Navigation Menu' with options like 'Public Species', 'Public Datasets', 'Public Samples', 'Public Phenotypes', and 'Public Covariates'. The main content area is titled 'Detailed Data View (here, Species View)' for 'Species: Arabidopsis thaliana'. It features a general information section with fields for Name, NCBI Taxid, and Description. Below this are two tables: 'Available Datasets' and 'Available Gene-Annotation Sets'.

Dataset Name	Build	#Samples	#SNPs	#Chromosomes	#Phenotypes	#Covariates	Shared	Private	Owner
1001 Genomes Data	TAIR10	1135	6973565	5	1	0	✗	✗	dgrimm
ATPolyDB (call method 75, Horton et al.)	TAIR9	1307	214051	5	107	0	✗	✗	dgrimm
80 genomes data (Cao et al.)	TAIR9	80	1438752	5	0	0	✗	✗	dgrimm

Gene Annotation Set Name	Number of Genes
Gene Annotations (TAIR10)	28496
Gene Annotations (TAIR9)	28412

Figure 4.2: Data repository view: The data repository consists of a public and private data section. The greenish menu on the left side allows the user to access different types of information, such as information about integrated species or phenotypes. The blueish menu in the lower left part offers methods for data management, such as data up- and download.

data bag decides to share it with other users. The root model of such a data bag represents the species data model for which the GWAS data was collected. Additional information about the species is stored in this data model and can be accessed and edited in the species view (Figure 4.2). A species data model can be linked to several

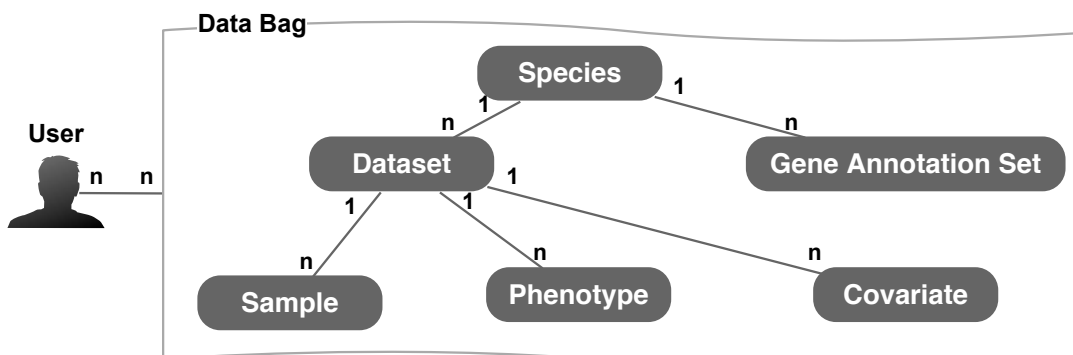


Figure 4.3: Data organisation schematic: Illustration of the data organisation. A user can have several data bags. A data bag contains several type of information about the GWAS data, e.g. the species, integrated datasets with their samples, phenotypes and covariates.

gene annotation set models. These different gene annotation sets are later used to annotate the results of GWASs. In addition, a species model can be linked to one or several GWAS dataset models. A dataset model contains the actual raw genotype data (stored in a single HDF5 file), as well as additional information about the dataset, such as the total number of samples and SNPs. Different data models for storing sample, phenotype and covariate specific data are linked to each GWAS dataset model. Detailed information for these data models can be obtained from their individual views via the navigation menu (at the top of Figure 4.2). The sample view provides useful sample specific meta-information, such as its origin or its source (Figure 4.4). How-

The screenshot displays the 'Sample: TD-1' view in the easyGWAS interface. At the top right, there is an 'Edit Information' button. The main content is organized into three panels:

- General Information:** Features a map of Europe on the left with a red pin in Sweden. To the right is a table:

Name:	TD-1
ID:	6188
Species:	Arabidopsis thaliana
Dataset:	AtPolyDB (call method 75, Horton et al.)
Country:	SWE
Region:	S Sweden
Latitude:	55.7683
Longitude:	14.1386
Source:	Mattias, Jakobsson
Site:	TDr
Description:	
- Additional Meta Information:** Includes an 'Add New Meta-Information' button and a table:

Meta Information Field	Meta Information Value
median_intensity	622.0
- Publications:** Includes an 'Add/Remove/Download Publications' button and a table:

authors	title	pub year	journal	volume	pages	doi
Matthew Horton et al.	Genome-wide patterns of genetic variation in worldwide <i>Arabidopsis thaliana</i> accessions from the RegMap panel	2012	Nature Genetics	44	212-216	10.1038/ng.1042

Figure 4.4: Detailed sample view: For each sample **easyGWAS** provides additional meta-information. Meta-information can be added or changed dynamically. Text in red ellipses are brief descriptions about certain functions.

ever, meta-information varies for samples within and between species. That is why we allow registered users to edit and dynamically add different types of meta-information to their private sample models in an interactive way (Figure 4.4). Similar views are implemented for the phenotype and covariate model. An example of such a view is given for the phenotype model in Figure 4.5. The view of the covariate model is similar to those of the phenotype. A histogram of the data distribution and a Shapiro-Wilk test [Shapiro and Wilk, 1965] to test the null hypothesis whether the data could have been drawn from a normal distribution are given in the phenotype and covariate view as well. The **easyGWAS Download Manager** can be used to retrieve different publicly available GWAS datasets in the traditional PLINK data format (Figure 4.6). Furthermore, users can make their private data publicly available to the whole community. An **Upload Manager** is integrated into **easyGWAS** to upload new genotype, phenotype or covariate data. Initially, each user has a total of 5GB of storage available for private data integration. Uploaded data will be securely integrated into the particular user profile. Users can either upload whole GWAS datasets, in PLINK format, or add new phenotypes or covariates to existing datasets. In addition, users can upload summary statistics of precomputed GWASs for visualisation or subsequent meta-analysis. To upload new GWAS datasets, users have to use the *Genotype Upload* functionality. A screenshot of this upload form is shown in Figure 4.7. For this purpose, a single ZIP file has to be created by the user containing at least the genotype data (*.ped and *.map files). Optionally, the ZIP archive can contain phenotype, covariate and gene annotation data, where the phenotype and covariate data have to be in PLINK format and the gene annotation data in GFF format. However, ZIP files cannot be uploaded directly via the web-browser to **easyGWAS**. Therefore, the users have to link their particular Dropbox accounts to **easyGWAS** and select the corresponding file with the Dropbox Chooser (Figure 4.7).

In summary, this section was a brief overview of the **easyGWAS** data repository and

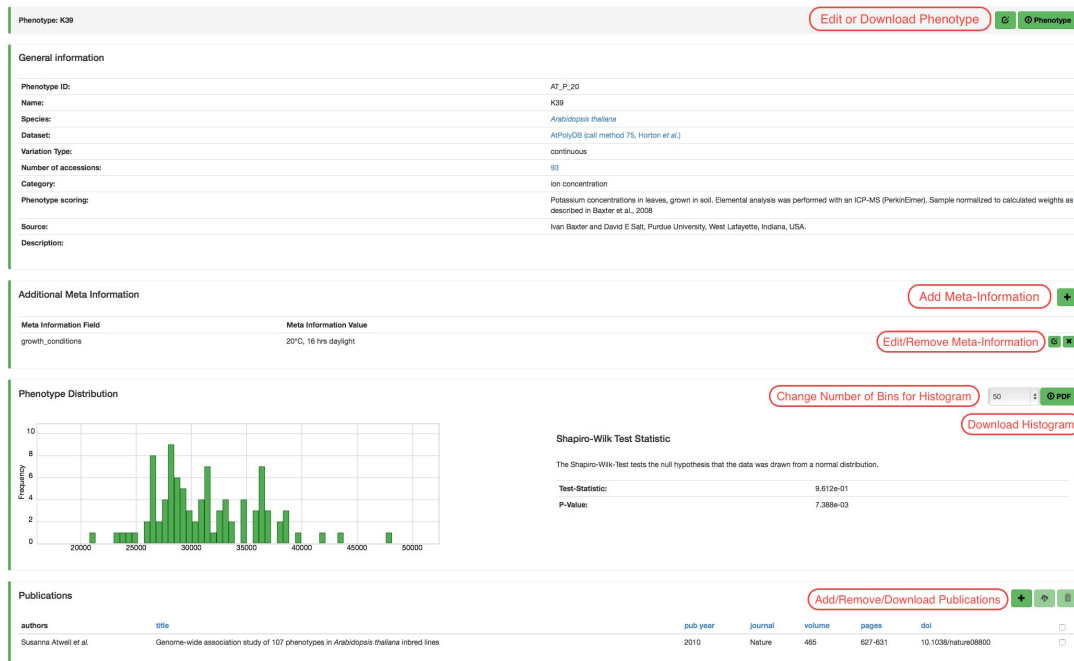


Figure 4.5: Detailed phenotype view: For each phenotype **easyGWAS** provides a detailed view about different types of meta-information. Meta-information can be added or changed dynamically. Several statistics are shown about the distribution of the phenotype. Text in red ellipses are brief descriptions about certain functions.

Download Manager							
Species	Dataset Name	#Samples	#SNPs	#Chromosomes	#Phenotypes	#Covariates	Download
<i>Drosophila melanogaster</i>	<i>Drosophila Genetic Reference Panel (DGRP, Mackay et al.)</i>	172	2476799	5	6	0	
<i>Arabidopsis thaliana</i>	<i>1001 Genomes Data</i>	1135	6973565	5	0	0	
<i>Arabidopsis thaliana</i>	<i>AtPolyDB (call method 75, Horton et al.)</i>	1307	214051	5	107	0	
<i>Arabidopsis thaliana</i>	<i>80 genomes data (Cao et al.)</i>	80	1438752	5	0	0	

Figure 4.6: Data download view: Download manager to download publicly available data.

its functionality. Managing large amounts of data and displaying it in a clear and informative way is crucial for any kind of research. In the next section we present the *GWAS Centre* component which performs and manages GWASs.

4.2.2 The **easyGWAS GWAS Centre**

The **easyGWAS GWAS centre** includes everything that is related with performing, analysing and managing genome-wide association and meta-studies. The publicly accessible area allows every user — including non-registered users — to look at published GWAS projects and allows the download of its summary statistics. Registered users are allowed to perform new GWASs and meta-analyses using publicly available data, as well as their privately uploaded data. Therefore, the users have to use the **easyGWAS GWAS Centre**. The GWAS centre menu is divided into three main parts. The first section contains two wizards for conducting either genome-wide association or meta-studies (Figure 4.8). The second section lists all conducted experiments and offers various different management options. In the third section, different options are available to group experiments into projects, as well as methods to share projects with other

The screenshot shows the 'Upload a new Genotype' form in the easyGWAS interface. The form is divided into several sections:

- Select a publicly available species or create a new one:** Species: Arabidopsis thaliana (with an 'Add new species' button).
- Select a publicly available Gene Annotation Set (Optional):** Gene Annotation Set: Gene Annotations (TAIR10).
- Add information about your dataset (this is needed to upload the data correctly):** Dataset name, Dataset version (0), Dataset description (optional), Make dataset public?, Download of data allowed?
- Data Upload Options:** A note that the data file must be a ZIP file containing genotype files in PLINK format. A checklist for required files (Genotype PED, Genotype MAP, Phenotypes, Covariate, Gene annotation) is shown with checkmarks. Below this, there is a section to 'Choose file from your Dropbox'.

A 'Dropbox Chooser' window is overlaid on the form, displaying a list of files for selection. The files include various genotype files for Arabidopsis thaliana and Capsella flowers. The window has a red border and a title bar that reads 'Dropbox Chooser to Select Files from Personal Dropbox'.

Figure 4.7: GWAS dataset upload view: Form to integrate a new GWAS dataset into easyGWAS. Data in PLINK format has to be stored in a single ZIP file and uploaded to the personal Dropbox account. easyGWAS can then fetch the data from the personal Dropbox account.

collaborators.

The GWAS Wizard

The GWAS wizard (accessible via “New GWAS”) is a step-by-step procedure that guides the user through all the necessary steps to successfully create a GWAS. First the user has to select an existing species, dataset and gene annotation set (if available). This can be either publicly available data for a certain species or a privately integrated one. In the second step, up to five different phenotypes can be selected. The wizard helps the user to find the correct phenotype by offering an autocompletion for all available or shared phenotypes for the selected species and dataset. For each

Transformation	Variation Type	Constraint	Description
Zero Mean	continuous, categorical, binary	–	Mean of data is set to 0
Zero Mean & Unit Variance	continuous, categorical, binary	–	Mean of data is set to 0 and variance is 1
SQRT	continuous, categorical	–	Square root of data
LOG10	continuous, categorical	No “0” in data allowed	Logarithm of data
BOXCOX	continuous, categorical	No “0” in data allowed	Boxcox transformation [Box and Cox, 1964]
Dummy Variable	categorical	data has to be categorical	Encode categorical data into dummy variables

Table 4.1: Available transformation methods: Overview of different methods to transform phenotypes. For each transformation method certain constraints are listed. The GWAS wizard determines on-the-fly which transformation method could be applied to which phenotype.

selected phenotype, detailed information about the data distribution and a Shapiro-Wilk test is shown in the next view (Figure 4.9). Here the user can dynamically explore the effect of different transformation methods on each phenotype. The choice of transformations for each phenotype is automatically determined by the wizard and is based on the *variation type* of the phenotype. Phenotypes are grouped into three main variation types: binary, continuous and categorical. Table 4.1 gives an overview about all available transformation methods. Changing the transformation triggers an

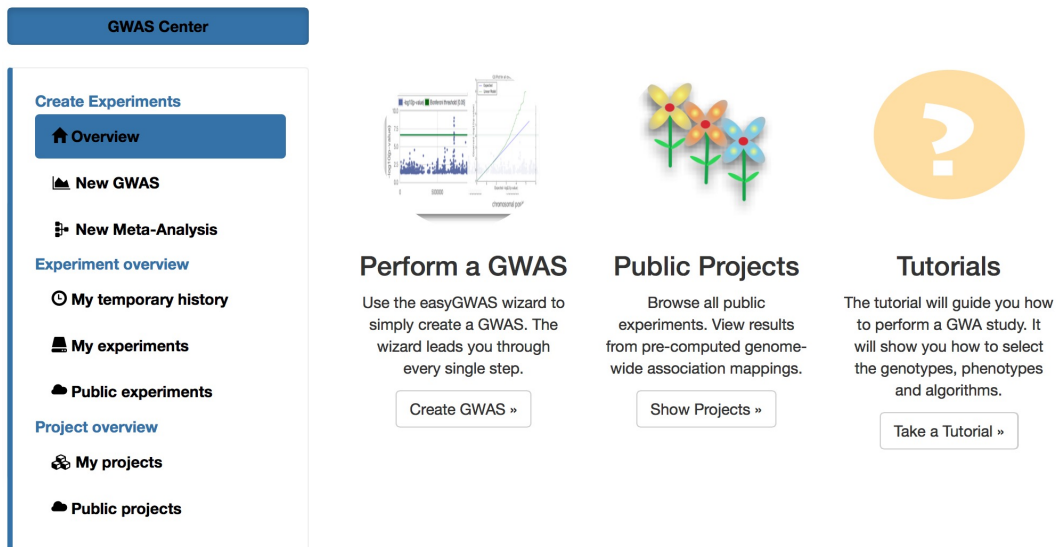


Figure 4.8: The GWAS centre: The easyGWAS GWAS centre is structured into three main areas: (1) methods for performing GWASs and meta-analyses, (2) analyses and management options for study results and (3) management and sharing options for study projects.

AJAX request to the server and the histogram, as well as the Shapiro-Wilk test, are updated dynamically. Thus, the user can easily determine the best transformation for each phenotype. In the next step of the GWAS wizard users can add covariates to their study. Covariates can be used to account for various confounders, such as population structure or environmental factors. The wizard offers three options: (1) Adding no covariates, (2) adding a certain number of principal components or (3) adding several measured covariates to the study. After that, the user is asked by the wizard to select either all available SNPs or certain chromosomes. Next, the wizard offers a selection of different algorithms to perform association tests between the genotype data and the phenotype. The selection is based on the phenotype, as well as on the genotype and is automatically determined by the wizard. An overview of all available algorithms and their data constraints are listed in Table 4.2. In addition, different filtering and

Algorithm	Homozygous	Heterozygous	Binary	Continuous	Categorical	Covariates
Wilcoxon Rank-Sum Test	✓	✗	✓	✓	✓	✗
Linear Regression	✓	✓	✓(△)	✓	✓(△)	✓
Logistic Regression	✓	✓	✓	✗	✗	✓
EMMAX	✓	✓	✓	✓	✓	✓
FaSTLMM	✓	✓	✓	✓	✓	✓
Linear Regression (Permutation)	✓	✓	✓(△)	✓	✓(△)	✓
Logistic Regression (Permutation)	✓	✓	✓	✗	✗	✓
EMMAX (Permutation)	✓	✓	✓	✓	✓	✓

Table 4.2: Available algorithms: Overview of all available algorithms in easyGWAS. The columns *Homozygous* and *Heterozygous* indicate whether the algorithm can be used with homozygous or heterozygous data, respectively. The columns *Binary*, *Continuous* and *Categorical* indicate whether the algorithm supports binary, continuous or categorical phenotypes. Additionally, the column *Covariates* indicates whether covariates can be added to the model. △ means that the model can be used with that type of data but it is not recommended.

encoding options can be selected. The user can select a minor allele frequency (MAF) filter to exclude SNPs that do not fulfil a certain allele frequency. This is especially useful for small populations to prevent spurious associations. For heterozygous genotypes, different genotype encodings can be selected. The default encoding, known as

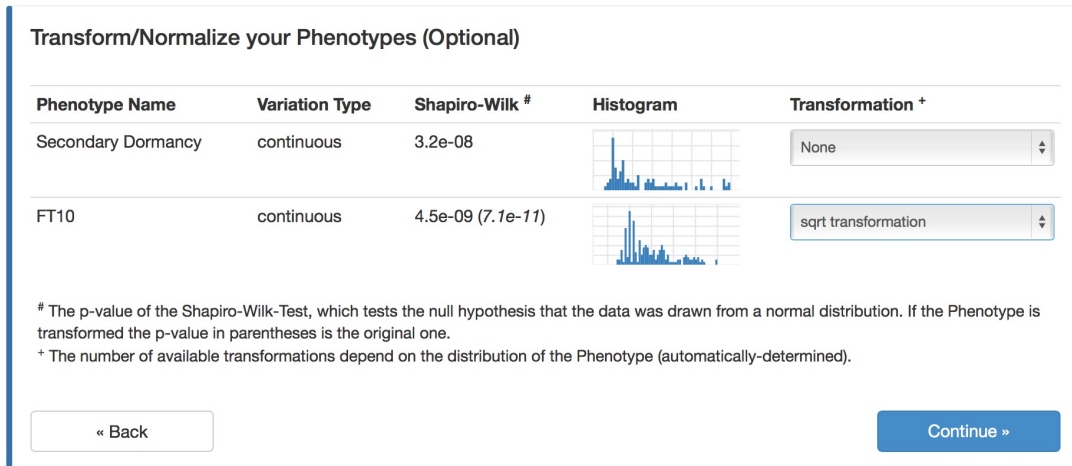


Figure 4.9: Transformation and normalisation view of the easyGWAS GWAS wizard: Data distributions and Shapiro-Wilk test for the selected phenotypes are shown. Different normalisation techniques can be applied to normalise the data. The Shapiro-Wilk test and histogram are updated dynamically for different normalisation functions.

the *additive* encoding is the one in which the major allele is encoded as 0, the heterozygous allele as 1 and the minor allele as 2. An overview of all encodings is given in the Appendix C.2. Finally, a summary page is shown such that the user can check all inputs, adjust them if necessary and submit the experiments to the computation queues.

The Meta-Analysis Wizard

Creating a meta-analysis for precomputed GWASs is as easy as creating GWASs. The meta-analysis wizard can be accessed via the side menu (*New Meta-Analysis*). Similarly to the GWAS wizard one has to select a species and gene-annotation set in the first step. A meta-analysis can be conducted across different datasets, thus it is not necessary to specify a certain dataset. In the second step the user can select different publicly and privately conducted GWASs by using an autocompletion. In the third step of the wizard an algorithm has to be selected. In Table 4.3 an overview of all available meta-analyses algorithms is shown. The wizard offers only those algorithms to the user

Algorithm	P-Value	Effect Size	Brief Description
Fisher's Method	✓	✗	Fisher's method to combine p-values from several studies
Stouffer's Z	✓	✗	Stouffer's Z combines z-scores derived from p-values of several studies
Stouffer's Z Weighted	✓	✗	Combines z-scores and weights each study by $\sqrt{\#samples}$
Fixed Effect Model	✗	✓	Combines effect sizes and assumes that effects are fixed for each study
Random Effect Model	✗	✓	Combines effect sizes and assumes that they arise randomly

Table 4.3: Available meta-analysis algorithms: Overview of all available meta-analyses algorithms in easyGWAS. The columns *P-Value* and *Effect Size* indicate whether the algorithm needs p-values or effect sizes as input.

that can be used with the selected studies, for example fixed and random effect models are based on the estimated effect sizes and their standard deviations. Subsequently, a summary page is shown with the selected species, algorithm and GWASs, as illustrated in Figure 4.10. After this, the meta-analysis can be submitted to the computation queues.

1. Select Species / 2. Select Experiments / 3. Select Algorithms / 4. Summary

General Summary

Selected Species: *Arabidopsis thaliana*

Selected Algorithm: fixedeffect

Ignore SNPs that are not shared between studies:

Selected Experiments

Experiment Name	Species	Dataset	Phenotype	Project	Private
FT16 - Flowering Phenotype	<i>Arabidopsis thaliana</i>	AtPolyDB (call method 75, Horton et al.)	FT16	Genome-wide association study of 107 phenotypes (Atwell et al.)	<input checked="" type="checkbox"/>
Sulfur (S34) - Ionomics Phenotypes	<i>Arabidopsis thaliana</i>	AtPolyDB (call method 75, Horton et al.)	S34	Genome-wide association study of 107 phenotypes (Atwell et al.)	<input checked="" type="checkbox"/>
Leaf serr 22 - Developmental Phenotype	<i>Arabidopsis thaliana</i>	AtPolyDB (call method 75, Horton et al.)	Leaf serr 22	Genome-wide association study of 107 phenotypes (Atwell et al.)	<input checked="" type="checkbox"/>
As2 - Defense-related Phenotype	<i>Arabidopsis thaliana</i>	AtPolyDB (call method 75, Horton et al.)	As2	Genome-wide association study of 107 phenotypes (Atwell et al.)	<input checked="" type="checkbox"/>
Bacterial titer - Defense-related Phenotype	<i>Arabidopsis thaliana</i>	AtPolyDB (call method 75, Horton et al.)	Bacterial titer	Genome-wide association study of 107 phenotypes (Atwell et al.)	<input checked="" type="checkbox"/>
Flowering time FT22	<i>Arabidopsis thaliana</i>	AtPolyDB (call method 75, Horton et al.)	FT22	General	<input checked="" type="checkbox"/>
FLC example	<i>Arabidopsis thaliana</i>	AtPolyDB (call method 75, Horton et al.)	FLC	General	<input checked="" type="checkbox"/>

Back Submit

Figure 4.10: Screenshot of the meta-analysis summary page: The meta-analysis summary page is the final view of the meta-analysis wizard. The user can check if everything is selected correctly and submit the analysis.

The Experiment and Project Overview

All submitted or finished experiments are initially presented in the “*My temporary history*” view (Figure 4.11). Here, users can check the current status of each submitted

Temporary Experiments Save Selected Experiments →

Temporary experiments are available for 48h! To store them permanently please save the experiments. Number of Running Experiments: 2

Type	Experiment Name	Species	Phenotype	Algorithm	date	Checkboxes
	Experiment 0	<i>Arabidopsis thaliana</i>	Secondary Dormancy	FaSTLMM	May. 19, 2015, 04:46 AM	<input checked="" type="checkbox"/>
	Experiment 2	<i>Arabidopsis thaliana</i>	<i>avrRpm1</i>	Logistic Regression	May. 19, 2015, 04:46 AM	<input checked="" type="checkbox"/>
	Experiment 1	<i>Arabidopsis thaliana</i>	Fe56	Linear Regression	May. 19, 2015, 04:46 AM	<input type="checkbox"/>




Figure 4.11: Temporary history view: All submitted and finished experiments are initially stored in a temporary list. This list is automatically cleaned after 48h. Experiments can be either deleted or stored permanently.

experiment. The symbol to the left of each experiment in Figure 4.11 indicates the *type* of the experiment. By now, three different types of experiments exist: (1) a genome wide association study () (2) a meta-analysis of several GWASs () and (3) uploaded summary statistics of a precomputed GWAS (). A spinning wheel indicates if an experiment is still running. Finished experiments are labeled as *Done*. Additionally, a progress bar summarises how many experiments are currently running. Each user is allowed to perform a maximum of five experiments in parallel and all experiments are stored for a maximum of 48 hours. After this time, the history is purged automatically. Both restrictions are important to save computational resources, as well as storage on the server. This ensures that as many users as possible can profit from this web-service simultaneously. However, experiments can also be saved permanently to the user’s profile. For this purpose, we added a checkbox next to each experiment (Figure 4.11). A *Save Experiments* button can then be used to store all available experiments permanently. Permanently saved experiments are grouped into projects (Figure 4.12). Users have the choice to create a new project or add the selected experiments to an existing one. All permanently saved experiments are presented in

Add your experiments to a Project

Select a Project (default: General): General + Add new project

Update your Experiment Names

Type	Experiment Name	Phenotype	Covariates	Algorithm	Date
	<input type="text" value="Experiment 1"/>	Fe56	None	Linear Regression	May 18, 2015, 11:59 a.m.
	<input type="text" value="Experiment 0"/>	Secondary Dormancy	None	FaSTLMM	May 18, 2015, 11:59 a.m.
	<input type="text" value="Experiment 2"/>	<i>avrRpm1</i>	None	Logistic Regression	May 18, 2015, 11:59 a.m.

« Back

Save experiment »

Figure 4.12: Save experiments permanently: Form to save experiments permanently. Experiments are always grouped into an existing or new project. The names of the experiments can be changed.

the “*My experiments*” view. In this view experiments can be filtered by projects, as well as grouped into new or other projects. All private and shared projects are listed in the “*My projects*” view. Projects can be shared with other registered users. For this purpose, the user has to invite the other user by entering the e-mail address of the other user. An e-mail notification is then sent to the other user and the new project with all associated experiments, datasets, phenotypes and covariates is automatically linked to the other user’s profile. In addition, projects can also be made publicly available by using the “*Publish Projects*” button. In that case a publishing inquiry is sent to the administrator of the website. After the administrator approves the request, the project and experiments are moved to the public sections of the web-application and can be accessed by all users, including non-registered users. Published experiments cannot be changed or deleted by the former owner of the data. Data and experiments from shared and published projects can be re-used for additional experiments, such as meta-studies or replication of GWASs.

The easyGWAS Results View

Each performed and stored experiment is linked to its particular HDF5 result file. By clicking the experiment name in any of the experiment overview tables (temporary, private or public) the user will be redirected to the detailed results view. The view is divided into several parts as illustrated in Figure 4.13. On the left side a brief summary is shown, including the selected species, dataset and phenotype, as well as an overview about the most important settings. Additionally, a ranking of the top ten associated hits and their genes is listed. Note that these SNPs may not necessarily be significantly associated SNPs. A new sub-menu is added at the top of the larger panel in the right side. This menu helps the user to navigate through the different result views. The first view — “*Manhattan Plots*” — is the default view for each experiment and shows dynamic Manhattan plots. Manhattan plots are generated for each selected/available

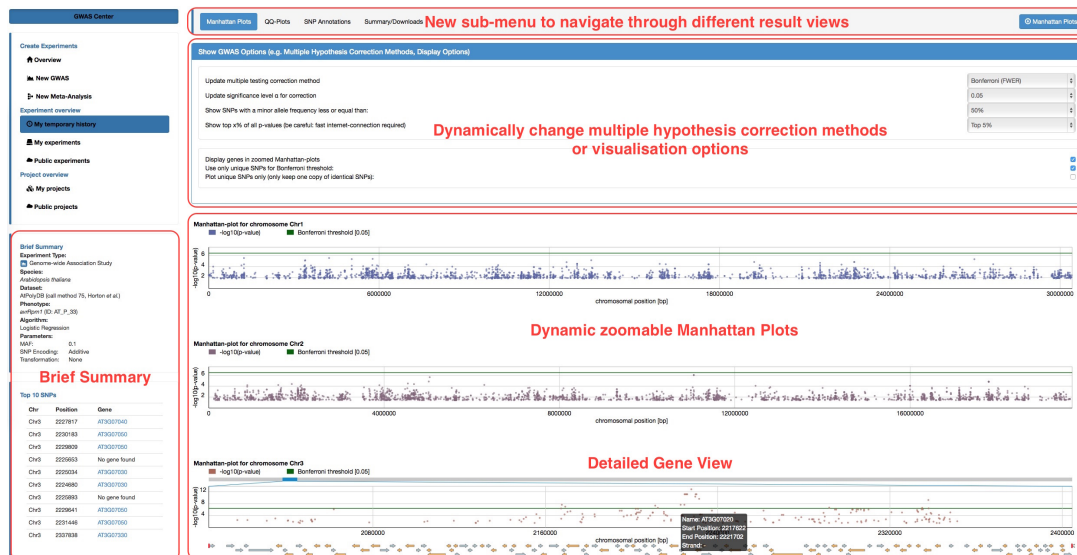


Figure 4.13: GWAS result view: On the left side a brief summary of the experiment is displayed together with information about the top 10 associated hits. The right shows dynamic zoomable Manhattan plots. Different multiple hypothesis correction methods are available, as well as different options to dynamically adjust the Manhattan plots. A new sub-menu is shown in the top of the result view to navigate through different result views.

chromosome. The green line in each Manhattan plot illustrates the global multiple hypothesis correction threshold. As a default, a Bonferroni correction [Abdi, 2007] is used to control the family-wise error rate (FWER) with a significance level α equal to 5%. The multiple hypothesis correction method and the significance level α can be changed dynamically by the user. Each change triggers an AJAX request and the plots are instantly updated. Three additional methods are available to correct the false discovery rate (FDR), *Benjamini and Hochberg* [1995], *Benjamini and Yekutieli* [2001] and *Storey and Tibshirani* [2003]. All Manhattan plots are zoomable. Thus, users can zoom into different regions of interest to explore this region in more detail. A gene-view will be displayed for these regions if the dataset was linked to a gene-annotation set (Figure 4.13). All Manhattan plots can be downloaded as PDFs to facilitate the integration of these plots into manuscripts.

The second view — “*QQ-Plots*” — shows QQ-plots using all available test statistics or only those for a certain chromosome. Additionally, the genomic control factor λ is computed and added to the plot [Devlin and Roeder, 1999]. The third view — “*SNP Annotations*” — lists gene annotations for the top associated SNPs per chromosome. Here, the user can change the number of SNPs for which annotations should be retrieved, the multiple hypothesis correction method to label SNPs as significantly associated or not, as well as a search window to find genes upstream or downstream of each SNP.

Finally, the last view — “*Summary/Downloads*” — contains a detailed summary of the whole experiment. The overview shows the owner of the experiment, when the experiment started and when it ended. Detailed information about all selected options and information about the algorithm are listed as well (see Figure 4.14). Adding too many covariates to regression based models can easily lead to overfitting, e.g. the

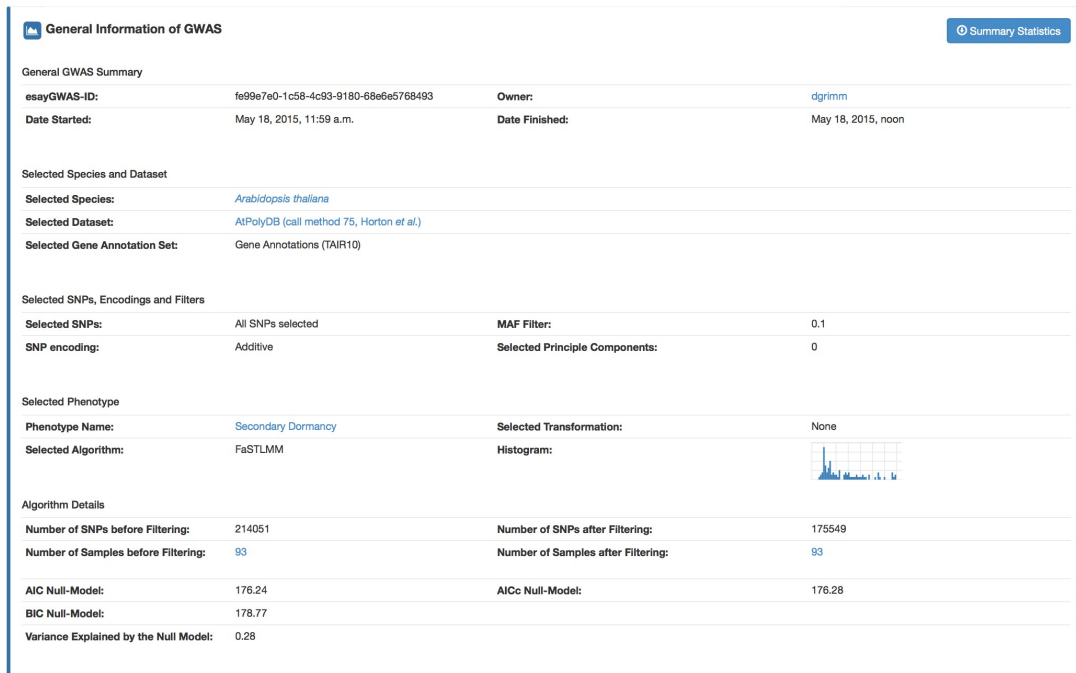


Figure 4.14: GWAS result summary: Summary of the experiment results. Detailed overview about all selected data sources and parameters.

goodness-of-fit measure R^2 will increase when adding additional covariates. Thus, it is important to find a good balance between the goodness of a fit and the model complexity. We therefore included three model selection parameters for regression based models to measure the relative quality of the models for the given dataset and to penalise models with high complexities (Figure 4.14). One of the most widely used methods the Akaike Information Criterion (AIC) is defined as:

$$\text{AIC} = -2l(\hat{\beta}_{ML}, \hat{\sigma}_{ML}^2) + 2d, \quad (4.1)$$

where $l(\hat{\beta}_{ML}, \hat{\sigma}_{ML}^2)$ is the log-likelihood function and d the total number of fixed effects in the model. The term $2d$ penalises overly complex models. Thus, models with small AIC values are preferred. The second measure, AICc, is a corrected version of AIC since the number of samples is finite and is defined for models with Gaussian distributed residuals as follows:

$$\text{AICc} = \text{AIC} + \frac{2d(d+1)}{n-d-1}, \quad (4.2)$$

where d is the number of fixed effects and n the number of samples. The last criterion easyGWAS provides is the Bayesian Information Criterion (BIC). This criterion favours even more parsimonious models than AIC because it penalises complex models even more. BIC is defined as:

$$\text{BIC} = -2l(\hat{\beta}_{ML}, \hat{\sigma}_{ML}^2) + d \log n, \quad (4.3)$$

where d is the number of fixed effects and n the number of samples. A detailed overview of AIC and BIC is given in *Burnham and Anderson [2002]* and *Aho et al. [2014]*.

In addition, **easyGWAS** computes how much of the phenotypic variance could be explained by the null-model when using a linear mixed model, such as **EMMAX** or **FaSTLMM**. Here **easyGWAS** computes, in a 10-fold cross-validation, which parts of the phenotypic variance could be attributed to the genetic contribution (random effect), using the kinship matrix only, and to the covariates (fixed effects). For this purpose, the data is split into ten subsets of equal size (to the extent possible). Then, nine subsets are combined to train a linear mixed model using only the kinship matrix and the covariates. The remaining subset is used to predict the phenotype $\hat{\mathbf{y}}$. This procedure is repeated ten times. Predictions for $\hat{\mathbf{y}}$ are obtained by summing up the contributions of the random and fixed effects as follows:

$$\hat{\mathbf{y}} = \mathbf{C}_{\text{test}}\hat{\boldsymbol{\beta}} + \mathbf{K}_{\text{test}} \left(\mathbf{K}_{\text{train}} + \hat{\boldsymbol{\delta}}\mathbf{I} \right)^{-1} \left(\mathbf{y}_{\text{train}} - \mathbf{C}_{\text{train}}\hat{\boldsymbol{\beta}} \right), \quad (4.4)$$

where \mathbf{C} are the included covariates (or a vector of ones if no covariates are included), \mathbf{K} is the kinship matrix, and $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\delta}}$ are the estimated parameters from the training step. The indices *train* and *test* indicate whether the data is coming from the training or testing subsets. Eventually, we can compute the variance explained as follows:

$$v(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}}) = 1 - \frac{\text{Var}(\mathbf{y}_{\text{test}} - \hat{\mathbf{y}})}{\text{Var}(\mathbf{y}_{\text{test}})}. \quad (4.5)$$

The variance explained is then shown in the **easyGWAS** summary page (Figure 4.14). All summary statistics and estimated parameters can be downloaded for further analysis using third-party tools.

In this section we gave a brief introduction of the **easyGWAS** web-application and its different views. In the following sections we will give an overview of all publicly available datasets in **easyGWAS**. Furthermore, we will demonstrate the functionality of the web-application by performing a toy example in *Arabidopsis thaliana*.

4.3 Publicly Available Data

One of the primary advantages of **easyGWAS** is that it also serves as a public data repository for GWASs data of various species and datasets. Thus, scientists have a central platform to access publicly available data and to reproduce or conduct GWASs on this data. As of June 2015, data for *Arabidopsis thaliana*, *Drosophila melanogaster* and *Pristionchus pacificus* are available in our public data repository.

Various data sources are integrated for *Arabidopsis thaliana* [Atwell *et al.*, 2010; Cao *et al.*, 2011; Horton *et al.*, 2012; Long *et al.*, 2013; Schmitz *et al.*, 2013]. The *AtPolyDB* dataset⁷ includes a set of 1,307 worldwide *Arabidopsis thaliana* accessions with a total of 214,051 SNPs sequenced with a 250k SNP chip [Horton *et al.*, 2012]. A total of 107 dichotomous, continuous and categorical phenotypes⁸ are integrated for a subset of these 1,307 accessions [Atwell *et al.*, 2010]. The phenotypes are grouped into four

⁷<https://cynin.gmi.oeaw.ac.at/home/resources/atpolydb/>

⁸<http://arabidopsis.gmi.oeaw.ac.at:5000/DisplayResults/>

main categories: (1) flowering time related phenotypes, (2) defence related phenotypes, (3) ionomic phenotypes and (4) developmental related phenotypes. The *80 genomes* dataset includes 80 accessions from the first phase of the 1,001 genomes project in *Arabidopsis thaliana* [Cao *et al.*, 2011]. The SNP data was retrieved from the original genome matrix from the 1001 genomes website⁹. We excluded all singletons and SNPs with incomplete information, which resulted in a final set of 1,438,752 SNPs. Eventually, we included the latest data from the 1,001 genomes project including a total of 1,135 samples and 6,973,565 non-singleton SNPs. These 1,135 accessions were sequenced in a collaborative manner between the Max Planck Institute (Weigel lab), the Gregor Mendel Institute (Nordborg lab), the Salk Institute (Ecker lab), the Wellcome Trust Center for Human Genetics (Mott lab), University of Chicago (Bergelson lab) and Monsanto using the Illumina platform. Additionally, we integrated TAIR9 and TAIR10 gene annotation sets¹⁰.

For the species *Drosophila melanogaster* we integrated the *Drosophila Genetic Reference Panel* (DGRP)¹¹ with a total number of 172 samples, 2,476,799 SNPs and three phenotypes¹² [Harbison *et al.*, 2004; Jordan *et al.*, 2007; Mackay *et al.*, 2012; Morgan and Mackay, 2006]. The three phenotypes are split into male and female phenotypes, which resulted in a final set of six phenotypes. Missing SNPs in the *Drosophila melanogaster* genome are imputed using a majority allele imputation. Gene annotations were downloaded from the FlyBase website¹³ and integrated into *easyGWAS*.

Finally, we integrated a total of 149 samples with 2,135,350 SNPs of the species *Pristionchus pacificus* ([McGaughran *et al.*, 2015], journal publication under preparation). In addition, three phenotypes and four covariates were integrated.

4.4 Case Study in *Arabidopsis thaliana*

In this section we will demonstrate the functionality of *easyGWAS* by conducting a GWAS and meta-analysis for the phenotype 4W in *Arabidopsis thaliana*. The phenotype was collected for a total of 119 accessions and measures the number of days to flowering time under long days (16h daylight at 23°C) with four weeks of vernalisation (at 5°C with 8h daylight) [Atwell *et al.*, 2010; Zhao *et al.*, 2007]. All 119 accessions can be found in the *AtPolyDB* dataset sequenced with a 250k SNP chip [Atwell *et al.*, 2010; Horton *et al.*, 2012]. A subset of 79 accessions, with a total of 6,973,565 non-singleton SNPs, are available in the latest 1,001 Genomes dataset sequenced using NGS platforms. In this case study we artificially split the phenotype into two parts. One part contains all 79 samples available in the latest 1,001 Genomes dataset (4W-1001) and the other part contains the remaining 40 samples which are available in the *AtPolyDB* (4W-AtPolyDB).

First we performed a GWAS using the original phenotype with all 119 samples and the

⁹<http://1001genomes.org/data/MPI/MPICao2010/releases/>

¹⁰<http://www.arabidopsis.org>

¹¹http://dgrp.gnets.ncsu.edu/freeze1/Illumina_+_454_SNP_genotypes_filtered_for_GWAS/

¹²<http://dgrp.gnets.ncsu.edu/freeze1/Phenotypes/>

¹³ftp://ftp.flybase.net/releases/FB2008_10/dmel_r5.13/gff/

AtPolyDB dataset. Therefore, we used the *GWAS Wizard* and selected the *AtPolyDB* dataset for the species *Arabidopsis thaliana*. In the second step we selected the phenotype 4W from the list of all publicly available phenotypes. In the third step a phenotype transformation can be chosen. The Shapiro-Wilk test for the non-transformed phenotype has a p-value of $2e^{-14}$. Thus, the null hypothesis that the data was drawn from a normal distribution can be rejected. However, regression based models assume normally distributed residuals. Because of that, we normalised the phenotype by choosing the box-cox transformation. Consequently, the Shapiro-Wilk test was updated dynamically and reported a p-value of $1e^{-5}$. In the next steps, we selected no covariates and all available SNPs with an allele frequency higher than 5%. Because *Arabidopsis thaliana* has a high degree of population structure, we chose *FaSTLMM* to account for hidden confounding due to population stratification. Finally, we submitted the experiment to the *easyGWAS* computation queue. Computations were finished within one minute and stored in the temporary user history of *easyGWAS*. The results are illustrated as a Manhattan and QQ-plot in Figure 4.15. On chromosome 1 and position 3,978,064 a

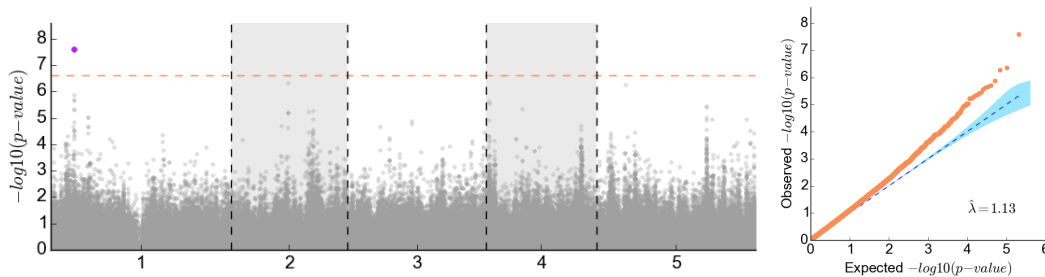


Figure 4.15: Manhattan plot and QQ-plot for the original phenotype 4W: Results for the original 4W phenotype using all 119 samples in the *AtPolyDB* dataset. The phenotype is box-cox transformed and a MAF filter of 5% was applied. Only one hit was found to be significantly associated after Bonferroni correction.

significantly associated SNP is reported by *easyGWAS* (p-value = $2.6e^{-8}$) after correcting for multiple hypothesis using the Bonferroni method ($0.05/206,022 = 2.4e^{-7}$). The associated SNP is located in the gene AT1G11780, which is a 2-oxoglutarate/Fe(II)-dependent dioxygenases protein.

Next, we repeated the previous steps but selected the 4W-*AtPolyDB* (40 samples) phenotype for the *AtPolyDB* dataset and the 4W-1001 phenotype (79 samples) for the 1,001 Genomes dataset. Again, we applied a box-cox transformation to both phenotypes and selected a minor allele frequency filter of 5% for both datasets. Manhattan and QQ-plots are shown in Figure 4.16 for the *AtPolyDB* analysis and in Figure 4.17 for the 1,001 Genomes analysis. In Figure 4.16 we observe that no significantly associated markers could be detected. This could be due to the smaller sample size of only 40 accessions. For the phenotype 4W-1001 — which has a total of 79 samples — we find a total of 11 significantly associated hits (Figure 4.17). Four significantly associated hits on chromosome 2 are located in the gene AT2G22540. This well-known gene — also referred to as SHORT VEGETATIVE PHASE (SVP) — plays an important role in the control of flowering time by negatively regulating the expression of the floral integrator FLOWERING LOCUS T (FT) [Lee *et al.*, 2007].

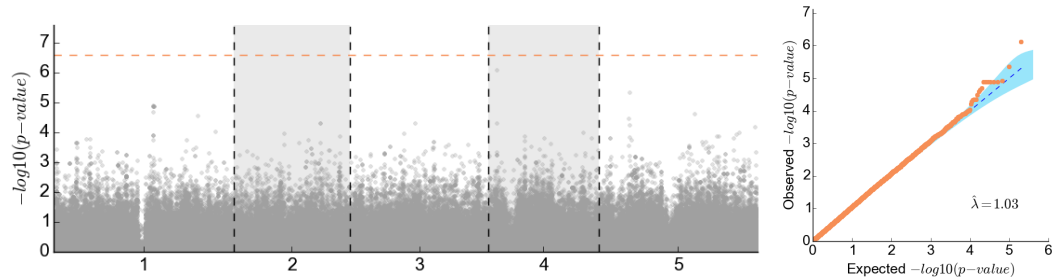


Figure 4.16: Manhattan plot and QQ-plot for the phenotype 4W-AtPolyDB: Results for the 4W-AtPolyDB phenotype using a subset of 40 samples on the *AtPolyDB* dataset. The phenotype is box-cox transformed and a MAF filter of 5% was applied.

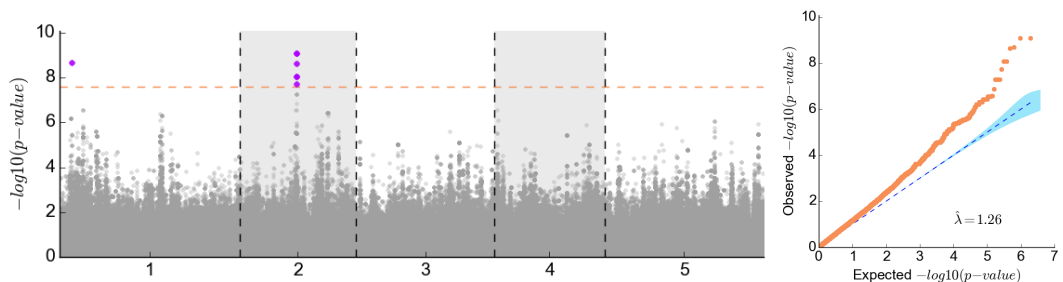


Figure 4.17: Manhattan plot and QQ-plot for the phenotype 4W-1001: Results for the original 4W-1001 phenotype using a subset of 79 samples on the 1,001 Genomes dataset. The phenotype is box-cox transformed and a MAF filter of 5% was applied.

Next, we grouped all these experiments into a common project and saved them permanently in the users profile. We then conducted a meta-analysis by using the *easyGWAS Meta-Analysis Wizard*. Here, we first selected the species *Arabidopsis thaliana* and the two previously computed GWASs for the phenotypes 4W-AtPolyDB and 4W-1001. Eventually, we selected Stouffer's weighted Z-score method to perform the actual meta-analysis. We chose the weighted version of Stouffer's method to differently weight the two experiments based on the number of samples. The Manhattan plot for the meta-analysis is shown in Figure 4.18. The meta-analysis reveals a total number of

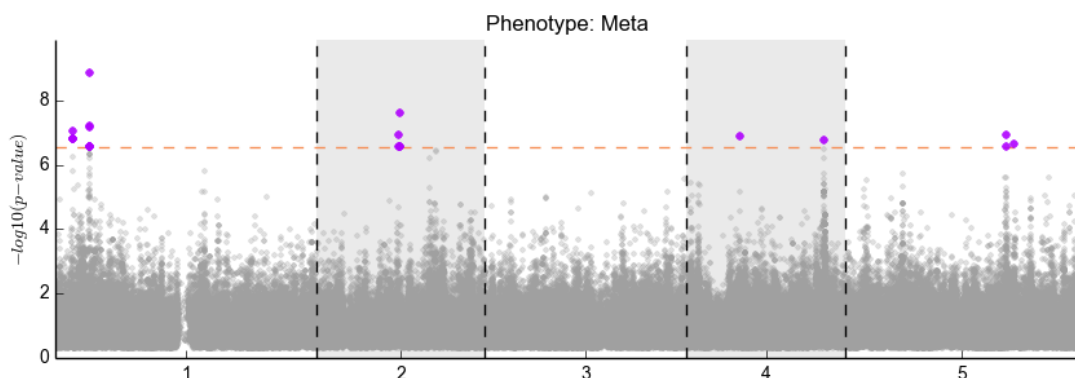
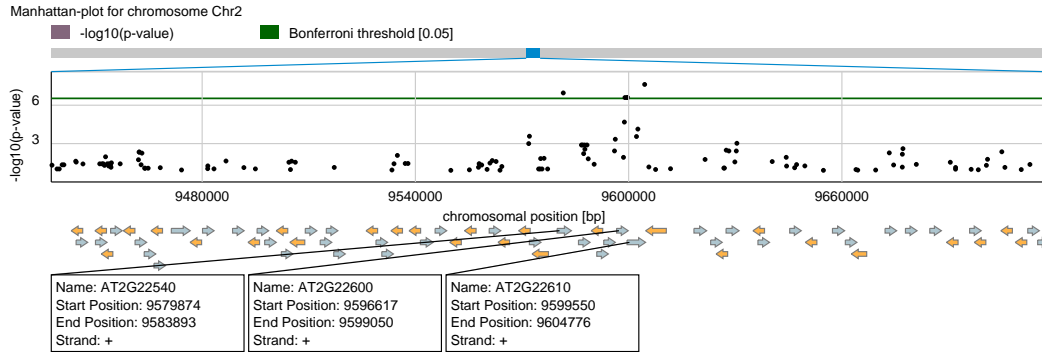


Figure 4.18: Manhattan plot of meta-analysis: Meta-analysis results combining the summary statistics for the GWASs on the phenotypes 4W-AtPolyDB and 4W-1001.

21 significantly associated hits. Interestingly, we find significantly associated hits on chromosome 4 and 5 that could not be detected with any of the previously computed

GWASs. From these 21 hits, 17 are located within 12 genes. Again, we detect a significant association with the SVP gene (AT2G22540) on chromosome 2 (Figure 4.19a). On chromosome 5 we find a significant association with a SNP located in a gene (AT5G45830) that encodes DELAY OF GERMINATION 1 (DOG1) (Figure 4.19b). This gene was found to not only influence germination, but also flowering time [Chiang

(a) Significant association with SVP



(b) Significant association with DOG1

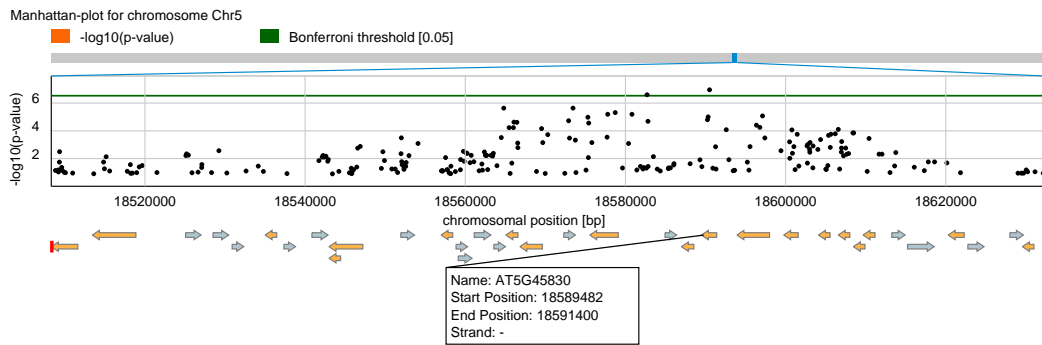


Figure 4.19: Zoomed gene annotation plots: Zoomed in Manhattan plots downloaded from *easyGWAS*. Two SNPs are associated with two well-known genes SVP and DOG1.

et al., 2013]. In *Atwell et al.* [2010] DOG1 was found to be highly associated in 20 different flowering time related phenotypes. However, in *Atwell et al.* [2010], DOG1 was found to be highly but not significantly associated for the phenotype 4W when using EMMAX.

We here demonstrated in a case study the capabilities of *easyGWAS* to perform GWASs and meta-analyses. We detected two novel associations in flowering time related phenotypes in the genes SVP and DOG1 that could not be detected with a traditional GWAS but when using a meta-analysis.

4.5 Chapter Summary

In this chapter we introduced a novel cloud-service and web-application for performing GWASs and meta-analyses via a web-browser. First, we gave a detailed overview about the architecture and design of *easyGWAS*. We developed a novel hybrid database scheme to efficiently store and mine different types of data. For this purpose, we used (i) a

standard `PostgreSQL` database for storing general information such as user and experiment specific data, (ii) a collection of `SQLite` databases for storing data related for the annotation of results and (iii) a collection of `HDF5` files for storing and managing GWASs data, as well as results of computed experiments. Further, we established a comprehensive task queue and message passing system to reliably and efficiently distribute heavy computational tasks to different computation nodes and queues. This is necessary to guarantee a smoothly running and interacting web-application. Thus, users can submit several experiments and computations while at the same time be able to still interact and work with the web-application.

We further gave a detailed description about the `easyGWAS` data repository and its functionality. We showed that it is straightforward to upload new genotype data and gene annotation sets to `easyGWAS` in a secure and private way. Integrated data can be either stored in a private environment or in publicly available manner for the whole community. A major advantage of `easyGWAS` is that private data can be easily shared with other users and collaborators, even in such a way that others can perform GWASs on shared data without having direct access to the original raw genotype data. All data sources and instances are represented in a structured and clean way. Publicly available data can be download by everyone.

To simplify the process of performing GWASs and meta-analyses we created two step-by-step procedures (wizards) that guide the users through every necessary step. Conducted experiments can be grouped into projects and can be made available to the public as well. All conducted experiments come with dynamic visualisations of the the results and with detailed annotations.

Finally, we demonstrated the functionality of `easyGWAS` by conducting a case-study on *Arabidopsis thaliana* genotype and phenotype data. We artificially splitting a phenotype into two parts such that it could be used with two distinct genotype datasets. We performed two GWASs on both datasets and combined the results by performing a meta-analysis. Here, we showed that we could detect novel associations in two flowering time related genes `SVP` and `DOG1` that could not be detected with a standard GWAS.

Both, `easyGWASCore` and `easyGWAS` are powerful tools to conduct GWASs and meta-analyses. However, we mostly concentrated on univariate methods. It has been shown that these methods often fail to explain much of the phenotypic variance [*Manolio et al.*, 2009]. In the next chapter we will describe two novel methods, `SConES` and `Multi-SConES`, for multi-locus and multi-trait mapping that could help to explain parts of this missing heritability. These algorithms make use of prior knowledge, such as gene-pathways or protein-protein interaction networks, to guide the discovery of sets of genetic markers that are highly associated with a given trait or correlated traits.

CHAPTER 5

Network Guided Multi-Locus and Multi-Trait Association Mapping

Identifying *causal* genetic markers, such as single nucleotide polymorphisms (SNPs), that can explain the phenotypic variability of observed traits or complex diseases is one of the ultimate goals in the field of GWASs. As a matter of fact, hundreds of genetic variants associated with complex traits could have been identified by GWASs [Atwell *et al.*, 2010; Zuk *et al.*, 2012]. Univariate statistical tests, however, often fail to explain much of the heritability of these complex phenotypes [Manolio *et al.*, 2009], leading to the problem of the missing heritability. Investigating effects of multiple genetic markers simultaneously, by considering additive or interactive effects between multiple markers, contributed to explain parts of this missing heritability [Marchini *et al.*, 2005]. However, the detection of additive and especially of multiplicative effects between multiple markers leads to a computational and a statistical multiple hypothesis testing problem. Many algorithms have been developed to efficiently detect interaction effects between pairs of genetic markers — often referred to as epistatic effects — using mathematical and algorithmic tricks [Achlioptas *et al.*, 2011; Zhang *et al.*, 2008, 2009, 2010a] or by leveraging graphical processing units (GPUs) [Hemani *et al.*, 2011; Kam-Thong *et al.*, 2011, 2012]. Nevertheless, a tremendous number of multiple hypothesis tests remain, which consequently leads to a larger false negative rate. The detection of additive multivariate associations between more than two markers is feasible due to efficient multiple regression based approaches [Cho *et al.*, 2010; Rakitsch *et al.*, 2013b; Wang *et al.*, 2011], but they are limited in power or hard to interpret. Attempts have been made to include *prior* biological knowledge to boost their statistical power and interpretability. Nonetheless, current methods are limited to a predefined number of potential candidate sets [Cantor *et al.*, 2010; Fridley and Biernacka, 2011; Wu *et al.*, 2011]. Further, the diversity and the current incompleteness of biological knowledge make it almost impossible to provide a complete network in which all the relevant connections are present. We here present a novel framework for detecting sets of genetic markers that are maximally associated with a phenotype while being connected in an underlying biological network [Azencott *et al.*, 2013]. We refer to this method as **SConES**, for Selecting Connected Explanatory SNPs. Our pro-

posed method is exact, efficient and biologically meaningful and scales to datasets and networks with millions of genetic markers and nodes. In addition, it automatically detects sets of genetics markers without the need of providing a predefined number of potential candidate sets. Eventually, **SConES** is able to handle incomplete networks, by identifying different subnetworks that *must not* form a single connected component.

Another important concept in genetics is *pleiotropy*, that is the effect of a single gene or marker to different traits or diseases. A technique called *multi-task learning* can be used to detect markers that are jointly associated with multiple correlated traits. In multi-task learning, a difficult problem is solved together with other related problems. Thus, it is possible to boost the performance of the model by looking at common characteristics across different task (in our case correlated traits) simultaneously [Baxter, 2000; Caruna, 1993]. Different methods have been developed for the detection of sets of genetic markers while considering multiple correlated traits [Korte et al., 2012; Rakitsch et al., 2013a]. An additional series of multi-task models were developed to also include *prior* knowledge, such as different types of networks [Kim et al., 2009; Zhang et al., 2010b; Zhou et al., 2010]. However, these models assume that the same set of features (genetic markers) should be selected across all tasks (correlated traits). While this is reasonable for some applications, several examples can be found for which this assumption is violated. For instance, lung diseases, such as asthma and chronic obstructive pulmonary disease (COPD) may be linked to a set of common genetic variants, but there is no indication that the exact same mutations are causal in both diseases. Different regularisations are used to find a tradeoff between the sparsity and connectivity of features within a given network. However, to the best of our knowledge, none of the structured regularised multi-task methods make it possible to consider different structural constraints for different tasks. Nevertheless, we may want to consider different biological pathways for different phenotypes, or to highlight different parts of brain connectivity networks for different correlated behaviours. Hence, we here present **Multi-SConES**, an extension of **SConES** to a multi-task framework with multiple network regularisers [Sugiyama et al., 2014]. This framework, allows to identify sets of genetic markers that are significantly associated with multiple correlated traits while being connected in at least one underlying biological network.

In the first part of this chapter we will describe the single-task framework **SConES** and demonstrate its abilities on several simulated and real-world experiments. In the second part we will extend the single-task formulation **SConES** to a multi-task formulation **Multi-SConES**. Eventually, we will integrate both methods into the **easyGWASCore** framework to facilitate their usage.

5.1 SConES: Selecting Connected Explanatory SNPs

5.1.1 Method and Problem Formulation

Let us assume we are given a genome matrix \mathbf{M} of size $n \times m$ and a phenotype $\mathbf{y} \in \mathbb{R}^n$, where n is the number of samples and m the number of genetic markers (e.g. SNPs).

The task we would like to solve is a feature selection problem in a graph-structured feature space, where the features are our genetic markers, and the selection criterion should be related to their association with the phenotype \mathbf{y} . The biological network is represented as a weighted graph $G = (V, E)$, where V are the vertices (nodes) and E are the edges. The vertices v are the individual genetic markers (features) and the edges represent if individual markers are connected through any kind of relationship (different types of biological networks will be discussed later). The graph G can be described by its adjacency matrix \mathbf{A} of size $m \times m$.

To measure the dependence $\mathbf{c} \in \mathbb{R}^m$ between each single genetic marker and a single given phenotype \mathbf{y} several techniques can be used, such as Pearson's correlation coefficient, the Hilbert-Schmidt Independence Criterion (HSIC) [Gretton *et al.*, 2005], maximum information coefficient (MIC) [Reshef *et al.*, 2011] or the Sequence Kernel Association Test (SKAT) [Wu *et al.*, 2011]. Under the common assumption that the joint effect of several genetic markers is additive, \mathbf{c} is such that the association between a group of genetic markers and the phenotype \mathbf{y} can be quantified as the sum of the scores of the genetic markers belonging to this group. In other words, given an indicator vector $\mathbf{f} \in \{0, 1\}^m$ such that, for any $p \in \{1, \dots, m\}$, f_p is set to 1 if the p -th genetic marker is selected and 0 otherwise, the score of the selected markers is given by:

$$Q(\mathbf{f}) = \sum_{p=1}^m c_p f_p = \mathbf{c}^\top \mathbf{f}. \quad (5.1)$$

For our framework the association term \mathbf{c} is derived from the weighted linear kernel version of SKAT (Linear SKAT) [Wu *et al.*, 2011], which makes it possible to simultaneously correct for confounders, such as population stratification or environmental factors, by adding different covariates to the model. In its simplest form, SKAT models the relationship of the phenotype \mathbf{y} , a set of covariates $\mathbf{C} \in \mathbb{R}^{n \times d}$ and a subset \mathbf{M}^S of genetic markers by either a classical multiple linear regression for continuous phenotypes \mathbf{y}_c or a multiple logistic regression for dichotomous phenotypes \mathbf{y}_b [Wu *et al.*, 2011]:

$$\mathbf{y}_c = \alpha_0 + \boldsymbol{\alpha}^\top \mathbf{C} + \boldsymbol{\beta}^\top \mathbf{M}^S + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}), \quad (5.2)$$

$$P(\mathbf{y}_b = 1 | \mathbf{M}; \mathbf{C}; \boldsymbol{\beta}) = \alpha_0 + \boldsymbol{\alpha}^\top \mathbf{C} + \boldsymbol{\beta}^\top \mathbf{M}^S, \quad (5.3)$$

where α_0 is an intercept term, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)^\top$ is a vector of regression weights for the d covariates and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_{|S|})^\top$ are the regression weights for the observed markers in the subset \mathbf{M}^S . SKAT tests the null hypothesis $\mathcal{H}_0 : \boldsymbol{\beta} = \mathbf{0}$, that is fitting the following null models without the genetic markers [Wu *et al.*, 2011]:

$$\mathbf{y}_c = \alpha_0 + \boldsymbol{\alpha}^\top \mathbf{C} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}), \quad (5.4)$$

$$P(\mathbf{y}_b = 1 | \mathbf{M}; \mathbf{C}; \boldsymbol{\beta}) = \alpha_0 + \boldsymbol{\alpha}^\top \mathbf{C}. \quad (5.5)$$

The variance component test statistic of SKAT is defined as follows:

$$Q^S = (\mathbf{y} - \hat{\mathbf{y}})^\top \mathbf{K}^S (\mathbf{y} - \hat{\mathbf{y}}), \quad (5.6)$$

where $\hat{\mathbf{y}}$ is the predicted mean of \mathbf{y} under \mathcal{H}_0 . Consequently, $(\mathbf{y} - \hat{\mathbf{y}})$ are the residuals \mathbf{r} of the regression models under \mathcal{H}_0 . \mathbf{K}^S is the $n \times n$ weighted linear kernel function describing the genetic similarity between the n samples, that is:

$$\mathbf{K}^S = \mathbf{M}^S \mathbf{W} (\mathbf{M}^S)^\top, \quad (5.7)$$

where \mathbf{W} is a $n \times n$ diagonal matrix and the p th diagonal element of matrix \mathbf{W} is the weight w_p for the genetic marker p . The weights can be used to adjust the importance of specific markers, e.g. by increasing the importance of rare or pathogenic genetic markers. Thus, the kernel value K_{ij}^S for sample i and j is equal to:

$$K_{ij}^S = \sum_{p \in \mathcal{S}} w_p M_{ip}^S M_{jp}^S. \quad (5.8)$$

With a reformulation of Q^S (Equation 5.6) we can derive the association term \mathbf{c} and describe the SKAT framework as a function of $Q(\mathbf{f})$ (Equation 5.1), as follows:

$$\begin{aligned} Q_{\mathbf{M}^S} &= (\mathbf{y} - \hat{\mathbf{y}})^\top \mathbf{K}^S (\mathbf{y} - \hat{\mathbf{y}}) = \mathbf{r}^\top \mathbf{K}^S \mathbf{r} \\ &= \sum_{i,j=1}^n r_i K_{ij}^S r_j = \sum_{p=1}^{|\mathcal{S}|} w_p \left(\sum_{i=1}^n M_{ip}^S r_i \right) \left(\sum_{j=1}^n M_{jp}^S r_j \right) \\ &= \sum_{p \in \mathcal{S}} w_p \left(\mathbf{M}^\top \mathbf{r} \right)_p^2 = \sum_{p \in \mathcal{S}} \mathbf{c}_p \\ &= \sum_{p=1}^m \mathbf{c}^\top \mathbf{f}^S = Q(\mathbf{f}). \end{aligned} \quad (5.9)$$

We want to find the indicator vector $\mathbf{f} \in \{0, 1\}^m$ that maximises the score $Q(\mathbf{f})$ while ensuring at the same time that the solution is (a) made of connected components of the network and (b) sparse. However, in general, it is difficult to find a subset of markers that satisfies these two constraints. In fact, given a positive integer k , the problem of finding a connected subgraph with k -vertices that maximise the sum of the weights on the vertices, which is equivalent to $Q(\mathbf{f})$ in our case, is known to be a strongly **NP**-complete problem [Lee and Dooly, 1996]. Therefore, this problem would generally be addressed based on enumeration-based algorithms, whose runtime grows exponentially with k . To cope with this problem, we consider an approach based on a graph-regularisation scheme, which allows us to drastically reduce the runtime.

5.1.2 Feature Selection with Graph Regularisation

SConES deals with a single phenotype \mathbf{y} , a set of genetic markers \mathbf{M} and a biological network described by the adjacency matrix \mathbf{A} . Our goal is to find the indicator vector $\mathbf{f} \in \{0, 1\}^m$ that maximises the association score $Q(\mathbf{f})$ while ensuring that the solution is made of connected components of the marker network. Rather than searching through all subgraphs of a given network, we reward the selection of adjacent genetic markers (features) through graph regularisation. This requirement can be addressed by means of a smoothness regulariser on the network [Ando and Zhang, 2007; Smola and Kondor, 2003]:

$$\arg \max_{\mathbf{f} \in \{0, 1\}^m} \underbrace{\mathbf{c}^\top \mathbf{f}}_{\text{Association Term}} - \underbrace{\lambda \mathbf{f}^\top \mathbf{L} \mathbf{f}}_{\text{Connectivity Term}}, \quad (5.10)$$

where \mathbf{L} is the Laplacian of the genetic marker network. The Laplacian \mathbf{L} is defined as:

$$\mathbf{L} = \mathbf{W} - \mathbf{A}, \quad (5.11)$$

where \mathbf{W} is a diagonal matrix and the p th diagonal element is the degree of node p . As $A_{pq} = 1$ if q is a neighbour of p (also written as $p \sim q$), and 0 otherwise, if we denote by $\mathfrak{N}(p)$ the neighbourhood of p , then the degree of p can be rewritten $W_{pp} = \sum_{q \in \mathfrak{N}(p)} 1$. The second term in Equation 5.10 can therefor be rewritten as:

$$\begin{aligned} \mathbf{f}^\top \mathbf{L} \mathbf{f} &= \sum_{p=1}^m \sum_{q=1}^m L_{pq} f_p f_q & (5.12) \\ &= \sum_{p=1}^m \sum_{q=1}^m (W_{pq} - A_{pq}) f_p f_q \\ &= \sum_{p=1}^m \sum_{q=1}^m W_{pq} f_p f_q - \sum_{p=1}^m \sum_{q=1}^m A_{pq} f_p f_q \\ &= \sum_{p=1}^m W_{pp} f_p^2 - \sum_{p=1}^m \sum_{q \in \mathfrak{N}(p)} f_p f_q \\ &= \sum_{p=1}^m \sum_{q \in \mathfrak{N}(p)} (f_p^2 - f_p f_q) \\ &= \sum_{p \sim q} [(f_p^2 - f_p f_q) + (f_q^2 - f_q f_p)] \\ &= \sum_{p \sim q} (f_p - f_q)^2. \end{aligned}$$

Thus the optimisation problem in Equation 5.10 is equivalent to:

$$\arg \max_{\mathbf{f} \in \{0, 1\}^m} \sum_{p=1}^m f_p c_p - \lambda \sum_{p \sim q} (f_p - f_q)^2. \quad (5.13)$$

As $(f_p - f_q)^2$ is 1 if $f_p \neq f_q$ and 0 otherwise, it can be seen that the connectivity term in Equation 5.10 penalises the selection of markers not connected to one another, as well as the selection of only subnetworks of connected components of the marker network. Note that it does not prohibit the selection of several disconnected subnetworks. In particular, solutions may include individual markers fully disconnected from the other selected markers. This is important since biological knowledge is incomplete and thus we do not want to enforce that solutions must form a single connected component. Furthermore, we would like to reward sparse solutions to avoid selecting large number of markers that are in LD and to avoid trivial solutions, such as selecting all or no markers. Selecting no markers is a sparse but useless solution. This second requirement can be enforced with a sparsity constraint. Thus, we can modify Equation 5.10 by adding an l_0 constraint:

$$\arg \max_{\mathbf{f} \in \{0,1\}^m} \underbrace{\mathbf{c}^\top \mathbf{f}}_{\text{Association Term}} - \underbrace{\lambda \mathbf{f}^\top \mathbf{L} \mathbf{f}}_{\text{Connectivity Term}} - \underbrace{\eta \|\mathbf{f}\|_0}_{\text{Sparsity Term}}. \quad (5.14)$$

Here, we directly minimise the number of nonzero entries in \mathbf{f} and do not require the proxy of an l_1 constraint to achieve sparsity (of course in the case of binary indicators, l_1 and l_0 norms are equivalent). The solution in Equation 5.14 is equivalent to:

$$\arg \max_{\mathbf{f} \in \{0,1\}^m} \sum_{p=1}^m f_p (c_p - \eta) - \lambda \sum_{p \sim q} (f_p - f_q)^2. \quad (5.15)$$

Also, as $\|\mathbf{f}\|_0 = \mathbf{1}_m^\top \mathbf{f}$, the sparsity term in Equation 5.14 is equivalent to reducing the individual association scores \mathbf{c} by a constant $\eta > 0$. Consequently, the positive regularisation parameters λ and η in Equation 5.14 control the importance of the connectedness of selected features and the sparsity of the solution, respectively.

5.1.3 Min-Cut Solution

Let us assume we are given an arbitrary graph $G^* = (V^*, E^*)$ that is described by its adjacency matrix \mathbf{A}^* . A cut over the vertices $V^* := \{1, \dots, m\}$ is defined as a partition of V^* in a nonempty set S and its complementary $V^* \setminus S^*$. A *s/t-cut* is defined as a cut such that $s \in S^*$ and $t \in V^* \setminus S^*$, where s and t in V^* are called the *source* and *sink* of the network. The *cut-set* of cut $C(S^*, V^* \setminus S^*)$ is a set of all pairs (u, v) for $u \in S^*$ and $v \in V^* \setminus S^*$ with positive weight A_{pq}^* . In other words, the *cut-set* of the cut $C(S^*, V^* \setminus S^*)$ is the set of edges E^* whose end vertices belong to different sets of the partition. The *minimum cut* of the graph is the cut such that the sum of the weights of the edges belonging to its cut-set is minimum. Finding the minimum cut of \mathbf{A}^* is equivalent to finding $S^* \subset V^*$ that minimises the *cut-function*:

$$\arg \min_{\mathbf{f} \in \{0,1\}^m} \sum_{p=1}^m \sum_{q=1}^m f_p (1 - f_q) A_{pq}^*, \quad (5.16)$$

where f_p is 1 if $p \in S^*$ and 0 otherwise. It is known from the max-flow-min-cut theorem [Elias et al., 1956; Ford and Fulkerson, 1956] that the minimum s/t -cut can be solved efficiently using the maximum-flow algorithm [Goldberg and Tarjan, 1988].

Solving the graph-regularised feature selection problem for graph G of adjacency matrix \mathbf{A} in Equation 5.14 is equivalent to finding a s/t min-cut on graph G . The vertices of graph G are augmented by two additional nodes s and t — representing the source and the sink — and whose edges are given by the adjacency matrix \mathbf{A}^* , where

$$\begin{aligned} A_{pq}^* &= \lambda A_{pq}, & \text{for } 1 \leq p, q \leq m & \text{ and} & (5.17) \\ A_{sp}^* &= \begin{cases} c_p - \eta & \text{if } c_p > \eta, \\ 0 & \text{otherwise,} \end{cases} & \text{and} \\ A_{tp}^* &= \begin{cases} \eta - c_p & \text{if } c_p < \eta, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The extended graph for the s/t min-cut formulation is illustrated in Figure 5.1. The

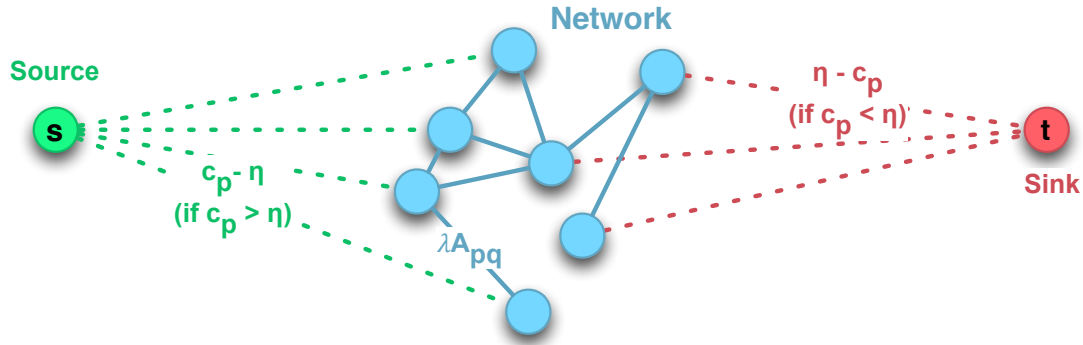


Figure 5.1: Extended graph for s/t min-cut. Graph is extended with a source s and sink t node. The source is connected with all nodes which association term $c_p > \eta$ and the sink is connected with all nodes which association term $c_p < \eta$.

problem in Equation 5.14 can be described as a s/t min-cut on the transformed graph defined by the adjacency matrix \mathbf{A}^* . For this purpose, we can rewrite the maximisation problem in Equation 5.14 as a minimisation problem:

$$\arg \min_{\mathbf{f} \in \{0,1\}^m} (\eta \mathbf{1}_m - \mathbf{c})^\top \mathbf{f} + \lambda \mathbf{f}^\top \mathbf{L} \mathbf{f}. \quad (5.18)$$

The first term of the objective can be encoded as a cut-function by adding two artificial nodes s and t :

$$\begin{aligned} (\eta \mathbf{1}_m - \mathbf{c})^\top \mathbf{f} &= \sum_{p=1}^m (\eta - c_p) f_p & (5.19) \\ &= \sum_{p \in S(c_p < \eta)} (\eta - c_p) + \sum_{p \in V(c_p \geq \eta)} (\eta - c_p) - \sum_{p \notin S(c_p \geq \eta)} (\eta - c_p) \\ &= \sum_{p=1}^m A_{sp}^* f_s (1 - f_p) + \sum_{p=1}^m A_{pt}^* f_p (1 - f_t) + \mathcal{C}, \end{aligned}$$

where $\mathcal{C} = \sum_{p \in V(c_p \leq \eta)} (\eta - c_p)$ is a constant, $f_s = 1$ and $f_t = 0$. The second term of the objective is a cut-function over the graph G :

$$\begin{aligned}
\mathbf{f}^\top \mathbf{L} \mathbf{f} &= \sum_{p=1}^m \sum_{q=1}^m L_{pq} f_p f_q & (5.20) \\
&= \sum_{p=1}^m \sum_{q=1}^m (W_{pq} - A_{pq}) f_p f_q \\
&= \sum_{p=1}^m f_p \left[\sum_{q=1}^m (W_{pq} - A_{pq}) f_q \right] \\
&= \sum_{p=1}^m f_p \left(\sum_{q=1}^m W_{pq} f_q - \sum_{q=1}^m A_{pq} f_q \right) \\
&= \sum_{p=1}^m f_p \left(W_{pp} - \sum_{q=1}^m A_{pq} f_q \right).
\end{aligned}$$

As by definition $W_{pp} = \sum_{q=1}^m A_{pq}$, we can write:

$$\begin{aligned}
\mathbf{f}^\top \mathbf{L} \mathbf{f} &= \sum_{p=1}^m f_p \left(W_{pp} - \sum_{q=1}^m A_{pq} f_q \right) & (5.21) \\
&= \sum_{p=1}^m f_p \left(\sum_{q=1}^m A_{pq} - \sum_{q=1}^m A_{pq} f_q \right) \\
&= \sum_{p=1}^m \sum_{q=1}^m f_p (1 - f_q) A_{pq}.
\end{aligned}$$

Consequently, the optimisation problem in Equation 5.14 can be solved by using a maximal flow algorithm. To efficiently optimise the objective function we use the Boykov-Kolmogorov algorithm [Boykov and Kolmogorov, 2004] that has a time complexity of $\mathcal{O}(m^2 m_E m_C)$, where m_E is the number of edges in graph G and m_C the size of the minimum cut. However, the performance is more efficient in practice, especially if the network is sparse.

5.1.4 Experimental Settings

Datasets

To assess the performance and the abilities of **SConES** to detect networks of phenotype associated genetic markers, we used *Arabidopsis thaliana* genotypes from Horton *et al.* [2012]. The dataset contains a total of 214,051 SNPs for 1,307 samples. The genotype data was used for simulations, real world experiments and for runtime comparisons. For the real world experiments we used 17 flowering time related phenotypes from Atwell *et al.* [2010]. For the networks protein-protein interaction data was downloaded

from The Arabidopsis Internet Resource (TAIR¹). If not explicitly stated we removed SNPs with a minor allele frequency lower than 10%, as typically done in *Arabidopsis thaliana* GWASs. To account for population structure in the real world experiments we conducted a Principle Component Analysis (PCA) on the genotype covariance matrix [Price *et al.*, 2006] and used the first x components as covariates in our model. The number of principle components x was chosen by adding them one by one to linear regression (logistic regression for binary phenotypes) until the genomic control value was close to one (see Equation 2.81 in Section 2.4.3).

Biological Networks

Our method SConES can handle any kind of biological networks between genetic markers. We here explored three special instances of biological networks. The first network simply connects all adjacent markers to a genomic sequence, also referred to as the Gene Sequence (*GS*) *network*. In this setting, we aim to recover sub-sequences of the genomic sequence that are associated with the phenotype (Figure 5.2a). The second

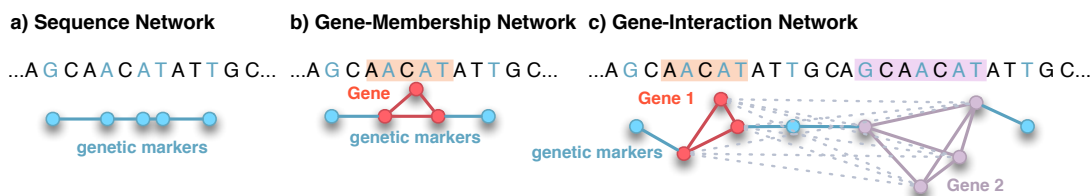


Figure 5.2: Three types of biological networks: a) Genomic sequence network: genetic markers adjacent on the genomic sequence are connected to each other. b) Gene membership network: adjacent markers and markers near the same gene are connected. c) Gene-interaction network: adjacent markers, markers near the same gene and markers near interacting genes are connected.

network is a Gene Membership (*GM*) *network*. Here, genetic markers are connected as in the *GS network* and in addition markers within or in close proximity to the same gene (here 20k bp window around the gene) are connected to each other (Figure 5.2b). The last network we investigated is a Gene Interaction (*GI*) *network*. For this network all genetic markers are connected to each other as described for the *GS* and *GM* network. In addition, genes that are interacting, e.g. in a biological pathway, are linked to each other.

Baseline and Comparison Methods

We compared SConES in all our experiments to a variety of baseline and state-of-the-art methods. As a baseline for comparison we ran a basic univariate linear regression (LR) to detect single SNPs that are significantly associated with a given phenotype. Similarly, we performed an association scan using a linear mixed model (LMM) to simultaneously correct for population stratification. These two methods only consider the effect of a single SNP on the phenotype. To also consider additive effects of SNPs we

¹<http://www.arabidopsis.org/portals/proteome/proteinInteract.jsp>

compared our method to a LASSO regression method. This model models the additive effect of all SNPs simultaneously with a sparsity constraint [Tibshirani, 1996]:

$$\arg \min_{\mathbf{f} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{M}\mathbf{f} - \mathbf{r}\|_2^2 + \underbrace{\eta \|\mathbf{f}\|_1}_{\text{sparsity constraint}}. \quad (5.22)$$

In addition, we compared to the network-constrained LASSO, also referred as to **ncLASSO** or **Grace** (graph-constrained estimation) [Li and Li, 2008, 2010]. Here, a graph-smoothing constraint is added in addition to the sparsity constraint:

$$\arg \min_{\mathbf{f} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{M}\mathbf{f} - \mathbf{r}\|_2^2 + \underbrace{\eta \|\mathbf{f}\|_1}_{\text{sparsity constraint}} + \underbrace{\lambda \mathbf{f}^\top \mathbf{L} \mathbf{f}}_{\text{connectivity constraint}}. \quad (5.23)$$

However, this approach has been developed with networks of genes rather than SNPs. The solution proposed by Li and Li [2008] requires to compute and store a single value decomposition of the Laplacian \mathbf{L} . This is not applicable for networks larger than $100k \times 100k$. However, we also compared to a more efficient solution by decomposing the Laplacian \mathbf{L} as the product of the network's incidence matrix.

Next, we also compared to the non-overlapping groupLASSO [Jacob et al., 2009], a sparse linear model designed to select features that belong to the union of a small number of predefined groups:

$$\arg \min_{\mathbf{f} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{M}\mathbf{f} - \mathbf{r}\|_2^2 + \lambda \sum_{g \in \mathcal{G}} \|\mathbf{f}^g\|_2, \quad (5.24)$$

where \mathcal{G} is the set of predefined groups of genetic markers that are possibly overlapping. If a graph over the features is given, defining those groups as all pairs of features connected by an edge or as all linear subgraphs of a given size yields the so-called graphLASSO. A similar approach is taken by [Huang et al., 2011]. The structured sparsity penalty used by their approach encourages selecting a small number of base blocks, where blocks are sets of features defined so as to match the structure of the problem. If a graph over the features is given, blocks can be defined as small connected components of that graph.

Implementation and Parameter Settings

We used Matlab to implement an initial version of SConES, as well as all the baseline and comparison methods. We used the SLEP library [Liu et al., 2009] for implementing the different regularised regression methods. To solve the maximal flow algorithm efficiently we used the Boykov-Kolmogorov algorithm² [Boykov and Kolmogorov, 2004]. All experiments are performed on a single core of an AMD Opteron CPU (2,048KB, 2,600MHz) with 512GB of memory, running Ubuntu 12.04.5 LTS.

Some methods have parameters that needed to be optimised. For this purpose, we ran a 10-fold cross-validation with an internal line-search for the optimisation of a single

²<http://vision.csd.uwo.ca/code/>

parameter or an internal grid-search in the case of two optimisation parameters. We chose seven values of λ and η , ranging from 10^{-3} to 10^3 . We then picked as optimal the parameters leading to the most stable selection of markers. Eventually, we reported the set of genetic markers (features) selected in all folds. We defined stability according to a consistency index similar to that of *Kuncheva* [2007]. The consistency index I_c between two feature sets \mathcal{S} and \mathcal{S}' is defined relative to the size of their overlap:

$$I_c(\mathcal{S}, \mathcal{S}') := \frac{\text{Observed}(|\mathcal{S} \cap \mathcal{S}'|) - \text{Expected}(|\mathcal{S} \cap \mathcal{S}'|)}{\text{Maximum}(|\mathcal{S} \cap \mathcal{S}'|) - \text{Expected}(|\mathcal{S} \cap \mathcal{S}'|)}, \quad (5.25)$$

where $\text{Maximum}(|\mathcal{S} \cap \mathcal{S}'|) = \min(|\mathcal{S}|, |\mathcal{S}'|)$ and $\text{Observed}(|\mathcal{S} \cap \mathcal{S}'|)$ is derived from the hypergeometric distribution as the expected probability of picking $|\mathcal{S}'|$ features out of x such that $|\mathcal{S} \cap \mathcal{S}'|$ are among the $|\mathcal{S}|$ features in \mathcal{S} :

$$P(|\mathcal{S} \cap \mathcal{S}'|) = \frac{\binom{|\mathcal{S}|}{|\mathcal{S} \cap \mathcal{S}'|} \binom{x-|\mathcal{S}|}{|\mathcal{S}'|-|\mathcal{S} \cap \mathcal{S}'|}}{\binom{x}{|\mathcal{S}'|}}, \quad (5.26)$$

and

$$\text{Expected}(|\mathcal{S} \cap \mathcal{S}'|) = \mathbb{E}(P(|\mathcal{S} \cap \mathcal{S}'|)) = \frac{|\mathcal{S}||\mathcal{S}'|}{x}. \quad (5.27)$$

Thus, we can write the consistency index I_c for two sets \mathcal{S} and \mathcal{S}' , as follows:

$$I_c(\mathcal{S}, \mathcal{S}') = \frac{x|\mathcal{S} \cap \mathcal{S}'| - |\mathcal{S}||\mathcal{S}'|}{x \min(|\mathcal{S}|, |\mathcal{S}'|) - |\mathcal{S}||\mathcal{S}'|}. \quad (5.28)$$

The consistency index I_{c_k} for an experiment with k -folds is defined as the average of the $k(k-1)/2$ pairwise consistencies between sets of selected features:

$$I_{c_k}(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k) = \frac{k(k-1)}{2} \sum_{i=1}^k \sum_{j=i+1}^k I_c(\mathcal{S}_i, \mathcal{S}_j). \quad (5.29)$$

5.1.5 Results

Runtime

First, we compared the runtime of SConES with that of our baseline and state-of-the-art comparison partners, that is linear regression, **ncLASSO** and non-overlapping **groupLASSO**. For this purpose, we selected subsets from 100 to 200,000 SNPs for 200 samples and generate exponential random networks with a density of 2% between SNPs. We reported the real CPU runtime of one cross-validation fold, examining the same number of parameters. We repeated each runtime experiment ten times and summarised the results in Figure 5.3. After three weeks, **graphLASSO** and **ncLASSO** had not finished running for a set of 50k SNPs. The accelerated version of **ncLASSO** ran out of memory for more than 150k SNPs. Further, we observed, that SConES is at least two orders of magnitude faster than **graphLASSO** and one order of magnitude faster than **ncLASSO**.

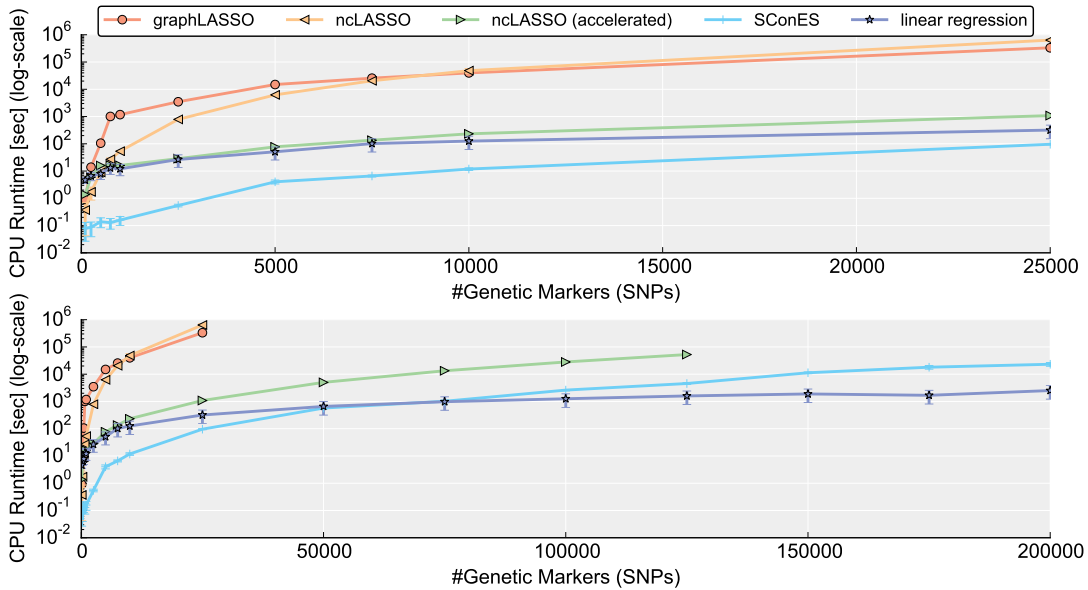


Figure 5.3: Runtime comparison between SConES, univariate linear regression, ncLASSO and graphLASSO: The left panel shows the runtime from 100 to 25k SNPs. The right panel shows the runtime up to 200k SNPs. After, three weeks, graphLASSO and ncLASSO had not finished running for 50k SNPs. The accelerated version of ncLASSO ran out of memory for more than 150k SNPs.

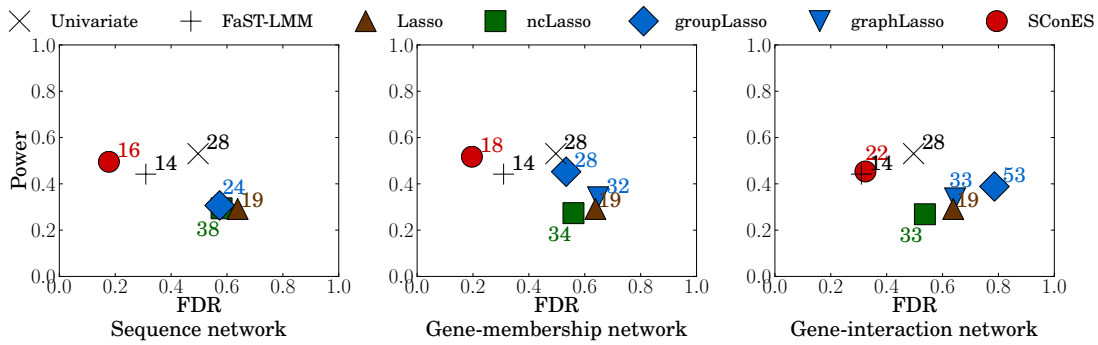
Simulations

We randomly selected 1,000 SNPs from the *Arabidopsis thaliana* genotype data from *Horton et al.* [2012] with a fixed sample size of 500 accessions. For our simulations, we generated artificial phenotypes for a subset of 20 markers that we deem to be causal for this phenotype. Phenotypes are simulated considering a linear additive model with normally distributed noise:

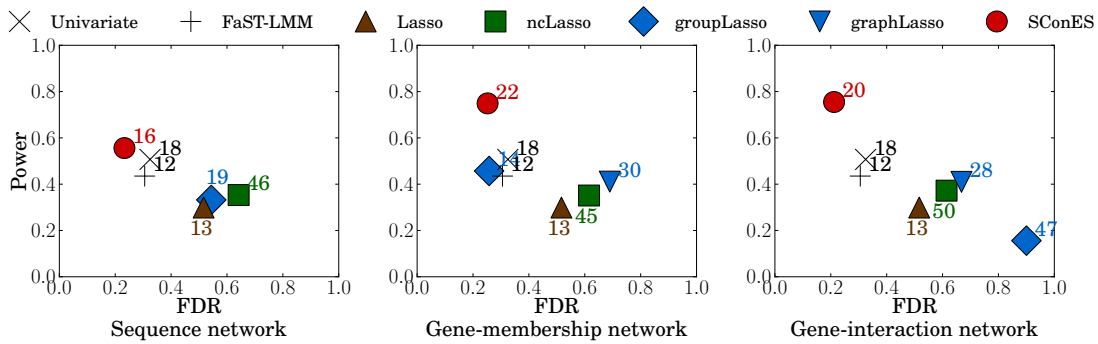
$$\mathbf{y}_c \sim \mathcal{N}(\mathbf{G}\boldsymbol{\beta}, \sigma^2 I), \quad (5.30)$$

where the weights $\boldsymbol{\beta} = (\beta_0, \beta_1)^\top \in \mathbb{R}^{n \times 21}$ are chosen to be normally distributed and $\mathbf{G} \in \mathbb{R}^{n \times 21}$ is a matrix containing the intercept (a vector of ones $\mathbf{1}$) and the subset of 20 causal genetic markers. For our simulations we considered the following scenarios, where the 20 causal markers are, (i) randomly distributed in the network, (ii) adjacent on the genomic sequence, (iii) near the same gene, (iv) near two interacting genes, (v) near three interacting genes or (vi) near five interacting genes. We repeated each experiment 30 times and compared the selected SNPs of either approach with the true causal ones in terms of power (fraction of causal markers selected) or False Discovery Rate (FDR, fraction of selected markers that are not causal). We summarised the results with F-scores (harmonic mean of power and one minus FDR) in Table 5.1. Further, we plotted the average FDR and power of our method and its comparison partners in Figure 5.4 for three out of the six scenarios. Since our method only returns a binary selection of features rather than a ranking of the features, it is not possible to draw receiver operating curves (ROC). For each scenario we investigated the impact of the different networks, *GS*, *GM* and *GI*. The closer the FDR/power point of an algorithm to the upper-left corner, the better this algorithm at maximising the power while at the same time minimising the FDR. Since regression methods tend to achieve better

(a) Scenario (ii): The true causal SNPs belong to the same genomic segment



(b) Scenario (iii): The true causal SNPs are near the same gene



(c) Scenario (vi): The true causal SNPs are near any of five interacting genes

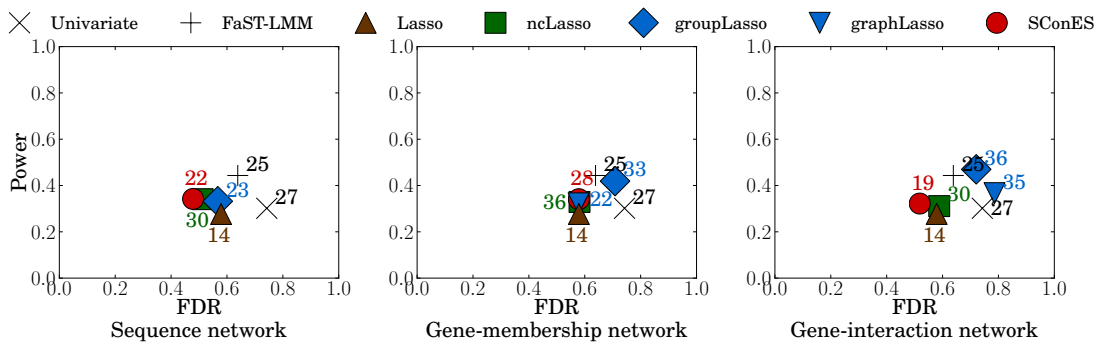


Figure 5.4: Evaluation of SConES on simulated data: Power and false discovery rate (FDR) of SConES, compared to state-of-the-art LASSO algorithms and a baseline univariate linear regression, in three different data simulation scenarios. Best methods are closest to the upper-left corner. Numbers denote the number of markers selected by the method.

Method		(i)	(ii)	(iii)	(iv)	(v)	(vi)
LR		0.26 ± 0.07	0.29 ± 0.12	0.28 ± 0.14	0.27 ± 0.07	0.26 ± 0.07	0.23 ± 0.08
LMM		<i>0.32 ± 0.01</i>	0.35 ± 0.01	0.33 ± 0.01	0.36 ± 0.02	0.38 ± 0.01	<i>0.33 ± 0.01</i>
LASSO		0.35 ± 0.01	0.32 ± 0.02	0.36 ± 0.01	0.36 ± 0.01	0.37 ± 0.01	0.32 ± 0.01
ncLASSO	<i>GS</i>	0.17 ± 0.01	0.25 ± 0.02	0.25 ± 0.01	0.45 ± 0.01	0.38 ± 0.02	0.30 ± 0.01
	<i>GM</i>	0.17 ± 0.01	0.26 ± 0.02	0.26 ± 0.02	0.38 ± 0.01	0.29 ± 0.01	0.27 ± 0.01
	<i>GI</i>	0.19 ± 0.01	0.26 ± 0.02	0.26 ± 0.02	0.43 ± 0.02	0.34 ± 0.02	0.28 ± 0.01
groupLASSO	<i>GS</i>	0.23 ± 0.01	0.30 ± 0.01	0.34 ± 0.01	0.37 ± 0.01	0.36 ± 0.02	0.32 ± 0.01
	<i>GM</i>	0.12 ± 0.00	0.44 ± 0.02	0.55 ± 0.01	<i>0.50 ± 0.01</i>	<i>0.40 ± 0.01</i>	<i>0.33 ± 0.01</i>
	<i>GI</i>	0.09 ± 0.00	0.26 ± 0.02	0.11 ± 0.01	0.54 ± 0.01	<i>0.40 ± 0.01</i>	0.34 ± 0.01
graphLASSO	<i>GS</i>	0.23 ± 0.01	0.30 ± 0.01	0.34 ± 0.01	0.37 ± 0.01	0.36 ± 0.02	0.32 ± 0.01
	<i>GM</i>	0.23 ± 0.01	0.28 ± 0.01	0.33 ± 0.01	0.36 ± 0.01	0.31 ± 0.01	0.31 ± 0.01
	<i>GI</i>	0.22 ± 0.01	0.28 ± 0.01	0.34 ± 0.01	0.33 ± 0.01	0.30 ± 0.01	0.27 ± 0.01
SConES	<i>GS</i>	0.21 ± 0.01	<i>0.55 ± 0.04</i>	0.57 ± 0.04	<i>0.50 ± 0.01</i>	0.43 ± 0.02	<i>0.33 ± 0.02</i>
	<i>GM</i>	0.19 ± 0.02	0.58 ± 0.03	<i>0.75 ± 0.03</i>	0.49 ± 0.01	<i>0.40 ± 0.02</i>	0.32 ± 0.02
	<i>GI</i>	0.20 ± 0.02	0.48 ± 0.03	0.78 ± 0.03	0.49 ± 0.01	0.39 ± 0.01	0.34 ± 0.02

Table 5.1: Comparison of tools: F-scores of SConES, compared to state-of-the-art LASSO algorithms and a baseline univariate linear regression (LR), in six different data simulation scenarios: The true causal SNPs are (i) randomly distributed; (ii) adjacent on the genomic sequence; (iii) near the same gene; (iv) near either of the same 2 connected genes; (v) near either of the same 3 connected genes; (vi) near either of the same 5 connected genes. Best performance in bold and second best in italics. *GS*: Genomic sequence network. *GM*: Gene membership network. *GI*: Gene interaction network.

power by selecting more features, we also reported the number of selected markers by each algorithm to show whether it remains reasonably close to the true value of 20 causal markers.

SConES was systematically better than its state-of-the-art comparison partners at leveraging structural information to retrieve the connected genetic markers that were causal. However, for the scenario (iv) our method was outperformed by groupLASSO. Note that groupLASSO is more sensitive to the definition of its groups than ncLASSO and SConES. Further, we investigated the effect of incomplete networks, that are networks with

Scenario	Fraction of Edges Removed				
	0%	2%	5%	10%	15%
(ii)	0.58 ± 0.03	0.58 ± 0.03	0.58 ± 0.03	0.57 ± 0.03	0.55 ± 0.03
(iii)	0.75 ± 0.03	0.75 ± 0.03	0.75 ± 0.03	0.75 ± 0.03	0.62 ± 0.03
(vi)	0.34 ± 0.02	0.34 ± 0.02	0.34 ± 0.02	0.33 ± 0.02	0.29 ± 0.02

Table 5.2: Effect of removing network edges: Effect on the F-scores of SConES of removing a small fraction of the network edges. Results reported for SConES + *GM* in three different scenarios: The true causal markers are (ii) adjacent on the genomic sequence; (iii) near the same gene; (vi) near either of the same five connected genes.

missing information. For this purpose, we randomly removed a range of 1% to 15% of the edges between the causal markers. Our results in Table 5.2 showed that this does not harm the performance of our method. This is an important feature since biological networks are likely to be incomplete. Nevertheless, the performance of SConES, as that of all other network-regularised approaches, was strongly negatively affected when the network is entirely inappropriate as in scenario (i). In addition, the decrease in performance from scenario (iii) to scenario (vi), when the number of integrating genes near which the causal markers are located increases from one to five, indicated that SConES, like its structured-regularised comparison partners, performed better when the causal

markers are less spread out in the network. Finally, **ncLASSO** was both slower and less performant than **SConES**. This indicates that solving the feature selection problem we pose directly, rather than its relaxed version, allowed for better recovery of true causal features.

Multi-locus Association Mapping in *Arabidopsis thaliana*

Next, we evaluated the performance of **SConES** on *Arabidopsis thaliana* genotype data and 17 flowering time phenotypes as described in Section 5.1.4. We excluded **graphLASSO** since it does not scale to datasets with more than 200k markers. While even our accelerated implementation of **ncLASSO** could not be run on datasets with more than 125k markers in our simulations, the networks derived for *Arabidopsis thaliana* are sparser than that used in the simulations, which made it possible to run **ncLASSO** on this data. For **groupLASSO** we used pairs of neighbouring markers, markers from the same gene or markers from interacting genes as predefined groups.

For many phenotypes the **LASSO** methods selected a large number of markers ($> 10k$), which made the results hard to interpret. Using cross-validated predictivity, as generally done for regression based models, instead of the consistency index didn't solve this issue entirely. Note that **SConES** directly maximises a score of association rather than minimising a prediction loss, as generally done for regression-based models. We filtered out solutions containing more than 1% of the total number of markers before using the consistency index I_c (Equation 5.29) to select the optimal parameters.

To evaluate the quality of the selected genetic markers $\mathbf{G} = (\mathbf{1}, \mathbf{g}_1, \dots, \mathbf{g}_p)$, where \mathbf{g}_i is the i th selected marker and p is the total number of selected markers, we conducted a 10-fold cross-validation for each flowering time phenotype \mathbf{y} . For each fold we trained a ridge regression on the training data (nine out of the ten subsets):

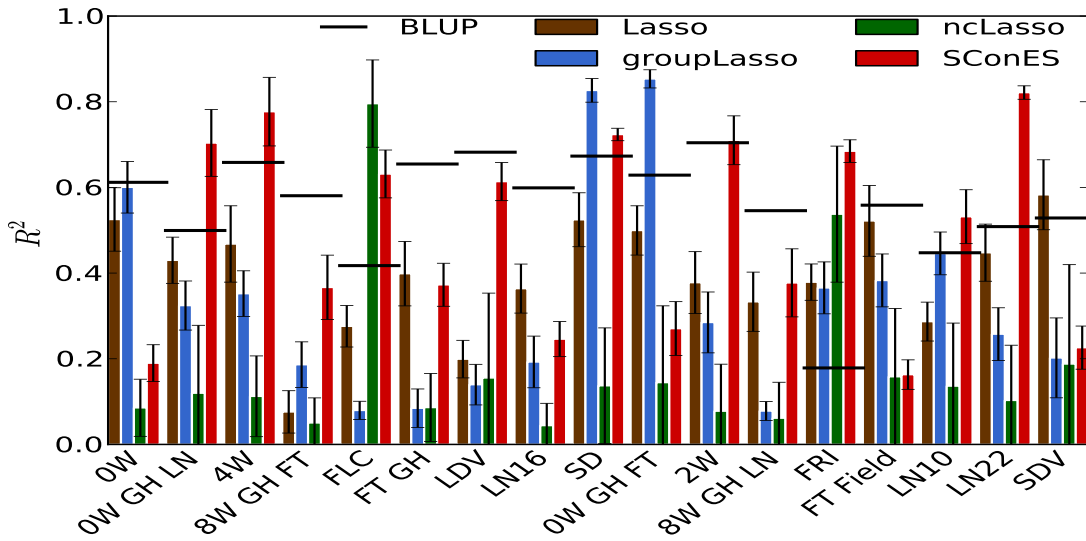
$$\arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{S}|}} \|\mathbf{y} - \mathbf{G}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2, \quad (5.31)$$

where $\boldsymbol{\beta}$ are the regression weights for the intercept and the selected markers and λ is a penalty parameter. Finally, we reported its average Pearson's squared correlation coefficient between the predicted phenotype $\hat{\mathbf{y}}$ and the true one of the left out evaluation set (Figure 5.5):

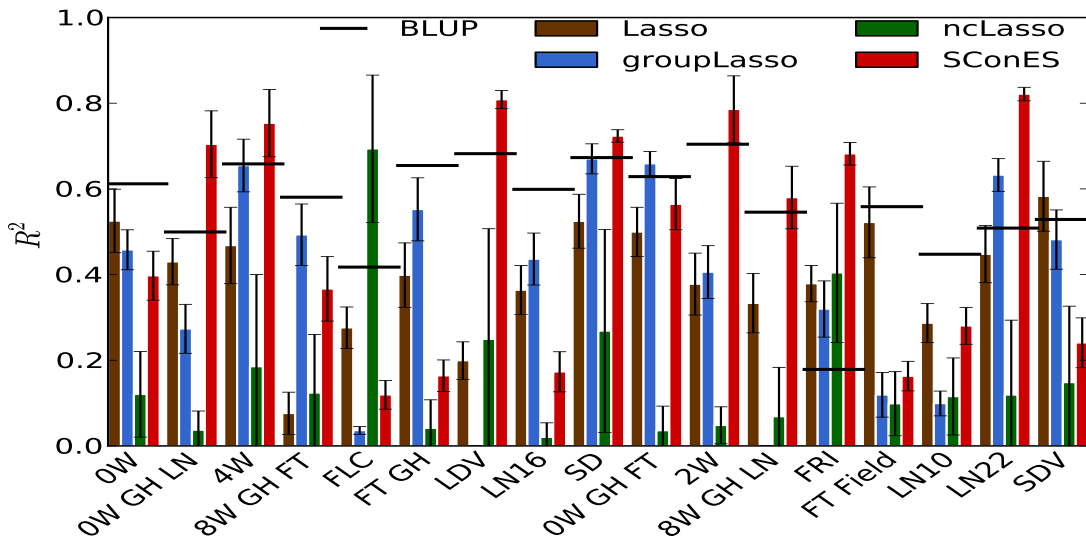
$$R^2 = \left(\frac{\text{Cov}(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}})}{\sigma_{\mathbf{y}_{\text{test}}} \sigma_{\hat{\mathbf{y}}}} \right)^2 = \left(\frac{\mathbb{E}[(\mathbf{y}_{\text{test}} - \mu_{\mathbf{y}_{\text{test}}})(\hat{\mathbf{y}} - \mu_{\hat{\mathbf{y}}})]}{\sigma_{\mathbf{y}_{\text{test}}} \sigma_{\hat{\mathbf{y}}}} \right)^2. \quad (5.32)$$

As additional baseline we reported the cross-validated predictivity of a standard Best Linear Unbiased Prediction (**BLUP**) [Henderson, 1975]. Although the features selected by **groupLASSO** + *GS* achieved higher predictivity than **SConES** + *GS* on most phenotypes, the features selected by **SConES** + *GM* were at least as predictive as those selected by **groupLASSO** + *GM* in two thirds of the phenotypes; the picture was the same for **SConES** + *GI*, whose selected markers were on average more predictive than those of **groupLASSO** + *GI*. The superiority of **groupLASSO** in that respect was to be

(a) Genomic sequence networks



(b) Gene membership network



(c) Gene interaction network

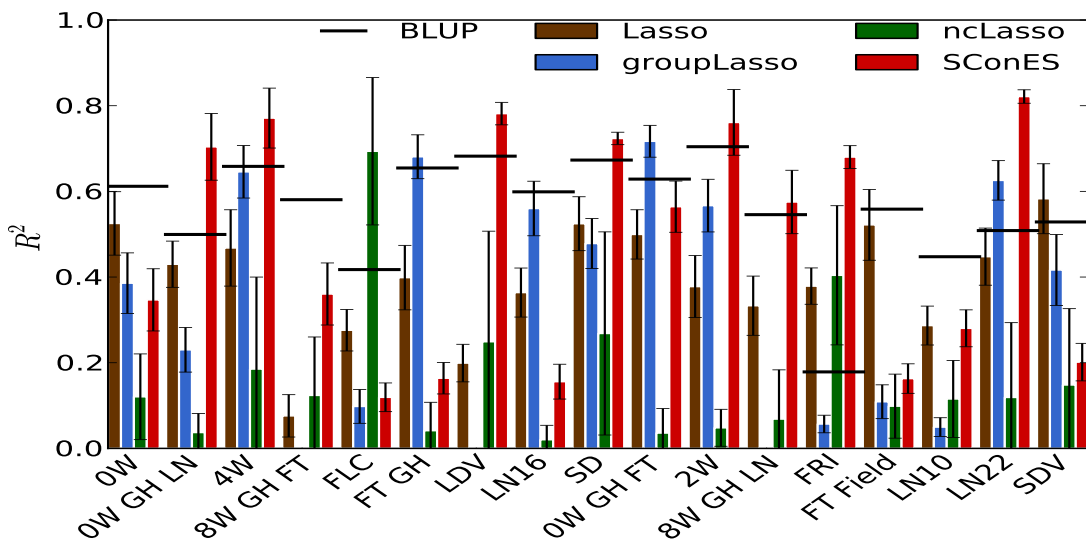


Figure 5.5: Cross-validated predictivity of SConES: Predictivity is measured as Pearson's squared correlation coefficient between the actual phenotype and the predicted phenotype by a ridge-regression over the selected markers, compared to that of LASSO, groupLASSO, and nLASSO. Horizontal bars indicate cross-validated BLUP predictivity.

expected, as predictivity is directly optimised by the regression. Also in 80% of the cases, if any of the feature selection methods achieved high predictivity ($R^2 > 0.6$), SConES outperformed all other methods including BLUP.

Next, we checked whether the selected markers from the three methods coincide with flowering time genes from the literature. We reported in Table 5.3 the number of markers selected by each of the methods and the proportion of these markers that were near flowering time candidate genes listed by *Segura et al.* [2012]. Here, the picture is re-

Phenotype	LR	LMM	LASSO	groupLASSO			ncLASSO			SConES		
				<i>GS</i>	<i>GM</i>	<i>GI</i>	<i>GS</i>	<i>GM</i>	<i>GI</i>	<i>GS</i>	<i>GM</i>	<i>GI</i>
0W	0/3	0/0	1/29	33/288	59/706	144/547	40/1077	14/318	14/318	123/271	0/85	0/69
0W GH LN	0/0	0/0	2/20	13/205	54/478	128/321	31/981	11/320	11/320	92/1251	92/1252	92/1253
4W	1/8	1/2	15/129	7/52	48/1489	80/436	2/238	6/298	6/298	104/1670	66/1078	42/859
8W GH FT	0/5	0/1	10/143	5/16	66/1470	0/0	14/427	11/398	11/398	26/322	26/322	26/319
FLC	0/1	0/1	1/31	2/95	0/101	0/214	4/135	1/35	1/35	115/1592	0/2	0/2
FT GH	0/1	2/10	7/46	8/106	90/841	177/1417	37/1434	42/1709	42/1709	0/626	0/59	0/59
LDV	0/4	1/2	10/80	8/32	0/0	0/0	14/437	7/177	7/177	39/674	86/1381	54/1091
LN16	0/5	0/0	9/222	0/95	138/957	89/1307	22/1094	33/1323	33/1323	73/73	0/3	0/4
SD	0/2	0/1	3/36	36/569	51/863	84/721	20/466	10/224	10/224	7/59	7/59	7/59
0W GH FT	0/9	1/3	20/194	49/654	52/898	241/1258	63/1597	84/1997	84/1997	0/6	29/317	29/317
2W	0/12	0/6	4/36	7/79	93/610	126/810	28/1006	43/1256	43/1256	76/756	78/1185	25/892
8W GH LN	0/2	0/3	8/122	13/168	0/0	0/0	19/493	21/501	21/501	11/73	75/776	68/757
FRI	6/11	5/9	6/18	8/64	8/20	10/10	2/9	2/4	2/4	101/1266	101/1271	101/1274
FRI Field	2/4	0/0	1/79	5/37	51/221	52/72	18/709	5/238	5/238	4/8	4/8	4/8
LN10	0/1	0/0	0/12	2/34	18/121	0/202	12/644	12/649	12/649	165/1921	0/91	0/91
LN22	2/14	0/0	6/65	0/12	33/894	81/1023	23/501	26/506	26/506	140/1378	140/1378	140/1378
SDV	0/5	0/1	4/208	3/94	1/721	105/936	14/379	15/384	15/384	53/454	0/8	0/8

Table 5.3: Associations close to known candidate genes: Associations detected close to known candidate genes, for all flowering time phenotypes of *Arabidopsis thaliana*. We report the number of selected markers near candidate genes, followed by the total number of selected markers. Largest ratio in bold. *GS*: Genomic sequence network. *GM*: Gene membership network. *GI*: Gene interaction network.

versed: SConES + *GS* and groupLASSO + *GI* retrieved the highest ratio of markers near candidate genes, whereas groupLASSO + *GS*, SConES + *GI* and SConES + *GM* showed lower ratios. At first sight, it seemed surprising that the methods with highest predictive power retrieved the least markers near candidate genes.

To further investigate this phenomenon, we recorded how many distinct flowering time candidate genes were retrieved on average by the various methods. A gene was considered retrieved if the method selected a marker near it (Table 5.4). Methods re-

Method	Network	#Markers	Near Candidate Genes	Candidate Genes Hit
LR		5	0.09	0.35
LMM		2	0.12	0.35
LASSO		86	0.09	3.82
groupLASSO	<i>GS</i>	153	0.10	4.35
groupLASSO	<i>GM</i>	611	0.09	1.35
groupLASSO	<i>GI</i>	546	0.20	2.65
ncLASSO	<i>GS</i>	684	0.04	4.88
ncLASSO	<i>GM</i>	608	0.06	4.59
ncLASSO	<i>GI</i>	608	0.06	4.59
SConES	<i>GS</i>	729	0.18	11.53
SConES	<i>GM</i>	546	0.08	14.82
SConES	<i>GI</i>	496	0.07	12.24

Table 5.4: Summary statistics: Averaged over the *Arabidopsis thaliana* flowering time phenotypes: average total number of selected markers (“#Markers”), average proportion of selected markers near candidate genes (“Near Candidate Genes”) and average number of different candidate genes recovered (“Candidate Genes Hit”). *GS*: Genomic sequence network. *GM*: Gene membership network. *GI*: Gene interaction network.

trieving a large fraction of markers near candidate genes did not necessarily retrieve

Phenotype	LMM	LR	LASSO	groupLASSO			SConES		
				<i>GS</i>	<i>GM</i>	<i>GI</i>	<i>GS</i>	<i>GM</i>	<i>GI</i>
4W	2	50%	0%	0%	0%	0%	50%	50%	50%
8W GH FT	1	0%	0%	0%	0%	0%	0%	0%	0%
FLC	1	0%	0%	0%	0%	0%	0%	0%	0%
FT GH	10	0%	0%	0%	0%	0%	0%	0%	0%
LDV	2	0%	0%	0%	0%	0%	0%	0%	0%
SD	1	100%	0%	0%	0%	0%	100%	100%	100%
0W GH FT	3	67%	0%	0%	0%	0%	0%	0%	0%
2W	6	33%	0%	0%	0%	0%	17%	33%	17%
8W GH LN	3	33%	0%	0%	0%	0%	0%	33%	33%
FRI	9	100%	100%	89%	56%	0%	100%	100%	100%
SDV	1	100%	0%	0%	0%	0%	0%	0%	0%

Table 5.5: Fraction of markers deemed significantly associated with the phenotype:

Comparison of markers identified by a LMM run on the full dataset to that selected by the other methods. We only report the phenotypes for which EMMAX returned at least one significant marker.

the largest number of distinct candidate genes. Good predictive power, as shown in Figure 5.5, however, seems to correlate with the number of distinct candidate genes selected by an algorithm, but with the percentage of selected markers near candidate genes. `groupLASSO + GI` had the highest fraction of candidate gene markers among all methods but detected only three distinct candidate genes. This was probably due to `groupLASSO` selecting entire genes or gene pairs; if `groupLASSO` detects a candidate gene, it will pick most of the markers near that gene, which led to its high candidate/-marker ratio in Table 5.3. We also compared the selected markers with those deemed significant by a LMM ran on the full data (Table 5.5). To summarise, `SConES` is able to select markers that are highly predictive of the phenotype. Among all methods, `SConES + GM` discovered the largest number of distinct genes whose involvement in flowering time is supported by the literature.

5.2 Multi-SConES

5.2.1 Multi-Task Formulation

To perform feature selection for multiple tasks (phenotypes) simultaneously, we can generalise the formulation of `SConES` as defined in Equation 5.14. For the multi-task approach, we assume that the set of vertices V (features/genetic markers) is shared all over T tasks (phenotypes). For each task t we have a network $G_t = (V, E_t)$ associated with a respective scoring function $Q_t(\mathbf{f})$ (Equation 5.1). Given such a set of T networks $\mathcal{G} = \{G_1, G_2, \dots, G_T\}$, the multi-task feature selection is formulated as:

$$\arg \max_{\mathbf{f} \in \{0,1\}^m} \sum_{t=1}^T \left(\underbrace{\mathbf{c}_t^\top \mathbf{f}_t}_{\text{Association Term}} - \underbrace{\lambda \mathbf{f}_t^\top \mathbf{L}_t \mathbf{f}_t}_{\text{Connectivity Term}} - \underbrace{\eta \|\mathbf{f}_t\|_0}_{\text{Sparsity Term}} \right) - \underbrace{\mu \sum_{t < t^*} \|\mathbf{f}_t - \mathbf{f}_{t^*}\|_2^2}_{\text{Task Penalty}}, \quad (5.33)$$

where $\mu \in \mathbb{R}^+$ is the newly introduced penalty parameter for the tasks. The penalty term represents our belief that similar networks should be associated with related features, and the larger the parameter μ , the more we enforce this belief. A large μ is thus better when it is desirable to select the same features across tasks.

Next, we will show that this problem can be reduced to a single-task feature selection similar to that of Equation 5.14 and thus can also benefit from the maximum flow algorithm. Let us assume an example with only two tasks ($T = 2$). The first step will be to replicate the vertices of each network G_t so that all sets of vertices are disjoint, that is, $G'_t = (V'_t, E'_t)$ such that $V'_t \cap V'_{t^*} = \emptyset$ for every $t, t^* \in \{1, \dots, T\}$ with $t \neq t^*$. All edges are copied on the replicated set V'_t and assume that vertices are indexed from 1 to m in each network G_t , where vertices have the same index if they are identical in the original set V . The i th vertex of a network G_t is denoted by v_t^i . We then construct a unified network $U(\mathcal{G}) = (\tilde{V}, \tilde{E})$ from the set of T networks $\mathcal{G} = \{G_1, \dots, G_T\}$ by connecting each pair of replicated vertices in the following manner:

$$\begin{aligned}\tilde{V} &:= \bigcup_{t=1}^T V'_t, \\ \tilde{E} &:= \bigcup_{t=1}^T E'_t \cup \bigcup_{i=1}^m F_i, \text{ where} \\ F_i &:= \{\{v_t^i, v_{t^*}^i\} | t, t^* \in \{1, \dots, T\}, t \neq t^*\}.\end{aligned}$$

The weight \tilde{w} of edges is given as $\tilde{w}(e) = w_t(e)$ if $e \in E'_t$ and $\tilde{w}(e) = \frac{\mu}{\lambda}$ otherwise. Thus, $U(\mathcal{G})$ has $|\tilde{V}| = Tm$ vertices and $|\tilde{E}| = \sum_{t=1}^T |E_t| + mT(T-1)/2$ edges. Note that graphs can be unified even if some features are missing for some tasks. In that case V'_t contains only vertices corresponding to the features available for task t and F contains only edges $\{v_t^i, v_{t^*}^i\}$, where the feature i is available for both tasks t and t^* . Figure 5.6 illustrates two scenarios of unified networks.

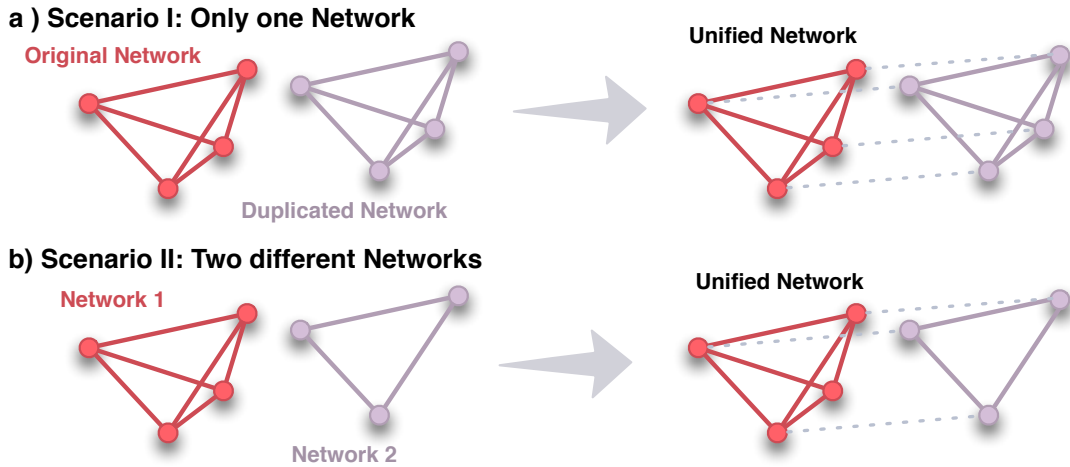


Figure 5.6: Two scenarios to generate an unified network: a) If only one network exists for the multi-task learning approach, than the original network is duplicated and new edges (dashed lines) between shared vertices are added. b) Two networks which share vertices but have different edges are given. The unified network contains new edges (dashed lines) between shared vertices.

Suppose that $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{c}}$ are concatenations of the vectors $\mathbf{f}_1, \dots, \mathbf{f}_T$ and $\mathbf{c}_1, \dots, \mathbf{c}_T$, re-

spectively. Then we directly can write:

$$\sum_{t=1}^T \left(\mathbf{c}_t^\top \mathbf{f}_t - \eta \|\mathbf{f}_t\|_0 \right) = \tilde{\mathbf{c}}^\top \tilde{\mathbf{f}} - \eta \|\tilde{\mathbf{f}}\|_0.$$

We describe the unified graph $U(\mathcal{G})$ by its adjacency matrix $\tilde{\mathbf{A}}$. The Laplacian is then defined as:

$$\tilde{\mathbf{L}} = \tilde{\mathbf{W}} - \tilde{\mathbf{A}},$$

where $\tilde{\mathbf{W}}$ is a diagonal matrix and the p th diagonal element is the degree of node p . Since newly introduced weights in the unified network are weighted as follows $\tilde{w}(e) = \frac{\mu}{\lambda}$ we can rewrite the connectivity and task penalty term as follows:

$$\sum_{t=1}^T \left(\lambda \mathbf{f}_t^\top \mathbf{L}_t \mathbf{f}_t \right) + \sum_{t < t^*} \mu \|\mathbf{f}_t - \mathbf{f}_{t^*}\|_2^2 = \lambda \tilde{\mathbf{f}}^\top \tilde{\mathbf{L}} \tilde{\mathbf{f}}.$$

Thus, we can rewrite Equation 5.33 as a single task feature selection problem:

$$\arg \max_{\tilde{\mathbf{f}} \in \{0,1\}^{Tm}} \tilde{\mathbf{c}}^\top \tilde{\mathbf{f}} - \lambda \tilde{\mathbf{f}}^\top \tilde{\mathbf{L}} \tilde{\mathbf{f}} - \eta \|\tilde{\mathbf{f}}\|_0. \quad (5.34)$$

5.2.2 Experimental Settings

Datasets & Networks

We used *Arabidopsis thaliana* data as described in Section 5.1.4. We derived a network of markers from TAIR and connected markers with a weight of 1 if they belong to the same gene or connected genes. Adjacent markers are connected with a small weight of 0.01.

In addition we generated synthetic data exactly as described in *Li and Li* [2008]. A total of 2,200 features composed of 200 transcription factors (TFs) and 2,000 genes are generated. Each TF is connected to ten regulatory target genes. That is, we have a network $G = (V, E)$ such that:

$$V = \bigcup_{i=1}^{200} \{w_i\} \cup G_i \text{ with } |G_i| = 10, \text{ and}$$

$$E = \bigcup_{i=1}^{200} E_i \text{ with } E_i = \{\{w_i, v\} | v \in G_i\},$$

which includes 200 connected subnetworks. Moreover, for each of the 200 TFs, an expression level x is generated from a standard normal distribution $\mathcal{N} = (0, 1)$. The expression levels for its regulatory genes follows $\mathcal{N}(0.7x, 0.51)$. Thus, the correlation between a TF and its regulatory genes is 0.7. Finally, we simulated a phenotype \mathbf{y} using the following model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \text{ where}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \text{ and } \sigma^2 = \sum_i^{100} \frac{\beta_i^2}{4}.$$

We generated four models with different feature weights β . In Model 1, β is:

$$\beta = \left(5, \frac{5}{\sqrt{10}}, \dots, \frac{5}{\sqrt{10}}, -5, \frac{-5}{\sqrt{10}}, \dots, \frac{-5}{\sqrt{10}}, 3, \frac{3}{\sqrt{10}}, \dots, \frac{3}{\sqrt{10}}, -3, \frac{-3}{\sqrt{10}}, \dots, \frac{-3}{\sqrt{10}}, 0, \dots, 0 \right)^\top.$$

The first four TFs and their 44 regulatory genes are causal to the response. Note that there is no edge between causal and non-causal features. Model 2 differs from Model 1 in that the signs of the first three target genes in each subnetwork are flipped to their opposites. Thus, this model describes a negatively correlated network. Model 3 and 4 are identical to the first two models, except that all $\sqrt{10}$ in β are replaced with 10, which leads to a weaker connection between TFs and genes. For each model, we generated training and test datasets of 100 samples each.

Baseline and Comparison Methods

We evaluated our method in both a single-task and a multi-task setting. In the single-task setting we compared to a standard feature selection algorithms, LASSO [Tibshirani, 1996], as well as four state-of-the-art structured regulariser methods, groupLASSO [Jacob et al., 2009], Elastic Net [Zou and Hastie, 2005], ncLASSO and ancLASSO (adaptive ncLASSO) [Li and Li, 2008, 2010]. ncLASSO and ancLASSO, which use a Laplacian graph regulariser, can be considered as LASSO analogous of SConES. Note that SConES is an association-based approach, whereas ncLASSO aims at minimising a prediction error in a LASSO framework. In addition, the adaptive version of ncLASSO (ancLASSO) allows that connected features have opposite effect directions.

For the multi-task version, we compared to different LASSO based multi-task versions. First we compared to Multi-Task LASSO [Obozinski et al., 2008]. This model uses a l_2 -norm on each weight across all tasks to reward features selected in all tasks. The Multi-Task LASSO solves the following optimisation problem for a given genotype matrix \mathbf{M} of size $n \times m$ and T phenotype (task) vectors \mathbf{y}_t :

$$\arg \min_{\mathbf{B} \in \mathbb{R}^{m \times T}} \frac{1}{2n} \sum_{t=1}^T \|\mathbf{y}_t - \mathbf{M}\beta_t\|_2^2 + \lambda \|\mathbf{B}\|_{l1/l2}, \quad (5.35)$$

where \mathbf{B} is a matrix of size $m \times T$ with all regression coefficients. The t th column of the matrix \mathbf{B} are the regression coefficients β_t for the phenotype (task) t . The $l1/l2$ -norm of \mathbf{B} is defined as:

$$\|\mathbf{B}\|_{l1/l2} = \sum_{i=1}^m \sqrt{\sum_{t=1}^T \beta_{t,i}^2} = \sum_{i=1}^m \|\beta_i\|_2. \quad (5.36)$$

Secondly, we also compared to a multi-task version of ncLASSO. For this purpose, we extended the single-task version to a multi-task version with a single network over the features. We therefore described the network by its Laplacian \mathbf{L} and formulated

Multi-ncLASSO of t phenotypes (tasks) as follows:

$$\arg \min_{\mathbf{B} \in \mathbb{R}^{m \times T}} \sum_{t=1}^T \left(\|\mathbf{y}_t - \mathbf{M}\boldsymbol{\beta}_t\|_2^2 + \lambda_1 \|\mathbf{B}\|_{l1/l2} + \lambda_2 \boldsymbol{\beta}_t^T \mathbf{L} \boldsymbol{\beta}_t \right), \quad (5.37)$$

where $\|\mathbf{B}\|_{l1/l2}$ is defined as in Equation 5.36, λ_1 is the penalty parameter for the tasks and λ_2 the network penalty parameter.

Implementation and Parameter Settings

All methods, including Multi-SConES were implemented in R, version 2.15.1. For LASSO, Elastic Net, and Multi-Task LASSO we used the `glmnet` package and for groupLASSO the `SGL` package. We conducted all experiments on a machine running Mac OS X version 10.7.4 with 2×3 GHz Quad-Core Intel Xeon CPUs and 16 GB of memory.

For all our experiments, we used the absolute value of Pearson's correlation coefficient between a feature v and the target vector \mathbf{y} to compute the association score \mathbf{c} . For the parameter selection we performed a 10-fold cross-validation and selected optimal parameters that yield the lowest Mean Squared Error (MSE). The main goal is to recover truly causal features, or in other words, to accurately classify the features into causal and non-causal. As this binary classification problem is imbalanced we evaluated the performance using Matthews correlation coefficient [Matthews, 1975] (see Appendix B).

5.2.3 Results

Runtime

First, we analysed the runtime of Multi-SConES with respect to the number of tasks. We created multi-task problems with varying number of tasks from 1 to 100 by repeatedly combining Model 1 with itself, and reported the runtime of Multi-Task LASSO, Multi-ncLASSO and Multi-SConES in Figure 5.7. Empirically, the runtime of

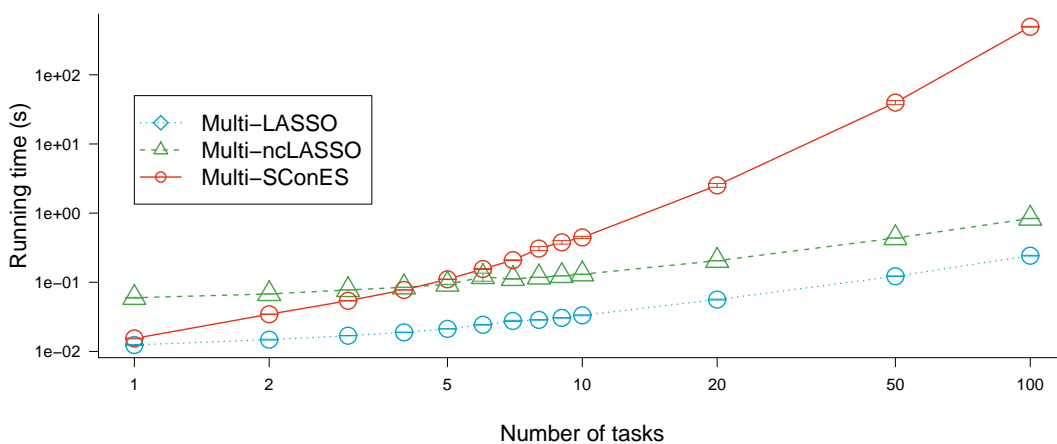


Figure 5.7: Runtime comparison: Runtime with respect to changes in number of tasks under fixed regularisation parameters.

Multi-SConES increases cubically with the number of tasks. While this is suboptimal, in particular compared with **Multi-ncLASSO**, we must remember that **Mutli-ncLASSO** cannot use different networks for different tasks. Moreover, **Multi-SConES** is still efficient enough to make it possible to analyse hundreds of thousands of features for the order of a dozen of tasks.

Simulations

Next, we analysed the behaviour of **Multi-SConES** with respect to changes in the regularisation parameters λ , η and μ . For that purpose, we fixed two of those parameters, and ran **Multi-SConES** for a two-task feature selection over Models 1 and 2. To understand parameter sensitivity with respect to the amount of causal features shared across tasks, we performed experiments for four cases: Models 1 and 2 share all, 3/4, half, or none of their features. We used $\lambda = 1$, $\mu = 1$, and $\eta = 0.2$ when they are fixed. Results are shown in Figure 5.8. **Multi-SConES** is sensitive to η , while more robust to

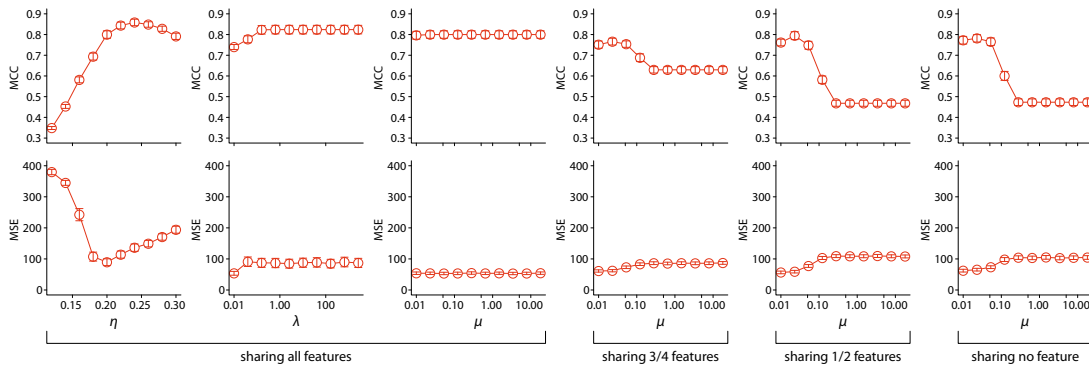


Figure 5.8: Parameter sensitivity: Feature selection performance with respect to changes in regularisation parameters η , λ and μ . Note that x-axes for λ and η have logarithmic scales. The effect of changes in μ are reported for various feature-sharing scenarios. Two parameters η and λ behave identically independently of the amount of true causal features shared by the tasks and corresponding plots are therefore not reported.

μ and robust to λ if it is set large enough. This can be explained as follows: once λ is large enough to cause the true causal features to be selected, if they form a subnetwork disconnected from the rest of the network, the corresponding penalty term becomes zero, and increasing λ will not affect the objective.

Similarly, if the true causal features are identical across all tasks, the penalty term controlled by μ is also zero, and varying μ will not affect the objective. However, if the causal features are not shared across all tasks, setting μ too large enforces the selection of too many identical features and leads to poor solutions. The behaviours of λ and η , however, remain unchanged across these different scenarios and is therefore not reported.

Next, we evaluated the feature selection performance of **Multi-SConES** in both single-task and multi-task settings. In the single-task setting **Multi-SConES** is equal to **SConES** since it is only using one phenotype. Note that we here used Pearson’s correlation coef-

ficient to compute an association score not SKAT. All results are illustrated in Figure

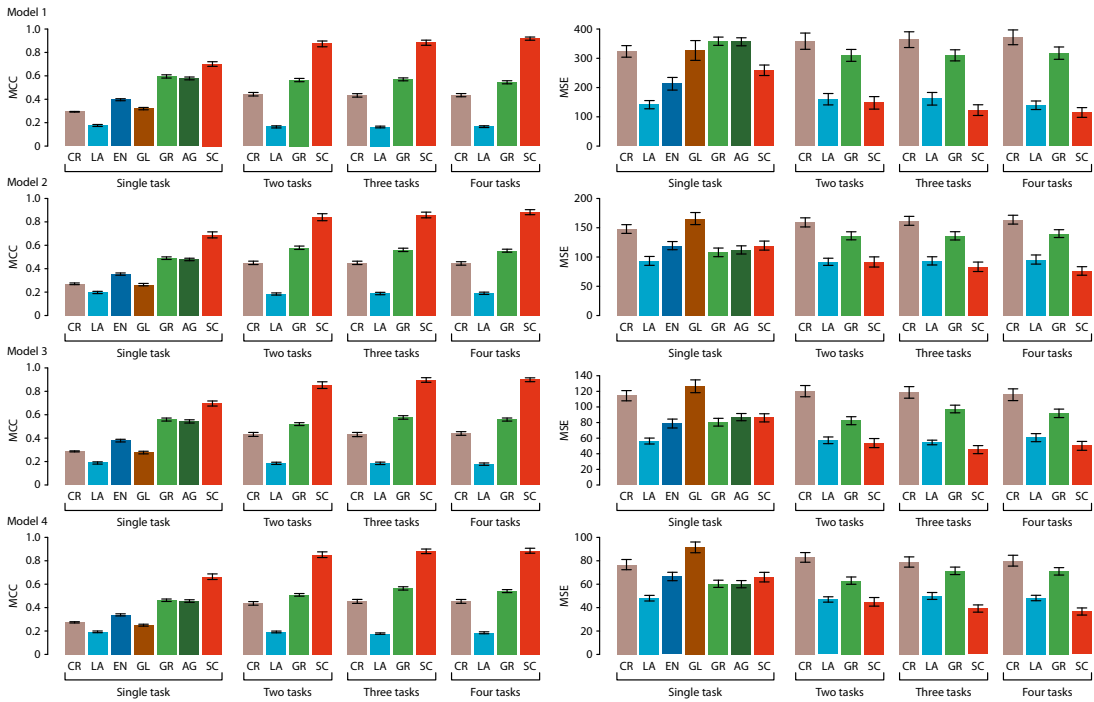


Figure 5.9: Feature selection performance for simulated data: MCC (left column) should be maximised and MSE (right column) should be minimised. CR: the ranking of correlations (baseline); LA: LASSO; EN: Elastic Net; GL: groupLASSO; GR: ncLASSO; AG: ancLASSO; SC: Multi-SConES

5.9. In the single-task setting, our method showed much better performance than the baseline Correlation Ranking. This supported the findings from our previous section, that our method works well to select connected features. Moreover, our method outperformed all the other methods in terms of MCC, showing that it is better in recovering true causal features. Only LASSO (and, in one case, Elastic Net) outperformed SConES in terms of predictivity of the selected features. However, the features it selects were too sparse and disconnected, resulting in notably worse MCC scores and difficulties in interpretability. These, results were consistent with the behaviour of SConES we showed in the latter section, although we used Pearson’s correlation coefficient as measure of association.

To evaluate Multi-SConES, we created multi-task problems by combining the models. More precisely, for Models 1 and 2, two-task problems were created by combining both models, and three-task problems by combining the with Model 3; for Models 3 and 4, two-task problems were created by combining both models, and three-task problems by combining them with Model 1. The four-task problem combines all four models. We observed that Multi-SConES outperformed the single-task version in all cases. Moreover, the performance (MCC and MSE) improved with the number of tasks. This confirmed that our multi-task formulation on feature networks is effective compared to solving each task independently. Furthermore, Multi-SConES achieved significantly better MCC than all of its comparison partners and was also superior in terms of predictivity. Our method was therefore effective for multi-task feature selec-

tion on networks.

Next we examined the ability of **Multi-SConES** in the case when not all causal features are shared among all tasks. For this purpose, we performed two-task feature selection (Models 1+2 and Models 3+4) by assuming that a fraction σ of the causal features are shared between the two tasks, where $\sigma = 3/4$, $\sigma = 1/2$ or $\sigma = 0$. We summarised all results in Figure 5.10. We observed that **Multi-SConES** clearly outperformed all

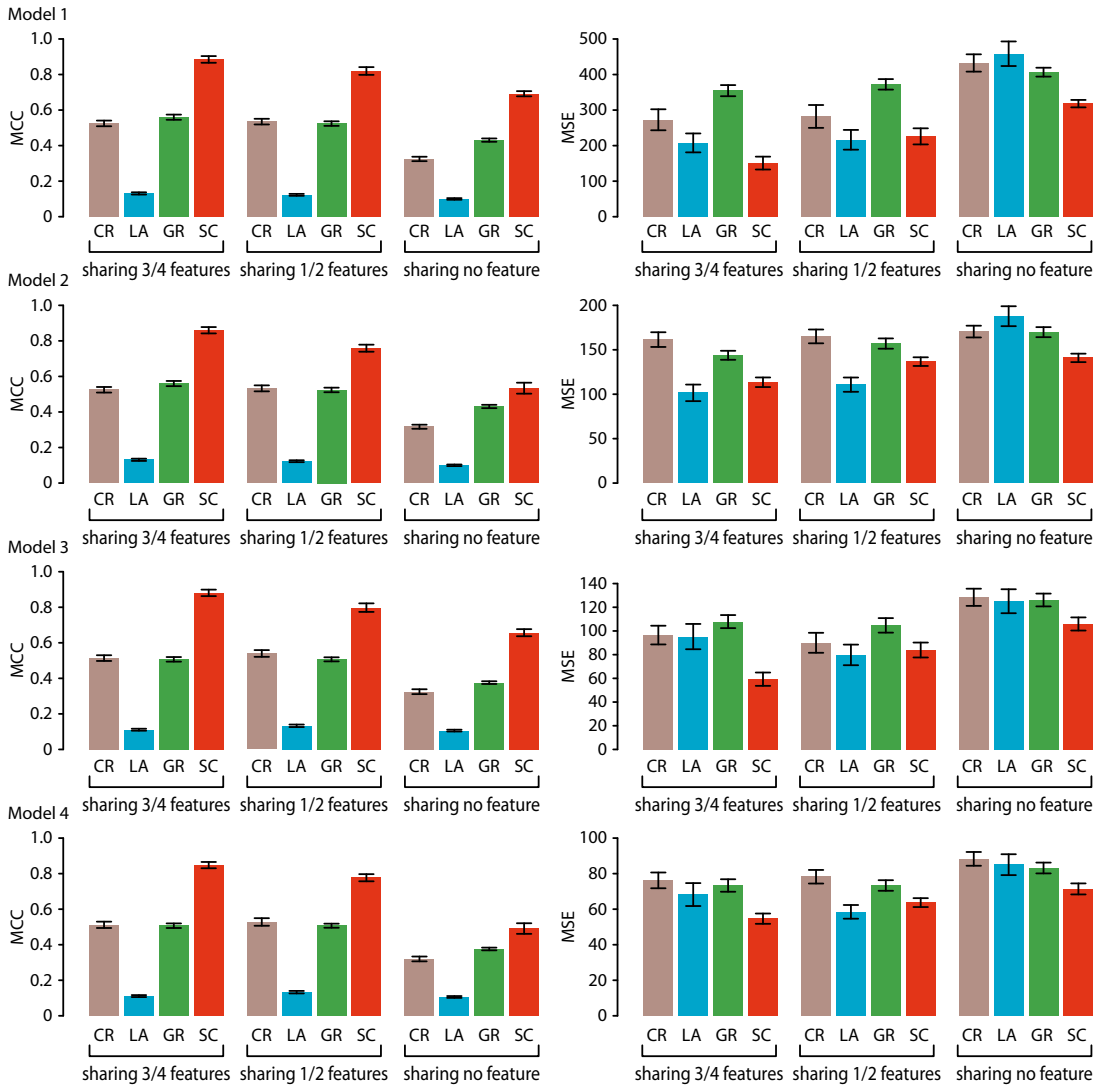


Figure 5.10: Feature selection performance in two tasks for simulated data: Only a fraction of the causal features are shared between the tasks. CR: the ranking of correlations (baseline); LA: LASSO; EN: Elastic Net; GL: groupLASSO; GR: ncLASSO; AG: ancLASSO; SC: Multi-SConES

other methods in terms of MCC. The features it selects were more predictive, with the only exception of **Multi-Task LASSO** when half of the features are shared between the tasks. Also, the more features are shared between the tasks, the better **Multi-SConES** was at recovering causal, explanatory features. This holds for all multi-task methods and is typical in multi-task learning.

Multi-locus & Multi-trait Association Mapping

Next we performed a multi-locus and multi-trait association mapping on *Arabidopsis thaliana* genotype data and flowering time related phenotypes. As described before, we used the first x principle components of the kinship matrix to correct for confounding due to population structure. Again, to evaluate the quality of the selected genetic markers, we used the 282 candidate genes for flowering time listed in *Brachi et al.* [2010] and *Atwell et al.* [2010], as an approximation of the gold standard. For each selected marker, we checked whether or not it was located within 20kb of one of the 282 candidate genes. If a marker belongs to more than two genes, we assigned it to the closest gene. We selected two flowering time phenotypes ($2W$ and LDV) which have on average high correlations to other related flowering time phenotypes. For each of these two phenotypes we picked two additional phenotypes that are highly correlated with ($2W$: $4W$ and $FT\ GH$; LDV : $0W$ and $FT10$). We then checked whether or not these additional phenotypes improved the performance of **Multi-SConES** and its competitors. For each method, we determined the optimal parameters by a 10-fold cross-validation and ran it on the full data to retrieve a final set of selected markers. We reported MCC, as well as the ratios of candidate markers (resp. genes) retrieved with respect to the number of selected markers (resp. genes) in Table 5.6. As in

Phenotypes	MCC			Hit ratio of markers			Hit ratio of genes		
	LASSO	ncLASSO	SConES	LASSO	ncLASSO	SConES	LASSO	ncLASSO	SConES
2W	0.001	-0.001	0.014	7/126	4/98	42/338	2/112	1/91	7/124
2W + 4W	-0.001	-0.003	0.016	7/175	6/198	81/802	2/163	2/191	11/240
2W + FT GH	0.001	0.000	0.024	9/173	7/146	106/818	9/162	7/135	13/250
2W + 4W + FT GH	0.005	0.002	0.027	15/183	16/265	101/679	6/174	3/256	13/208
LDV	0.001	0.000	0.016	6/116	7/144	73/667	2/107	2/131	9/202
LDV + 0W	0.005	0.007	0.020	16/196	19/206	86/702	2/183	2/187	10/209
LDV + FT10	0.001	0.001	0.021	12/214	10/191	92/762	1/199	1/181	10/221
LDV + 0W + FT10	0.003	0.002	0.023	18/283	19/323	81/482	2/265	1/307	10/153

Table 5.6: Results for multi-locus and multi-trait mapping: Results for different methods using between one and three correlated phenotypes. In case of more than one phenotype the multi-task version of the algorithm is used.

the simulations, **Multi-SConES** showed a superior performance in terms of MCC than its competitors. In addition, the proportion of markers near candidate genes among the selected markers was higher for **Multi-SConES** than those for the other methods. Finally, combining several phenotypes improved the detection of more causal candidate genes.

5.3 easyGWASCore Integration

Since the initial versions of **SConES** [Azencott et al., 2013] and **Multi-SConES** [Sugiyama et al., 2014] were implemented in **Matlab** and **R**, respectively, we re-implemented these methods using the **easyGWASCore** API in **C/C++**. In the following we compare the runtime between the initial versions and the **easyGWASCore** implementation. The runtime experiments are reported in seconds over a single core of an AMD Opteron CPU (2048 KB, 2600MHz) with 512GB of memory, running Ubuntu 12.04.5 LTS. We

used real genotype data from the 1001 genomes project in *Arabidopsis thaliana* and simulated continuous random phenotypes. We varied the number of genetic markers from ten thousand (10k) to one hundred thousand (100k), as well as the number of samples from 100 to 500. Data is stored and processed in PLINK format. For all these synthetic datasets we generated exponential random networks with edge densities between SNPs ranging from 0.1% (0.001) over 1% (0.01) to 5% (0.05). Further, we enforced that all adjacent genetic markers are connected within any network.

5.3.1 Data Processing and Algorithmic Runtime Analysis of SConES

First, we evaluated the easyGWASCore implementation of SConES with respect to data processing and real algorithmic runtime. For all experiments we fixed the connectivity parameter λ and the sparsity parameter η to 0.5. Results are illustrated in Figure

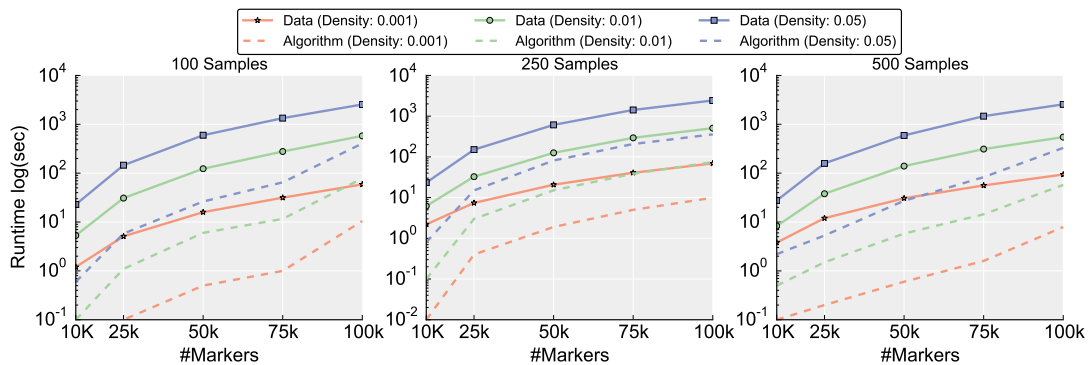


Figure 5.11: Data processing and algorithmic runtime evaluation: We varied the number of genetic markers, the number of samples and network densities. Solid lines represent data processing time, whereas dashed lines represent algorithmic runtime.

5.11. We observed that the time to load and process the data was always higher than the actual algorithm runtime. This is due to the fact, that the parameters λ and η were fixed and no grid-search was performed to find the optimal parameters. A runtime comparison including a full grid-search will be provided later. Further, we observed that the overall runtime increases with an increasing network density. This is true for both, data processing and algorithmic runtime. However, the runtime stayed approximately the same for an increasing number of samples.

5.3.2 Runtime Comparison Between Different Implementations

Next, we compared the original Matlab [Azencott et al., 2013] and R [Sugiyama et al., 2014] implementations for the single-task version of SConES to those from the easyGWASCore framework. Here, we only reported the actual CPU runtime. Again, the connectivity parameter λ and the sparsity parameter η were fixed to 0.5. Results are illustrated in Figure 5.12. We observed that the easyGWASCore implementation was approximately on par with the Matlab implementation. Both implementations are based on the same C++ maxflow optimisation algorithm and package by Boykov and Kolmogorov [2004]. The R implementation was between one and five orders of magnitude slower than the

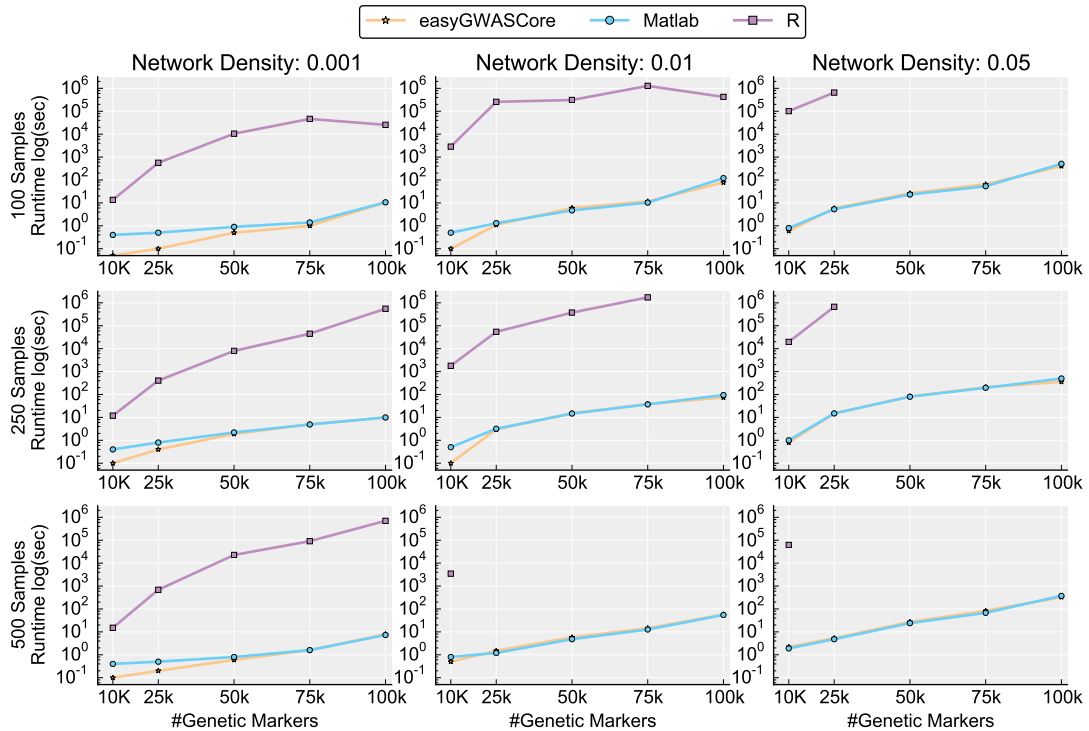


Figure 5.12: Runtime comparison between different implementations: Algorithmic runtime comparison with fixed parameters between different implementations of SConES. Experiments not finished after more than 20 days are truncated.

easyGWASCore and Matlab implementations. For most evaluations the R version even did not finish after more than 20 days. Again, we observed that an increased network density led to an extended runtime. Varying the number of samples did not affect the overall algorithmic runtime for these experiments.

5.3.3 Runtime Comparison of SConES Including a Grid-search

To evaluate the impact of different parameters λ and η we conducted a 5-fold cross-validations for five different λ and η values on a dataset of 500 samples and a network with a density of 0.1%. The runtime includes data processing time and the cross-validation time with an internal grid-search. Results are illustrated in Figure 5.13 for easyGWASCore, Matlab and R implementations. We observed that the Matlab imple-

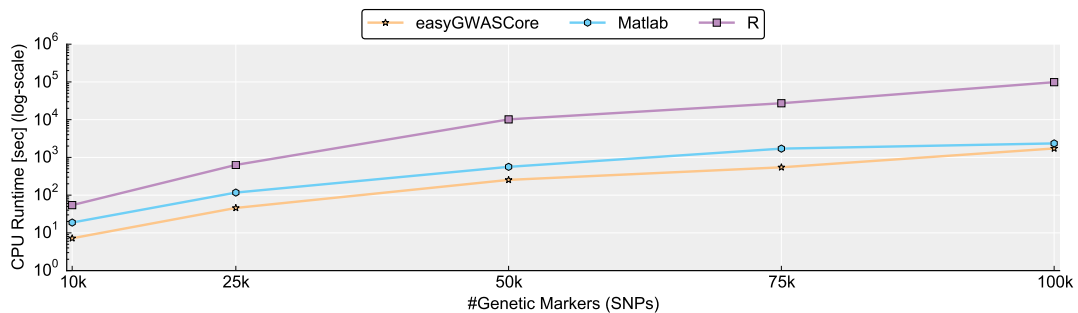


Figure 5.13: Runtime comparison of SConES including a grid-search: Comparison between the overall runtime of easyGWASCore and the original implementations in Matlab and R. Runtime includes data processing and the cross-validation with an internal grid-search.

mentation was on average half a magnitude slower than the `easyGWASCore` implementation. The R implementation, however, was on average two magnitudes slower than `easyGWASCore`.

5.3.4 Runtime Comparison of Multi-SConES

Finally, we compared the runtime of `Multi-SConES` between the `easyGWASCore` implementation and the original R implementation [Sugiyama *et al.*, 2014]. For this purpose, we varied the number of phenotypes (tasks) between one and four, as well as the number of SNPs (10k, 25k and 100k). We used a random network with a fixed density of 0.1%. For `Mutli-SConES` an additional parameter μ had to be optimised to adjust the

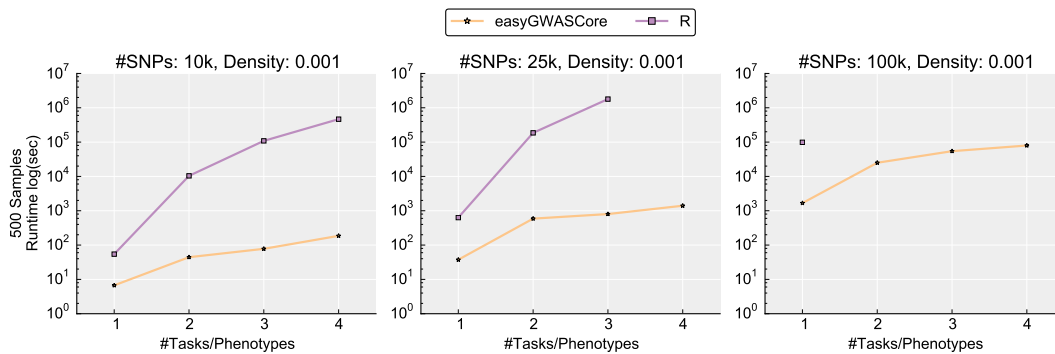


Figure 5.14: Runtime comparison of Mutli-SConES: Runtime comparison of `Mutli-SConES` between `easyGWASCore` and R. Number of tasks are varied between one and 4. Computations are truncated after 20 days of runtime.

importance between the different tasks. We ignored this parameter in the case of a single-task problem. We observed that the runtime increases rapidly with the number of tasks and SNPs (Figure 5.14). For 100k SNPs the R version could not finish the computations for more than one phenotype within 20 days.

5.4 Chapter Summary

In this chapter, we described two novel methods, `SConES` and `Multi-SConES`, to detect genetic markers associated with at least one phenotype that tend to be connected in a given biological network without restricting the search to predefined sets of loci. We showed that our solution can be solved efficiently by maximum flow and that it scales to whole genome-wide data. Further, `SConES` and `Multi-SConES` showed improved abilities to discover true causal features in simulated, as well as real-world data, compared to state-of-the-art structured regression-based approaches. In addition, loci selected by `SConES` tend to explain larger proportions of the phenotypic variance than univariate methods. Compared to other approaches that require a predefined set of groups, our method does not have this constraint. An advantage of our multi-task version is that `Multi-SConES` can use different networks for different tasks, and yields a clear, binary classification of features. Another attractive property of our approach is the possibility to incorporate cardinality constraints on the size of the solution set. However, for

now, we only consider an additive model between genetic markers and do not consider interaction effects, such as pairwise multiplicative effects. Also, how to incorporate linear mixed models to account for hidden types of population structure remains an interesting open question.

Eventually, we utilised the `easyGWASCore` API to integrate the `SConES` algorithms. We conducted a comprehensive runtime evaluation between different implementations and found that the `easyGWASCore` version is both efficient and scales to whole genome-wide settings even when including several correlated phenotypes.

The `easyGWASCore` framework and the `easyGWAS` web-application we developed two powerful and easy-to-use tools for performing, visualising, annotating and sharing GWASs and meta-analyses. In the next chapter we will apply these tools in a novel case-study to demonstrate its full potential to also perform non-standard GWASs and show that our framework can help to explain more of the missing heritability.

CHAPTER 6

Non-Additive Components of Genetic Variations in *Arabidopsis thaliana*

Throughout this thesis we developed an integrated framework, as well as methods for performing GWASs and demonstrated its abilities in several examples. For most experiments we used publicly available genotype and phenotype data from inbred lines from the model organism *Arabidopsis thaliana*. Inbreeding leads — in general — to harmful effects on reproductive capacity and general vigour of the mean phenotypic value [Falconer and Mackay, 1995]. This phenomenon is well known by biologists and breeders, and is often referred to as *inbreeding depression*. However, the effects of inbreeding do not apply to self-fertilising plants, since it is their normal mating system [Falconer and Mackay, 1995]. Wild-populations of *Arabidopsis thaliana* are predominantly self-fertilising species and thus highly homozygous, but outcrossing rates — that is the crossing with a different line — between 0.3% and 2.5% have been observed [Abbott and Gomes, 1989; Bakker et al., 2006; Bergelson et al., 1998; Bomblies et al., 2010; Pico et al., 2008]. To ensure homogeneity of wild collected individuals they are selfed in the lab for multiple generations before re-sequencing. This, as well as the small size of the genome, makes *Arabidopsis thaliana* an ideal model organism to study, especially because these isogenic lines have non-negligible advantages. For example the same line can be grown repeatedly in the lab under controlled environmental conditions. Thus, it becomes possible to phenotype and study the same line under multiple environments. The complementary phenomenon to inbreeding depression is called *heterosis*. The term heterosis was first described by George Shull [Shull, 1908, 1948] and is the phenotypic superiority of progeny or vigour of a hybrid cross relative to their genetically distinct parents [Baranwal et al., 2012; Falconer and Mackay, 1995]. Although, heterosis is an universal phenomenon and geneticists sought to dissect its genetic architecture, scientists still lack a comprehensive understanding of the genetic basis of hybrid phenotypes [Li et al., 2008]. In quantitative genetics the phenotypic value of a certain individual is defined as the contribution of a genetic and environmental component [Falconer and Mackay, 1995]:

$$P = G + E, \tag{6.1}$$

where P is the phenotypic value, G the genotypic value and E the environmental deviation. The sum of all these values in a whole population contributes to the population-wide variance, that is:

$$\sigma_P^2 = \sigma_G^2 + \sigma_E^2. \quad (6.2)$$

The genetic variance σ_G^2 can be divided in an additive genetic variance component σ_A^2 , the dominance genetic variance σ_D^2 and the interaction or epistatic genetic variance σ_I^2 :

$$\sigma_G^2 = \sigma_A^2 + \sigma_D^2 + \sigma_I^2. \quad (6.3)$$

If the entirety of genetic variation were the result of only additive genetic variance then first generation hybrids are expected to exhibit phenotypic values exactly between its two parents [Falconer and Mackay, 1995]. Since this expectation is frequently violated we know that hybrid superiority and inferiority result from non-additive genetic interactions [Falconer and Mackay, 1995]. As illustrated in Equation 6.3 non-additive genetic variance can arise from both dominance (inter-locus) variance σ_D^2 and epistatic genetic variance σ_I^2 . There are two main hypothesis to explain heterosis that received support from empirical studies over the last century: (i) the dominance hypothesis suggests that heterosis is the result of genome-wide complementation of weakly deleterious alleles. This hypothesis was first expressed in 1908 by Davenport [1908], Bruce [1910] and Jones [1917]. The (ii) hypothesis, referred to as the overdominance hypothesis, states that a single heterozygous locus can fully explain superior hybrid phenotypes and was described by Shull [1908] and East [1908]. Several examples can be found in the literature supporting the dominance [Li et al., 2008; Xiao et al., 1995] and the overdominance hypothesis [Li et al., 2001; Luo et al., 2001; Stuber et al., 1992].

In a collaborative study with the Weigel lab [Seymour et al., 2015] we investigated the effect of non-additive genetic variance on hybrid phenotypes in *Arabidopsis thaliana* and characterised the contribution of dominance to heterosis as a potential source of missing heritability. For this purpose, we utilised the `easyGWASCore` framework and API to characterise the underlying genetic architecture of hybrid superiority in 10 phenotypic traits. In the following we will give a detailed summary of the experimental settings and how the `easyGWASCore` framework was used for performing the GWASs, as well as the visualisations and annotations of results.

6.1 Data Generation and Preparation

6.1.1 Generation of Plant Material

For this study 30 parental inbred lines of *Arabidopsis thaliana* were chosen from publicly available genome sequences [Cao et al., 2011]. These accessions were chosen because they span much of the genetic diversity in continental Europe. These inbred lines, however, make it impossible to interrogate the contribution of allelic interactions or dominance to the total phenotypic variance, because only homozygous loci are available. For this purpose, hybrid crosses were generated between these 30 parental

inbreed lines. A half diallel (from the greek word *diallēlos*, from *dia*: through; *allēlōn*: one another) crossing scheme was used [Christie and Shattuck, 1992; Gilbert, 1958]. Diallel crossing schemes are used for parental genotypes that provide both, maternal and paternal gametes. A full diallel crossing generates a F_1 generation of hybrids where all possible crossing combinations between the parents are conducted (Figure 6.1a)). Thus, a total of p^2 hybrids can be generated, where p is the total number of parents. In

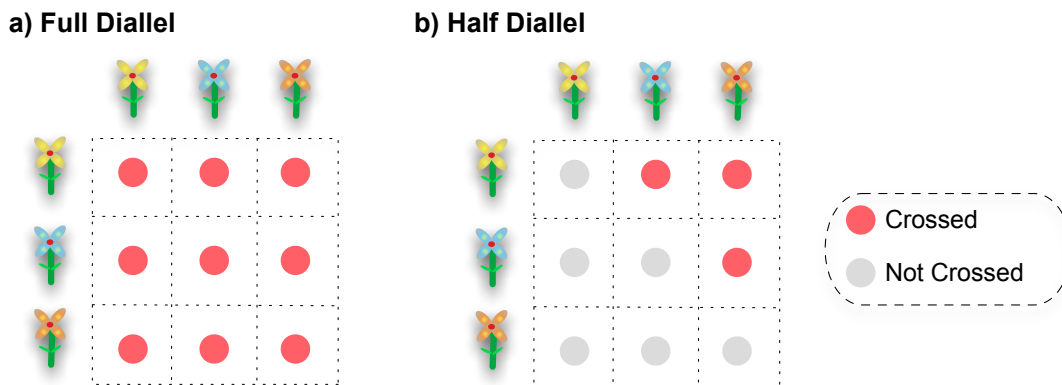


Figure 6.1: Illustration of diallel crossing schemes: a) In a full diallel crossing scheme, parents p are crossed in all possible combinations with each other. A total of p^2 crosses are possible. b) In a half diallel crossing scheme, parents are crossed excluding self-crosses and reciprocal crosses.

a half diallel cross, self-crosses between parents and reciprocal crosses are excluded, e.g. $\text{parent}_1 \times \text{parent}_2$ is allowed but not $\text{parent}_2 \times \text{parent}_1$ (Figure 6.1). The total number of possible hybrids in a half diallel is $\frac{p(p-1)}{2}$. Full diallels require more than twice as many crosses as half diallels. However, the inclusion of reciprocal crosses would also allow the investigation of maternal and paternal effects on hybrids.

Because, *Arabidopsis thaliana* is a self-fertilizing plant an artificial miRNA against the floral identity gene, APETALA3 (AP3), was used to knock-down the male floral organs. Note that a knock-down is not a gene knock-out. Here, AP3 is suppressed to prevent the formation of stamen in transgenic plants [Chae *et al.*, 2014]. Thus, it was not necessary to manually dissect the inflorescence during pollination. To ensure that the maternal environment of the hybrid genotypes was equivalent to the parental ones, manual self-crosses of the parental strains were included as well. In addition, parents from self-fertilisation were also included in the study to remove a potential bias resulting from the AP3 knock-down for subsequent experiments (Figure 6.2). Thus, a total of 435 hybrid crosses, 30 parents from self-crosses and 30 self-fertilised parents were grown in a completely randomised design with 5 replicates per genotype. Seeds were stratified in the dark at 4°C for 10 days. After stratification seeds were sown into soil pots in a completely randomised design. Flats were covered with humidity domes and placed into 16°C growth chambers under long day conditions (16 hours light and 8 hours dark). Humidity domes were removed after one week and pots were manually thinned to one plant per pot.

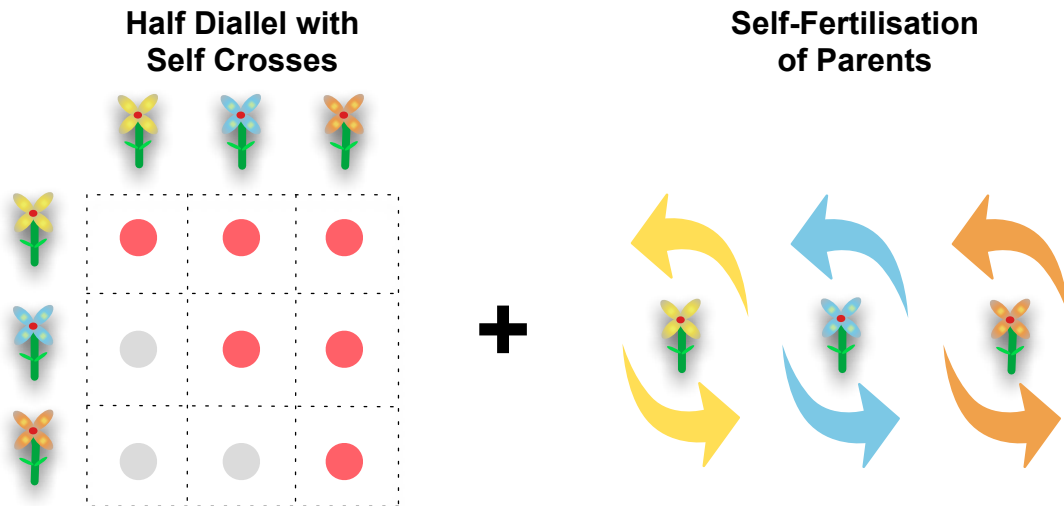


Figure 6.2: Illustration of experimental crossing scheme: Half diallel crosses (435 hybrids) including self-crosses between parents (30) and self-fertilisations of parents (30).

6.1.2 Plant Phenotyping

Ten different phenotypes were measured for all 495 plants and their 5 replicates. The phenotype DTF measures the Days To first Flowering and LTF the rosette leaf count at the first open flower. Once the plants had created approximately 10 siliques the plants were sacrificed. Then, the rosette diameter was measured and placed into paper bags and dried at 80°C for 24 hours. After the rosettes were completely desiccated their weight was measured and recorded. Additionally, images of each tray were taken at days 21 and 29. From these images the following measurements were extracted using a custom `imageJ` macro: area (day_{21} and day_{29}), perimeter (day_{21} and day_{29}), area growth ($\frac{day_{29}-day_{21}}{8}$), and perimeter growth ($\frac{day_{29}-day_{21}}{8}$).

6.1.3 F_1 Genotype Data Generation for GWASs

Since all parents are isogenic lines, *in silico* heterozygous F_1 genotypes could be easily generated for all crosses by combining known parental genotypes from the published whole genome re-sequencing data from *Cao et al.* [2011]. For this purpose, all 30 parental genotypes were filtered to i) remove all loci with missing information, ii) remove all loci that were not polymorphic in at least two of the 30 parental lines, iii) remove all tri-allelic SNPs with respect to the reference genome Col-0 and iv) remove all singletons. After this initial filtering step 723,403 SNPs remained. Extensive long-distance LD across chromosomes between loci could be observed, because of the limited number of parental genotypes. This small number of 30 parental genotypes led to a limited genetic diversity in the hybrid crosses. Positions in long distance LD across chromosomes and positions in LD with more than ten additional loci were excluded. For this purpose, all 723,403 SNPs were first encoded using the standard additive genotype encoding, where “0” is the major allele, “1” the heterozygous allele and “2” the minor allele. After encoding, only 75,346 SNPs were only be observed once within this population of hybrid crosses (from now on referred to as distinct SNPs). Next, we

created categories for how often a specific SNP pattern could be observed within this dataset (“Pattern occurrence”). These categories range from 2 to 7,364. For example, a SNP is located in “category 2” if this SNP shares the same pattern with exactly one other SNP in the genome and into “category 1,000” if the SNP in question shares the same pattern with exactly 999 other SNPs. In Figure 6.3 the cumulative number of SNPs for all categories is plotted. We observe that 32.97% of all our SNPs fall into the

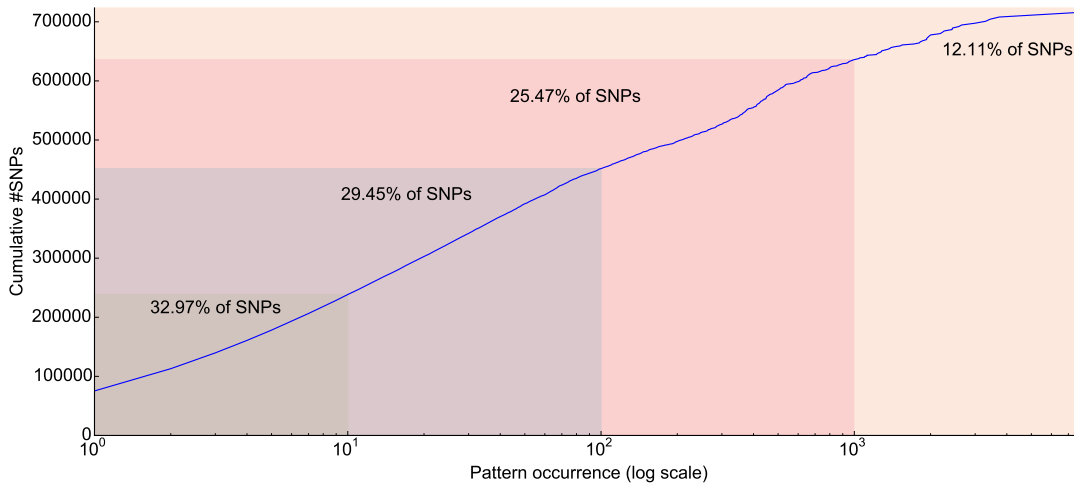


Figure 6.3: Cumulative distribution of pattern occurrences: Shows the number cumulative number of SNPs for different categories.

categories 1-10, which includes all distinct SNPs plus the number of SNPs for each of the categories from 2 to 10. Approximately, 29% of all SNPs fall into the categories 10-100 and nearly 38% into the categories 100 – 7,364.

We then evaluated if SNPs with shared patterns were located on the same chromosome or distributed across multiple chromosomes. Figure 6.4 shows the distribution of SNPs across chromosomes for categories 2-20. For the final SNP set, we allowed SNPs to share their pattern with up to 9 other positions (categories 1-10), however we removed all sites that exhibited long distance LD across chromosomes. The final data set consisted of 204,753 SNPs and these loci were used for all association mapping experiments.

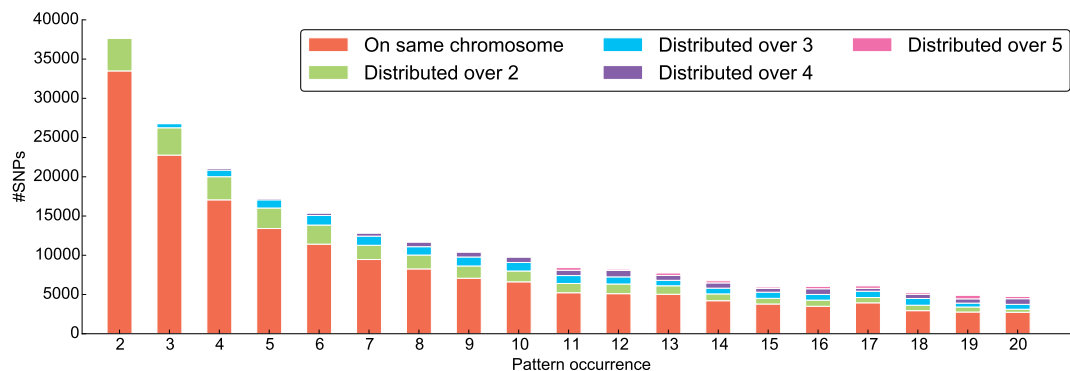


Figure 6.4: Distribution of SNPs within the first 20 categories: Distribution of SNPs across chromosomes for 20 different pattern occurrence categories.

6.1.4 Phenotype Data Preparation for GWASs

Seven hybrid crosses and two manually selfed parents (Bak-2 and ICE61) failed to germinate in all replicates and were excluded from further analyses from all experiments. Of the remaining lines only 2% of all plants failed to germinate and most germination failures only occurred in a single replicate. In these 58 cases, the missing phenotypes were imputed as the mean of the phenotyped replicates for each genotype. Eventually, all phenotypes were box-cox transformed to improve the normality of the data.

For GWASs we dissected the hybrid phenotypes into their additive phenotypic component a and their non-additive phenotypic component d (Figure 6.5). For this purpose, we first fitted for each phenotype a linear mixed model over all replicates:

$$y_{jkr} = \mu + G_{jkr} + A_{jkr} + \epsilon_{jkr}, \quad (6.4)$$

where G_{jkr} is the random genotypic effect of the j th and k th parent for the r th replicate. A_{jkr} is the random effect of the AP3 transgene on the hybrid cross of the j th and k th parent for the replicate r . Eventually, ϵ_{jkr} is the Gaussian distributed noise term of the model. For fitting the linear mixed model we used the `lme4`¹ R package. For each phenotype, the above model was fit with and without the transgene variable. The transgene effect was tested for statistical significance and subsequently removed from the model if it was not significant. The transgene was not significant for the phenotypes DTF, LTF and Dry Mass. After fitting the model the coefficients of each genotype (or in other words the estimated corrected mean phenotype) were extracted. We then used these estimates and calculated the standard quantitative genetic components of the phenotypes [Falconer and Mackay, 1995]. Typically, the additive phenotypic component a is calculated as half the distance between the two parental phenotypes, whereas the dominance deviation d , or the non-additive phenotypic component, refers to the discrepancy of an observed hybrid phenotype from its expected value, the mean phenotype of its two parental strains [Falconer and Mackay, 1995]. Thus, dominance deviation (d) was calculated as the distance of the hybrid pheno-

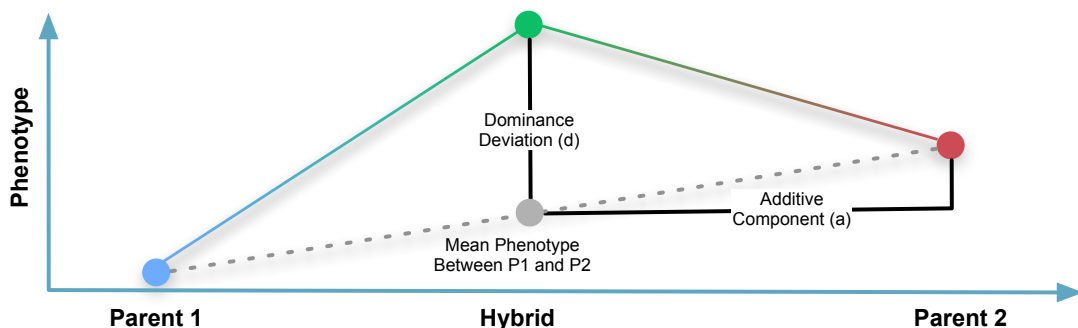


Figure 6.5: Illustration of phenotypic components: Dominance deviation of a hybrid phenotype is the distance of the hybrid phenotype from the mid-parent value. The additive phenotypic component a is calculated as half the distance between the two parental phenotypes.

¹<http://cran.r-project.org/web/packages/lme4/index.html>

type from the mid-parent value, or mean of the two parental genotypes (Figure 6.5). Because of the bidirectional discrepancy between self- and manually-fertilised parental genotypes, phenotypic means from manually crossed parents were used to estimate the dominance deviation d . Two of the thirty manually selfed parental genotypes did not germinate and as a result the dominance deviation could not be calculated for hybrids generated from these two parents (Bak-2 and ICE61). For the actual GWASs only 372 hybrid genotypes were used. For all phenotypes, d is approximately normally distributed with a mean close to zero (Figure 6.6).

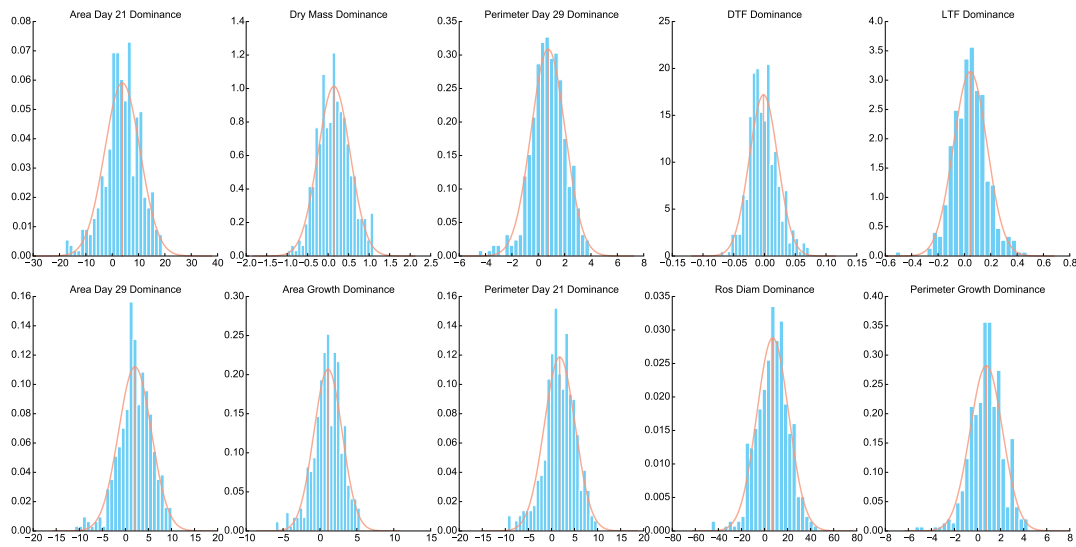


Figure 6.6: Dominance deviation histograms: Histograms show the distribution of the dominance deviation for all 10 phenotypes.

6.2 Methods and Experimental Settings

6.2.1 Heritability Estimation Based on Family Data

Diallel populations are cumbersome to construct but come with important advantages for variance component estimation [Lynch *et al.*, 1998]. With diallel designs it is possible to estimate the genetic combining ability (GCA), as well as the specific combining ability (SCA) of parents in its hybrid combination. The GCA is also called the breeding value and describes the average performance of each parent in its hybrid combination and is largely due to additive effects of genes. The SCA, however, describes the deviation from the expected average performance of two parents in their hybrid and is due to dominance or epistatic effects of genes [Sprague and Tatum, 1942]. The estimation of these variance components was done using all available transformed phenotype data from all hybrid crosses excluding self-crosses and self-fertilisation crosses. Because of low level of missing data (some lines didn't germinate) a linear mixed model was used which was implemented in SAS (code is based on the implementation by Fikret Isik²).

²<http://www4.ncsu.edu/~fisik/Analysis%20of%20Diallel%20Progeny%20Test%20with%20SAS.pdf>

The linear mixed model is defined as:

$$y_{ikr} = \mu + G_j + G_k + S_{jk} + \epsilon_{jkr}, \quad (6.5)$$

where, y_{ikr} is the r th replicate for the jk th cross, μ is the overall mean (or intercept), G_j and G_k are the random GCA of the j th and k th parents. S_{jk} is the random SCA of the j th and k th parent and ϵ_{jkr} is the Gaussian distributed error term. Equation 6.5 can be rewritten in matrix form as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \mathbf{H}\boldsymbol{\xi} + \boldsymbol{\epsilon}, \quad (6.6)$$

where $\boldsymbol{\beta}$ is a vector of fixed effect parameters. Since no fixed effects are included \mathbf{X} is a vector of ones to model the intercept. The parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ are the estimated random effect parameters (GCA and SCA, respectively). \mathbf{Z} is a design matrix of size $\#crosses \times \#parents$ and indicates which parents are used for which cross. The matrix \mathbf{H} is of size $\#crosses \times 1$ with the ids from all crosses.

Broad and narrow sense heritability estimates can be computed because additive and dominance genetic variance components can be derived from the estimated variance components σ_{GCA}^2 and σ_{SCA}^2 . The additive σ_a^2 , dominance σ_d^2 and the total phenotypic variance σ_p^2 for our data can be computed as follows [Lynch *et al.*, 1998]:

$$\sigma_a^2 = 2\sigma_{GCA}^2, \quad (6.7)$$

$$\sigma_d^2 = \sigma_{SCA}^2, \quad (6.8)$$

$$\sigma_p^2 = 2\sigma_{GCA}^2 + \sigma_{SCA}^2 + \sigma_\epsilon^2 = \sigma_a^2 + \sigma_d^2 + \sigma_\epsilon^2. \quad (6.9)$$

Thus, we easily can derive estimates for both, broad sense H_b^2 and narrow sense h_n^2 heritability [Lynch *et al.*, 1998]:

$$H_b^2 = \frac{\sigma_a^2 + \sigma_d^2}{\sigma_p^2}, \quad (6.10)$$

$$h_n^2 = \frac{\sigma_a^2}{\sigma_p^2}. \quad (6.11)$$

6.2.2 Genome-Wide Association Mapping

For GWASs we used the `easyGWASCore` framework and its Python command line interface. For the single-locus association mapping we used the `FaSTLMM` implementation to account for confounding due to population stratification and cryptic relatedness between the hybrids [Lippert *et al.*, 2011]. Here, we investigate two different genetic models, the *additive* genotype model and the *overdominant* genotype model. For this purpose, we employed the `FaSTLMM` algorithm to a standard additive SNP encoding (referred to as the “additive model”) and a non-standard overdominant SNP encoding (“overdominant model”), where both homozygous genotypic classes are encoded as “0” and the heterozygous genotype as “1”. For both models, the genetic similarity between

all *Arabidopsis thaliana* hybrids were estimated by computing the realised relationship kinship matrix [Hayes *et al.*, 2009] using the appropriate SNP encodings. For the additive model, genome-wide association mappings were performed on the estimated phenotypic values of the hybrids. For the overdominant model, both estimates were used, the phenotypic values of the hybrids, as well as the dominance deviation d of each strain.

To account for multiple hypothesis testing we used a conservative Bonferroni threshold of $\frac{0.05}{(\#\text{Tested SNPs: } 204,753)}$ ($p_v < 2.44e^{-7}$). Additionally, we performed an even more stringent correction by accounting for the total number of experiments per phenotype. Here, a total of three experiments per phenotype were performed, one for the additive model and two for the overdominant model which led to a corrected significance threshold of $\frac{0.05}{(\#\text{Tested SNPs: } 3 \times 204,753)}$ ($p_v < 8.14e^{-8}$).

In addition to the single-locus association mapping we searched for multi-locus associations of SNPs with the phenotype of interest. For this purpose, we used the networked guided algorithm SConES implemented in the `easyGWASCore` framework. For the network protein-protein interaction data was downloaded from TAIR³. For this analysis, we connected all adjacent SNPs to a genomic sequence and connected each SNP within a gene and in close proximity to the same gene (10k bp window around the gene). Further, we connected all SNPs to each other that lie in interacting genes of the protein-protein interaction network. To account for population structure we conducted a principle component analysis (PCA) on the genotype covariance matrix [Price *et al.*, 2006] and used the first x components as covariates in our model. The number of principle components x was chosen by adding them one by one to a linear regression until the genomic control value was close to one. We used the `--pc_iterative` command line argument to let `easyGWASCore` automatically determine the number of PCs for each phenotype.

6.2.3 GWAS Visualisations and Annotations

We used the plotting options from the `easyGWASCore` framework to generate Manhattan and QQ-plots for all experiments. The alternative, more stringent, Bonferroni threshold is added as a second dashed line to the Manhattan plots. In addition, we generated for all significantly associated hits linkage disequilibrium plots and enriched the plots with TAIR10 gene annotations and pathogenicity predictions. We used the tool SIFT4G⁴ to retrieve pathogenicity scores and predictions for all missense variants in our *Arabidopsis thaliana* hybrids.

³<http://www.arabidopsis.org/portals/proteome/proteinInteract.jsp>

⁴<http://sift-db.bii.a-star.edu.sg/AboutSIFT4G.html>

6.2.4 Estimation of Variance Explained

Variance Explained by all SNPs

First, we estimated how much of the phenotypic variance could be attributed to all SNPs jointly or to the total genetic contribution (random effect in a LMM) by using a cross-validation approach. Therefore, we generated 1,000 randomly drawn training sets (containing 90% of all hybrid genotypes) and testing sets (remaining 10% of hybrid genotypes). We then trained the FaSTLMM algorithm from `easyGWASCore` using only the kinship matrix (random effect) on the training data and subsequently predicted the phenotype $\hat{\mathbf{y}}$ of the remaining testing set. Predictions were obtained as described in Equation 4.4 from Chapter 4:

$$\hat{\mathbf{y}} = \mathbf{C}_{\text{test}}\hat{\boldsymbol{\beta}} + \mathbf{K}_{\text{test}} \left(\mathbf{K}_{\text{train}} + \hat{\delta}\mathbf{I} \right)^{-1} \left(\mathbf{y}_{\text{train}} - \mathbf{C}_{\text{train}}\hat{\boldsymbol{\beta}} \right),$$

where \mathbf{C} are the included covariates (or a vector of ones if no covariates are included), \mathbf{K} is the kinship matrix, and $\hat{\boldsymbol{\beta}}$ and $\hat{\delta}$ are the estimated parameters from the training step. The indices *train* and *test* indicate whether the data is coming from the training or testing subsets. Eventually, we computed variance explained as follows:

$$v(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}}) = 1 - \frac{\text{Var}(\mathbf{y}_{\text{test}} - \hat{\mathbf{y}})}{\text{Var}(\mathbf{y}_{\text{test}})},$$

where $\text{Var}()$ is the variance. Note that this measure might get negative and in such cases the phenotypic mean would provide a better fit than the actual trained model. Results were averaged across all 1,000 training sets.

Variance Explained by all Significant SNPs

Secondly, we estimated the variance explained by all significantly associated SNPs per phenotype. To estimate the variance explained by all significantly associated SNPs we trained a ridge regression on \mathbf{G} , where \mathbf{G} contains all significantly associated SNPs. A ridge regression has a penalty term to regularise the importance of different SNPs and thus implicitly takes into account the relatedness between individual SNPs:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I} \right)^{-1} \mathbf{G}^T \mathbf{y}, \quad (6.12)$$

where λ is the penalty term. λ is optimised by performing an internal line-search for a range of λ -values: $\lambda = \{1e^{-3}, 1e^{-2}, 1e^{-1}, 1, 1e^1, 1e^2, 1e^3\}$. Again, 1,000 cross-validation sets were run and averaged.

6.2.5 Power Analysis

We performed a simulation experiment to evaluate the power of the different encoding strategies. Here, we measured the power of each test with respect to the effect size, the minor allele frequency of the causal SNP, and the SNP encoding. We

varied the effect size between 0.05 and 0.80 (0.05,0.10,0.15,0.20,0.40,0.60,0.80) and binned the SNPs according to their minor allele frequency into the following bins $\{0.10 - 0.15, \dots, 0.45 - 0.50\}$. All experiments were performed with both the additive and overdominant SNP encoding. As the background covariance matrix (kinship matrix) we used the realised relationship matrix based on all SNPs, applying the appropriate encoding. For combination of factors (effect size, minor allele frequency, and SNP encoding), we first randomly chose a causal SNP with the selected minor allele frequency from our genotypic data. For these simulations, the SNP effect size is defined as [Park et al., 2010]:

$$e = 2\beta^2 f(1 - f), \quad (6.13)$$

where β is the regression coefficient and f is the minor allele frequency of the causal SNP. Thus, we can simulate a phenotype for different effect sizes as follows:

$$\mathbf{y} = \mathbf{G}\beta + \epsilon, \text{ where } \beta = \sqrt{\frac{e}{2f(1-f)}} \text{ and } \epsilon \sim \mathcal{N}(0, (1-e)\mathbf{I}), \quad (6.14)$$

with \mathbf{G} being the causal SNP. Each combination of factors (effect size, minor allele frequency, and SNP encoding) was repeated 1,000 times.

6.3 Results

6.3.1 Phenotypic Analysis Based on Family Data

First, we computed for all transformed phenotypes broad sense (Equation 6.10) and narrow sense heritability (Equation 6.11) estimates using the linear mixed model described in Equation 6.6 and plotted the results in Figure 6.7. The total genetic variances (H_b^2) ranged from 24% (Perimeter 21) to 78% (DTF) of the total phenotypic variance. Narrow sense heritability estimates ranged from 6% (Perimeter 21) to 48% (DTF). The large difference between broad sense (total genetic variance of the phenotype) and narrow sense (additive genetic variance the phenotype) heritability estimates suggests that the non-additive variance contributes significantly to the genetic variance of all measured phenotypes (between 18% and 32% difference). Furthermore, we observed

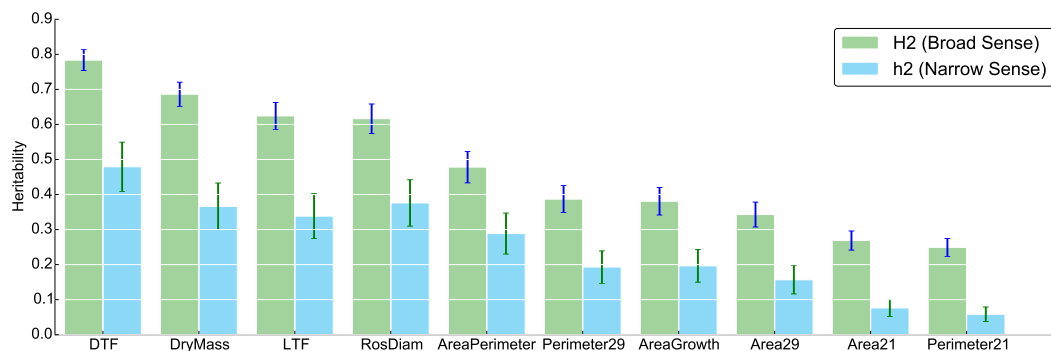


Figure 6.7: Heritability estimates: Broad sense H_b^2 and narrow sense h_n^2 heritability estimates including standard errors for all 10 phenotypes.

that young phenotypes at day 21 (Area21 and Perimeter21) showed lower heritability

estimates than older phenotypes at day 29 (Area29 and Perimeter29).

6.3.2 Association Mapping of Phenotypic Components

Next, we performed genome-wide association mappings for each of the ten phenotypes using the `easyGWASCore FaSTLMM` implementation. We not only performed association mappings on the estimated phenotypic means, but also on the non-additive dominance deviation d . By mapping the dominance deviation, or the discrepancy between the observed hybrid phenotype and the expected mid-parent value, we were able to remove potentially confounding additive effect which provided greater sensitivity to detect non-additive loci. For each phenotype we conducted three GWA mapping experiments. In the first experiment we simply fitted the estimated phenotypic means (estimated means of all measured replicates) using a standard additive model. Regardless of phenotype, no significantly associated SNPs could be detected after Bonferroni correction for multiple hypothesis. For the second experiment we used an overdominant SNP encoding and fitted the estimated phenotypic means using `FaSTLMM`. Here, we detected for the three phenotypes DTF (days to flower), LTF (leaves to flowering) and Area29 (rosette area extracted from the images of 29 day-old plants) a total of eight significantly associated hits. Manhattan plots and QQ-plots for these three phenotypes are shown in Figure 6.8. To account for multiple hypothesis testing two Bonferroni thresholds were computed. The red dashed line represents the standard Bonferroni threshold of $\frac{0.05}{(\#\text{Tested SNPs: } 204,753)}$ ($p_v < 2.44e^{-7}$), whereas the blue dashed line represents the more stringent Bonferroni threshold accounting for multiple testing across the three performed experiments per phenotype $\frac{0.05}{(\#\text{Tested SNPs: } 3 \times 204,753)}$ ($p_v < 8.14e^{-8}$). For the third experiment we again used the overdominant model but fitted the dominance deviation d of each phenotype to exclude potentially confounding additive effects from the model. Here, we detected far more associations (48 significant SNPs) than with the predicted trait means (8 significant SNPs). Significant hits were detected for the three phenotypes DTF (days to flower), LTF (leaves to flowering) and DryMass (dry mass of rosette). Manhattan plots and QQ-plots for these three phenotypes are illustrated in Figure 6.9. Significant hits found by the phenotypes DTF and LTF were the same when mapping the mean phenotypes using the overdominant model. From these 48 significantly associated SNPs, 34 are distributed across 15 different genes located on four different chromosomes. In addition, 14 associated loci were shared at least with one of these three phenotypes (DTF dominance, LTF dominance and DryMass dominance). Two linkage disequilibrium plots for significant SNPs associated with the dominance deviation d of the phenotype LTF are shown in Figure 6.10. As we can see in Figure 6.10a, two SNPs in close LD are significantly associated with the dominance deviation d of the phenotype LTF. Both SNPs are located in the gene AT1G14250 that is a GDA1/CD39 nucleoside phosphate family protein. The SNP at position 4,869,029 has a predicted deleterious effect on this protein. In addition, these two SNPs are also found to be significantly associated with the dominance deviation d of the DTF phenotype. The LD plot illustrated in Figure 6.10b shows seven highly associated SNPs

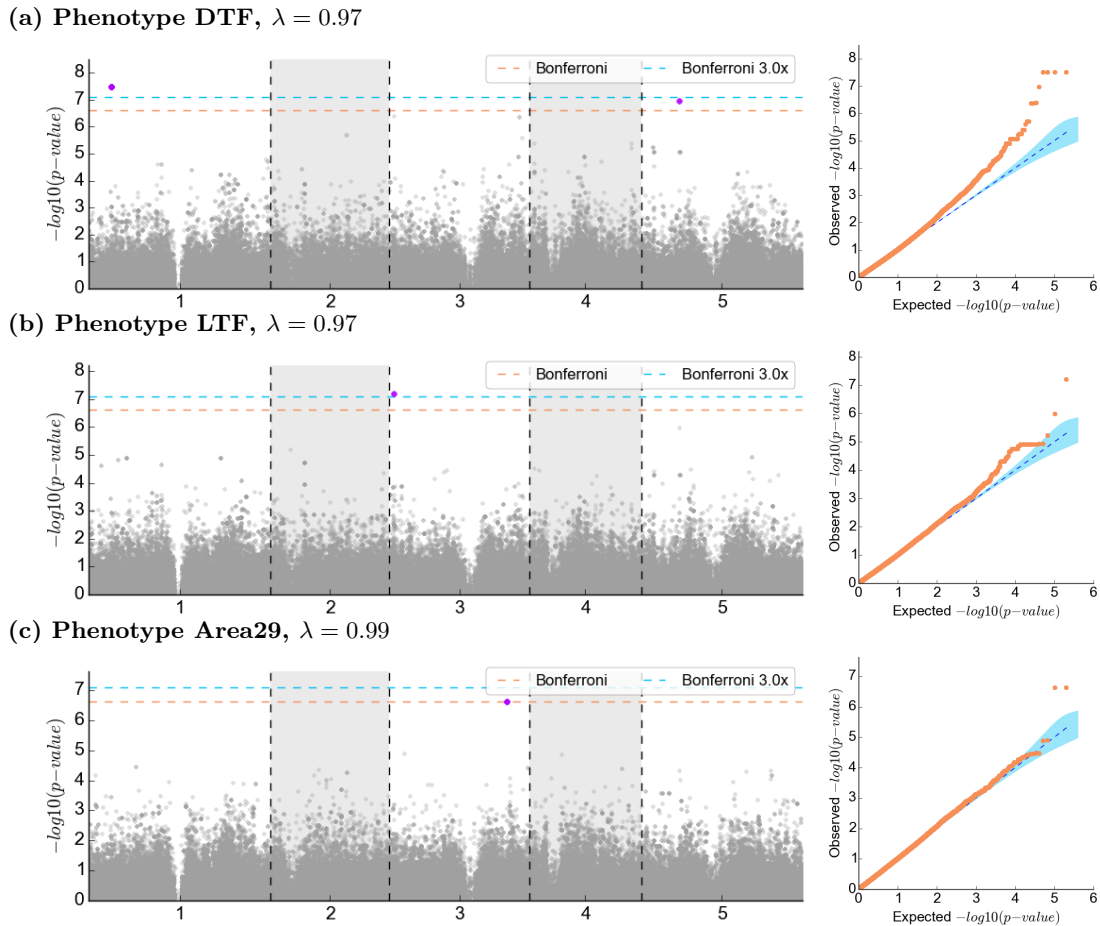


Figure 6.8: Manhattan plot and QQ-plot for estimated mean phenotypes (overdominant model): Manhattan and QQ-plots for three phenotypes with significantly associated SNPs when using an overdominant model and the estimated mean phenotypes. Two Bonferroni thresholds are shown. The standard one using all SNPs (red dashed line) and the stringent one (blue dashed line) correcting for 3 experiments (additive, overdominant mean and overdominant dominance deviation). Magenta points are significantly associated SNPs.

located in the gene AT2G13540 and these SNPs are only found to be significant for the LTF phenotype. The gene is also called ABA HYPERSENSITIVE 1 (ABH1) and encodes a nuclear cap-binding protein and is involved in flowering and abscisic acid (ABA) signalling.

All three phenotypes DTF, LTF and DryMass have high broad sense heritability estimates (Figure 6.7). Lower heritability phenotypes, such as Area21 and Perimeter21 (rosette phenotypes of young plants extracted from images), showed no significant associations with any position in the genome. These results suggest, that either dominance or overdominance, contributes significantly to non-additive genetic variance.

In addition to univariate association tests, we used **SConES** to detect multiple loci associated with each phenotype of interesting guided by a *Arabidopsis thaliana* protein-protein interaction network. We used an overdominant SNP encoding and the dominance deviations d as phenotypes. Principle components were selected and used as covariates to account for population stratification. Using this strategy we were able to detect between 10 and 324 associated loci per phenotype.

The historical hypothesis of heterosis — the dominance and overdominance hypothesis

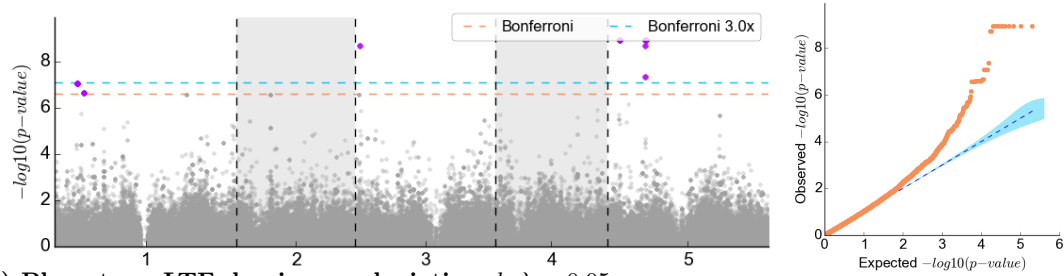
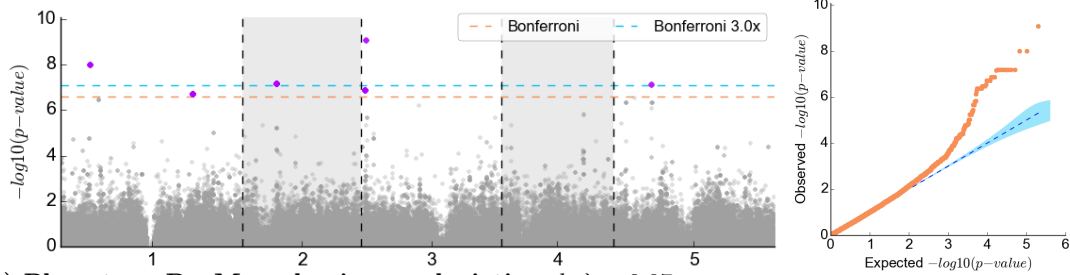
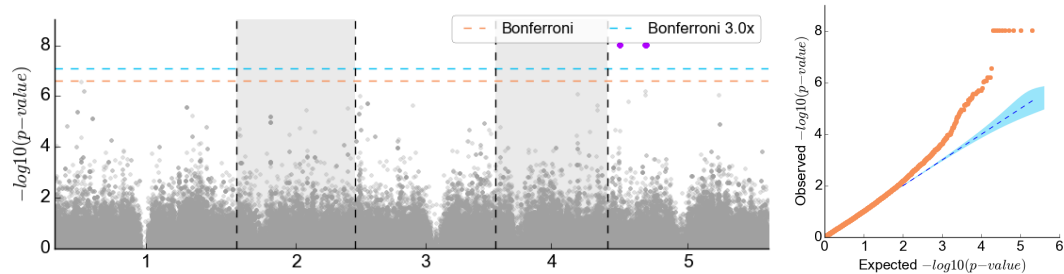
(a) Phenotype DTF dominance deviation d , $\lambda = 0.95$ (b) Phenotype LTF dominance deviation d , $\lambda = 0.95$ (c) Phenotype DryMass dominance deviation d , $\lambda = 0.97$ 

Figure 6.9: Manhattan plot and QQ-plot for dominance deviations (overdominant model): Manhattan and QQ-plots for three phenotypes with significantly associated SNPs when using an overdominant model and the derived dominance deviations d . Two Bonferroni thresholds are shown. The standard one using all SNPs (red dashed line) and the stringent one (blue dashed line) correcting for 3 experiments (additive, overdominant mean and overdominant dominance deviation). Magenta points are significantly associated SNPs.

— have specific predictions regarding the allele frequencies of causal loci. The dominance hypothesis expects that causal loci are rare in the population while the overdominance hypothesis forecasts intermediate frequencies of such loci. We performed a two sample t-test between the minor allele frequencies of all significantly associated SNPs (selected hits from FaSTLMM and SConES) and the SNPs not selected by any of our methods. We found that the mean of the minor allele frequency across all selected SNPs (mean = ~ 0.25) is significantly lower than the overall mean of all not selected SNPs (mean = ~ 0.31 , t-test $p_v = 3.96e^{-50}$). All these results suggest that hybrid superiority or inferiority may be due to genome-wide complementation of rare deleterious alleles in support of the dominance hypothesis of heterosis.

6.3.3 Analysis of Variance Explained

Next, we analysed variance explained by all available SNPs, all significantly associated hits, as well as all detected hits by SConES. Results are illustrated in Figure 6.11. Variance explained by all available SNPs was estimated using a linear mixed model

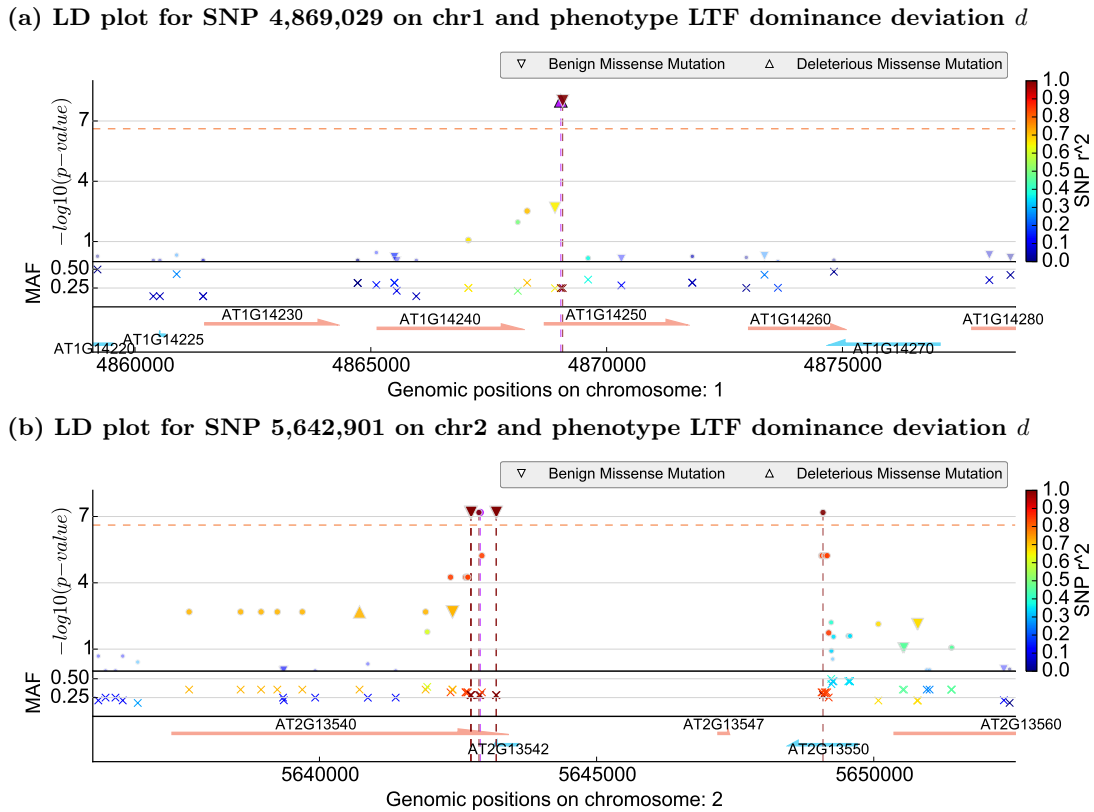


Figure 6.10: Linkage disequilibrium plots: LD plots for significantly associated hits using an overdominant model and the dominance deviation of the phenotype LTF.

by fitting the kinship matrix to the phenotype of interest (see Methods). Estimates ranged between 18% and 45% which accounts for 53% to 78% of the total genetic variation (H_b^2). In addition, we used a ridge regression to calculate variance explained measures using all significantly associated hits. A ridge regression was used to account for non-independence, or linkage, between significant hits. We found that significant hits account for a large fraction of the total genetic variance and explained up to 30% of the total genetic variance. Eventually, we also computed variance explained for all detected SNPs using SConES and found that these SNPs can explain up to 77% of the total genetic variance when using the overdominant genetic model.

6.3.4 Simulation of Phenotypes and Power-Analysis

Although we detect no significantly associated loci with the additive model and the estimated mean phenotypes, based on narrow sense heritability estimates (h_n^2) there is clearly an additive component to genetic variance in these diallels as shown in Figure 6.7. Using simulations, a power analysis was performed to evaluate whether loci with additive effects can be detected in our experimental population of hybrids. We evaluated the power by performed four experiments with different genotype encodings and simulated phenotypes: a) using an additive genotype encoding for a simulated additive phenotype, b) using an additive genotype encoding for a simulated overdominant phenotype, c) using an overdominant genotype encoding for a simulated additive phenotype and d) using an overdominant genotype encoding for a simulated overdominant phenotype. Results are illustrated in Figure 6.12. We found that the additive model

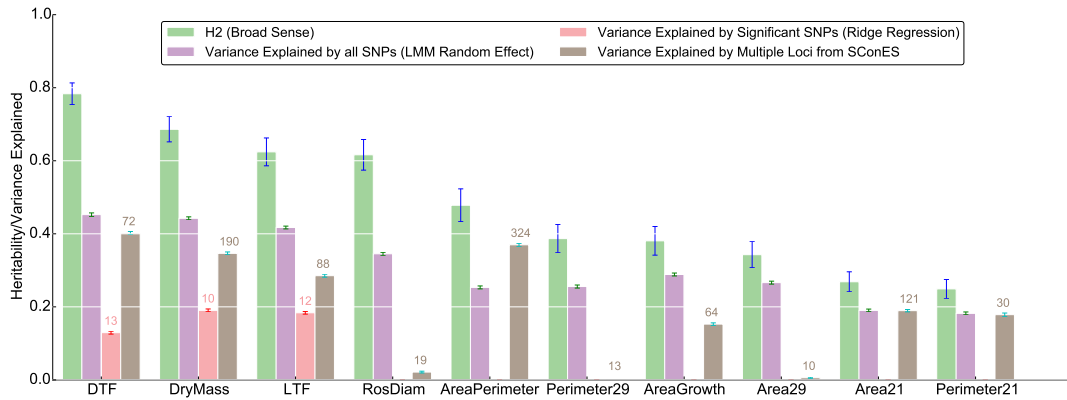


Figure 6.11: Variance explained and broad sense heritability estimates: Variance Explained by all SNPs, all significant hits and SConES. Numbers above bars indicate the number of associated hits used to determine variance explained.

is extremely underpowered in this dataset regardless of the effect size of the allele frequency of the causal SNP (Figure 6.12a). This could be the result of either correlation of such loci with population structure or to the limited genetic diversity of the source population. In addition, we observed that using the a non-suitable genotype encoding for a phenotype with an additive or overdominant effect leads to a severe loss in power (Figures 6.12b-c). Eventually, simulations showed that, in contrast to the additive model, the overdominant model had sufficient power to detect associations at SNPs with a range of effect sizes and minor allele frequencies (Figure 6.12d). This power analysis showed the importance of the diallel design to detect non-additive associations.

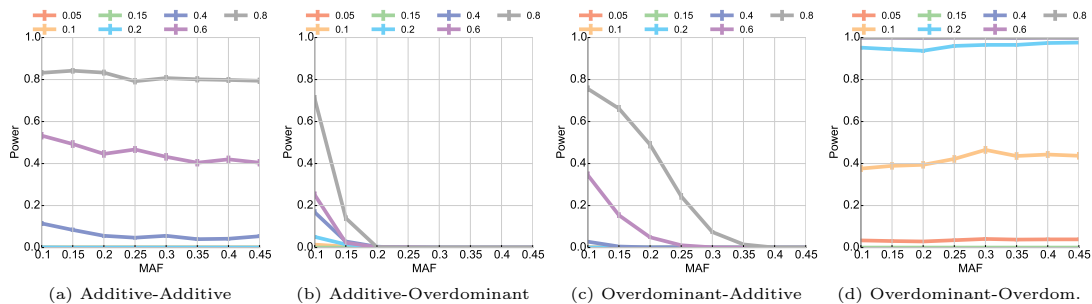


Figure 6.12: Power analyses of simulated phenotypes: a) Additive genotype encoding and simulated additive phenotype. b) Additive genotype encoding and simulated overdominant phenotype. c) Overdominant genotype encoding and simulated additive phenotype. d) Overdominant genotype encoding and simulated overdominant phenotype.

6.4 Chapter Summary

In a novel collaborative study with the Weigel lab [Seymour *et al.*, 2015] we demonstrated the full potential of our **easyGWAScore** framework by investigating the effect of non-additive genetic variance of hybrid phenotypes in *Arabidopsis thaliana*. For this purpose, we characterised the contribution of dominance to heterosis as a potential source of missing heritability. A large population of hybrid *Arabidopsis thaliana* individuals was created by using a half-diallel crossing scheme and in a first experiment

we computed broad- and narrow-sense heritability estimates based on all hybrids. We found that a large proportion of the heritability can be attributed to non-additive genetic variance. Further, we conducted several GWASs using the `easyGWASCore` framework. We investigated three different models: (i) using `FaSTLMM` with a standard additive SNP encoding on the hybrid mean phenotype, (ii) using `FaSTLMM` with an overdominant genotype encoding on the hybrid mean phenotype and (iii) using `FaSTLMM` with an overdominant genotype encoding on the estimated dominance deviation d . We found that model (iii) had the greatest power in detecting novel associations. Further, we found that SNPs detected with model (iii) can explain up to $\sim 30\%$ of the total genetic variance. Using `SConES` we even could explain up to $\sim 77\%$ of the total genetic variance when only selecting between 10 and 324 SNPs per phenotype. We showed that all selected SNPs had a significantly lower minor allele frequency than the remaining SNPs. These results suggest that hybrid superiority may be due to genome-wide complementation of rare deleterious alleles in support of the dominance hypothesis of heterosis.

CHAPTER 7

Conclusions and Outlook

Recent advances in next generation sequencing technologies have made it possible for the first time to sequence and analyse the genomes of whole populations of individuals in both a cost-effective manner and in a reasonable amount of time [*1000 Genomes Project Consortium et al.*, 2010; *Cao et al.*, 2011]. To better understand and investigate the genetic basis of common traits or diseases in a whole population of individuals, genome-wide association studies (GWASs) are often used as an integral tool [*McCarthy et al.*, 2008]. A variety of methods and tools have therefore been developed to tackle this question. Until 2013, more than 2,000 associations of more than 300 complex phenotypes have been identified [*Manolio*, 2013], including various human diseases (e.g. [*Pillai et al.*, 2009; *Rioux et al.*, 2007; *Schizophrenia Psychiatric Genome-Wide Association Study (GWAS) Consortium et al.*, 2011]). GWASs have also been successfully conducted in other species, such as *Arabidopsis thaliana* [*Atwell et al.*, 2010; *Filialt and Maloof*, 2012; *Meijón et al.*, 2014], *Oryza sativa indica* [*Zhao et al.*, 2011] or *Drosophila melanogaster* [*Mackay et al.*, 2012].

However, conducting GWASs is a challenging endeavour: first, different types of hidden confounding factors, such as population structure, environmental and technical influences could lead to spurious associations. Second, it has been shown that more than 80% of all identified variants are found in non-coding regions and that many of these identified variants often fail to explain much of the phenotypic variability [*Manolio et al.*, 2009]. Additional methods and tools have been developed and tailored to partly address some of these challenges, which consequently led to a large diversity of different tools and methods. Thus, a highly confusing and fragmented landscape of these tools was created and complicated the process to easily find the appropriate tool for a certain task. In addition, most of these tools do not share a common data input and output format which makes them unnecessarily difficult to use. Moreover, visualising and annotating the results is imperative for their interpretation but most often ignored by all of these tools. Third party solutions, such as *Haploview* [*Barrett et al.*, 2005], or custom *Python*, *R* or *Matlab* scripts have therefore be used.

In this thesis, we developed an integrated framework and cloud-service to (i) explain

more of the missing heritability, (ii) to simultaneously simplify the process of conduction and managing large GWAS projects and (iii) to also provide straightforward solutions to visualise and annotate the results.

Simplifying the Process of Conducting GWASs and Meta-Analyses

In Chapter 2 we described a selection of popular methods for GWASs and meta-analyses and created a framework to facilitate the usage of these methods. The framework, called `easyGWAScore`, includes various univariate methods, such as linear and logistic regression, as well as different mixed models to simultaneously account for population stratification [Kang *et al.*, 2008, 2010; Lippert *et al.*, 2011]. The core code of the `easyGWAScore` framework is written in C/C++. We created a common `Application Programming Interface` (API), as well as an easy to use `Python` interface. A data managing module was created, such that all algorithms can easily be used without the need to convert between different data formats. Additionally, we also integrated several popular meta-analysis methods to allow researchers to combine the summary results from precomputed GWASs. This is important because many labs do not share the raw genetics data anymore, due to the apparent success to infer surnames from anonymised genetic data [Gymrek *et al.*, 2013].

We showed on various examples how scientists can utilise the `easyGWAScore` API and gave detailed examples of how to use the API to develop novel user-specific algorithms. The modular structure of the `easyGWAScore` API is a key feature for a sustainable, flexible, easy extendable and competitive framework. We compared the runtime of the implemented algorithms to the runtime of its individual tools and found that the `easyGWAScore` implementations are at least as efficient as its individual tools. For some algorithms the `easyGWAScore` implementations were between 0.5 and 1 magnitude faster, e.g. the re-implementation of `FaSTLMM` [Lippert *et al.*, 2011]. However, the implementation of logistic regression was significantly slower than the implementation in `PLINK` [Purcell *et al.*, 2007]. This can be explained due to the custom implementation of the iterative Newton-Raphson procedure to solve the optimisation problem [Ypma, 1995]. Different and more efficient optimisation techniques should be applied and tested in future releases, such as a gradient descent optimiser.

The field of GWASs is a heavily studied research field and new contributions and discoveries are made frequently. Thus, it is essential for a sustainable framework to continuously update the `easyGWAScore` framework with novel and state-of-the art methods. Until now, the `easyGWAScore` framework includes popular univariate methods for performing GWASs (e.g. linear regression, logistic regression, linear mixed models [Kang *et al.*, 2010; Lippert *et al.*, 2011]) and meta-analyses, as well as two novel methods for multi-locus and multi-trait mapping [Azencott *et al.*, 2013; Sugiyama *et al.*, 2014] (see Chapter 5). In the future we would like to extend the framework with a larger set of different mapping methods, including an improved linear mixed model to better account for population stratification [Listgarten *et al.*, 2012], multi-locus and multi-trait approaches that are able to account for population structure [Korte *et al.*, 2012;

Lippert et al., 2013; Rakitsch et al., 2013a,b; Segura et al., 2012].

Currently, all methods in the **easyGWASCore** framework require at least one binary or continuous phenotype. We would like to extend the framework to more complex phenotypes, such as automatically detected shape phenotypes [*Karaletsos et al., 2012*], or to non-continuous phenotypes, such as trees or graphs [*Feragen et al., 2013*]. Both measurements could help to explain larger parts of the phenotypic variance since we would not rely on hand-picked characteristics and measurements.

The steady growth of sequencing data and the formation of large genetic consortia will require algorithms that scale to several hundreds of thousands of individuals in the near future. Thus, it is important to re-investigate many GWAS algorithms to speed up the computations for this enormous amount of data. Also, distributing computations across several computing nodes and exploiting graphical processing units (GPUs) will be of utmost importance in the near future. Techniques that have been successfully applied in big search engines, such as MapReduce [*Dean and Ghemawat, 2008*], will become more important and might help to build more scalable algorithms.

Until now, the **easyGWASCore** framework mainly uses SNPs as genetic markers. We would like to allow the use of other structural variations (e.g. deletions and insertions [*Grimm et al., 2013; Ye et al., 2009*]). Including structural variations might help to detect novel associations that affect the function of a gene and to gain additional insight about a certain diseases or phenotype [*Österberg et al., 2002; Weischenfeldt et al., 2013*].

Enhancing the Visualisation and Annotation Capabilities

A second tedious and labour-intensive step for any kind of GWASs are visualisations, annotations and interpretations of their results. The **easyGWASCore** framework provides an out-of-the-box solution to also generate commonly used Manhattan and QQ-plots. In Chapter 2 we described how to use the **easyGWASCore** Python command line interface to easily generate these visualisations. Moreover, linkage disequilibrium (LD) plots can be created for investigate a region of interest. LD plots are zoomed in Manhattan plots and illustrate the LD structure in a window around a focal SNP (e.g. a significantly associated SNP). In addition, these plots are enriched with an annotation functionality to also illustrate the minor allele frequency and the genes of these SNPs.

Furthermore, reliable strategies that allow scientists to further prioritise associated loci for further biological investigation are of high practical relevance. We therefore extended the annotation pipeline of the **easyGWASCore** framework to also highlight missense variants (mutations that lead to an amino-acid change), as well as if a given missense variant has a predicted damaging effect on the protein or not. These damaging effects are often referred to as pathogenic or deleterious effects as well. Recently, many tools have been developed that can predict the pathogenicity status of a given missense variant (e.g. [*Adzhubei et al., 2010; Kircher et al., 2014; Ng and Henikoff, 2003; Schwarz et al., 2014*]). However, it is not obvious which of these tools work best, that is generalise best to unknown missense variants. In Chapter 3 we investigated

the question whether there are systematic differences in the quality of the predictive performance of pathogenicity prediction tools when evaluated on a large number of variant databases [Grimm *et al.*, 2015]. We found that the existence of two types of circularity hinder the evaluation of these tools. The first type of circularity is due to overlaps between their training sets and the evaluation sets used to assess their predictive performance. Type 2 circularity, however, is closely linked to a statistical property of current variant databases. Variants from the same gene are often jointly labeled as being either pathogenic or neutral. This might lead to classifiers that predict pathogenicity based on known information about specific variants in the same gene. Thus, these classifiers could achieve excellent performances on these datasets, while being unable to detect novel risk genes. There exists also a potential third type of circularity we have not investigated in this thesis: it might be that variants already annotated by existing pathogenicity prediction tools and subsequently entered into a publicly available variant database. Here, it could be that tools that appear to perform well on “new” data, are in fact only recovering labels that they have given themselves. Thus, it is an important and necessary step to document the source of evidence that was used to assign the label to variants. In summary, we demonstrated in Chapter 3 that ignoring these types of circularity could lead to overly optimistic assessments of tool performances. Nevertheless, these tools are still an important resource to narrow down certain variants for further biological investigation. Therefore it is imperative knowing and avoiding these types of circularity. In Chapter 3 we provided several guidelines on how to avoid these two types of circularity and also proposed a new evaluation strategy to measure the performance of these tools in a fair and competitive way.

In the future we would like to extend the visualisation and annotation pipeline to also include additional biological information, such as gene ontologies (GO) [Ashburner *et al.*, 2000], or predictions about the potential effects of a given variant on genes, proteins or regulatory regions, e.g. from tools, such as `snpEFF` [Cingolani *et al.*, 2012] or variant effect predictor (VEP) [McLaren *et al.*, 2010]. The more additional biological information we take into account the more we could learn about the coherences between these different types of biological knowledge as we are moving away of only looking at a “single” event to a more “system” related view. This might help to better understand the mechanisms why certain loci are associated with a trait of interest.

A Cloud Service for Performing, Analysing and Sharing GWASs

Although the `easyGWASCore` framework facilitates the use of different algorithms, visualisation and annotation methods, it still requires that the user has basic Unix knowledge of how to use the command line or of how to write code. Many GWAS projects are collaborative studies between several scientists and different labs. Sharing data and results is therefore of utmost importance for large projects. Also, the ever-growing resources of publicly available GWASs data stored on different servers also complicates the retrieval and re-analysis of this data.

In Chapter 4 we introduced the `easyGWAS` cloud service and web-application for per-

forming, analysing, visualising and annotating, as well as sharing and hosting GWASs [Grimm *et al.*, 2012]. This web-application utilises the **easyGWASCore** API and offers a large set of popular methods for performing GWASs and meta-analyses. While existing web-applications for GWASs mainly focus on a fixed set of publicly available data from a single species [Childs *et al.*, 2012; Kirby *et al.*, 2010; Mackay *et al.*, 2012; Seren *et al.*, 2012], **easyGWAS** also allows the upload of new data for an arbitrary set of species. Moreover, **easyGWAS** provides methods to share data and results with collaborators and allows to publish those data centrally for the whole scientific community. For this purpose, we created a flexible data model that allows to control the permission rights of private data in an easy and simple fashion, e.g. users can share their data in such a way that their collaborators can perform GWASs on this data but will not have full access to the raw genotype data. These community related features, such as data sharing and publishing, are a central element of **easyGWAS**. With an increasing availability of publicly available GWAS scientists might gain more knowledge about these GWASs, for example when studying pleiotropic effects, that is the effect of a single marker or gene on different phenotypes. Also, comparing new GWASs with existing ones might help to obtain additional biological insights. Further, we developed a novel hybrid database model to efficiently and reliably store the large amount of GWASs data. The hybrid database is composed of a **PostgreSQL** database, several user specific **SQLite** databases for annotation related information and several **HDF5** files to store the GWAS data and results.

Another central element of **easyGWAS** are dynamic visualisation and annotations of results. Visualisations are dynamically updated when the user interacts with the plots, e.g. by changing the multiple hypothesis correction method or when zooming into interesting regions. Until now, Manhattan plots are enriched with gene annotations. However, it is important that the visualisation functionality will be extended to include other types of biological information as well.

To facilitate the process of conducting GWASs or meta-analysis via the web-browser we created an easy-to-use step-by-step procedure, also referred to as wizard, that guides the user through every necessary step. The user's input is analysed on the fly by the GWAS wizard. Based on the user's input the wizard offers automatically a selection of different filtering options and algorithms. One of the future goals would be to also develop a graphical click & drop module that supports users to create sophisticated procedures to conduct more intricate GWAS projects.

Network Guided Multi-Locus and Multi-Trait Methods to Explain Parts of the Missing Heritability

Many theories have been suggested that potentially could explain more of this missing heritability, such as including rare variants or variants with small effect sizes [Manolio *et al.*, 2009]. Also, considering additive or interactive effects between multiple markers could contribute to explain parts of the missing heritability [Marchini *et al.*, 2005]. However, investigating additive and especially multiplicative effects increases

the number of statistical tests enormously, which leads to additional computational, as well as statistical challenges and problems. Investigating multiplicative effects between pairs of loci is already computationally infeasible on a single desktop machine. Thus, tremendous efforts have been undertaken to develop novel algorithms that are able to detect these epistatic effects on a genome-wide setting by using mathematical and algorithmic tricks [Achlioptas *et al.*, 2011; Zhang *et al.*, 2008, 2009, 2010a] or by leveraging graphical processing units (GPUs) [Hemani *et al.*, 2011; Kam-Thong *et al.*, 2011, 2012]. While these algorithms address the computational problems, they still ignore the tremendous number of multiple hypothesis tests. Recently, different methods have been proposed to account for this enormous amount of multiple hypothesis by excluding non-testable hypothesis [Llinares-López *et al.*, 2015a,b; Sugiyama *et al.*, 2015; Terada *et al.*, 2013] based on a trick proposed by Tarone [1990]. For the detection of additive associations between several markers various regression based models have been developed [Cho *et al.*, 2010; Rakitsch *et al.*, 2013b; Wang *et al.*, 2011]. Although this models are able to detect multiple markers, they are often limited in power or hard to interpret. However, including *prior* biological knowledge can help scientists to better interpret results while at the same time boosting the statistical power. One of the largest problems is that current methods are limited to a predefined number of potential candidate sets [Cantor *et al.*, 2010; Fridley and Biernacka, 2011; Wu *et al.*, 2011]. For this purpose, we developed two novel methods for automatically detecting sets of genetic markers that are maximally associated with a given phenotype while being connected in an underlying biological network.

In Chapter 5 of this thesis we described a single-task version for network guided multi-locus mapping, called **SConES** for Selecting Conected Explanatory SNPs [Azencott *et al.*, 2013], as well as a multi-task version, called **Multi-SConES** [Sugiyama *et al.*, 2014]. We showed that the optimisation problem of **SConES** could be reformulated as a min-cut problem and thus solved exactly and efficiently by a maximum-flow algorithm [Boykov and Kolmogorov, 2004; Goldberg and Tarjan, 1988]. An advantage of **SConES** is that it automatically detects the sets of markers without the need of predefining potential candidate sets. Furthermore, our method is able to handle incomplete networks and is able to select different subnetworks. We also showed that the multi-task version, **Multi-SConES**, could be reformulated as single-task min-cut problem and thus solved exactly and efficiently.

We demonstrated on several simulated and real world experimenters that our methods have improved abilities to discover true causal features compared to other state-of-the-art methods, such as structured regression-based methods [Jacob *et al.*, 2009; Li and Li, 2008; Tibshirani, 1996]. Furthermore, we showed that **SConES** is able to account for parts of the missing heritability by explaining larger proportions of the phenotypic variance than univariate regression-based methods. This is an interesting point since regression based methods directly optimise with respect to predictivity. However, using a min-cut reformulation makes no assumptions about the distribution of the phenotype. So it might be possible that **SConES** outperforms the other methods if phenotypes are

not normally distributed. These results support the hypothesis that including multiple loci could contribute to explain parts of the missing heritability [Manolio *et al.*, 2009; Marchini *et al.*, 2005].

Both methods were integrated into the `easyGWASCore` framework and comprehensive runtime evaluations were conducted between different implementations in `Matlab` and `R`. Implementations in `easyGWASCore` were both efficient and scaled to genome-wide settings including several correlated phenotypes.

We analysed three different types of biological network, (i) a gene sequence network, (ii) a gene membership network and (iii) a gene interaction network. However, understanding the effects of the network topology and density in more detail is of high importance and should be studied in future projects. Also, including other types of *prior* biological knowledge and exploring their effects could be an interesting topic for future studies. One possibility would be to include pathogenicity prediction scores [Grimm *et al.*, 2015] by reweighting the SKAT association scores [Wu *et al.*, 2011]. Thus, one could prioritise the importance of SNPs based on their predicted damaging effect on proteins. A second possibility would be to include the three dimensional genome organisation of the local chromatin packing by using detailed Hi-C data [Wang *et al.*, 2015]. Doing so we could explore the effects of genes or regions in the DNA that are in close physical distance.

Another important research topic for `SConES`, as well as all other structured regularised regression based methods, is to evaluate the statistical significance of sets of selected features. Regularised feature selection approaches, such as `SConES` or its `LASSO` comparison partners, do not lend themselves well to the computation of p-values. Permutation tests could be an option, but the number of permutations to run is difficult to evaluate. Another possibility would be to implement the multiple-sample splitting approach proposed by Meinshausen *et al.* [2012]. However, the loss of power from performing selection on only subsets of the samples is too large, given the sizes of current genomic datasets, to make this feasible.

Non-Additive Genetic Variance as a Potential Source of Missing Heritability

In Chapter 6, we investigated another potential source that might contribute in part to the missing heritability. Heritability can be estimated as broad sense or narrow sense heritability. Broad sense heritability quantifies the overall genetic contributions to the total phenotypic variance of a whole population of individuals, including additive, dominant and epistatic effects. However, narrow sense heritability only quantifies the additive genetic contributions to the total phenotypic variance. In GWASs we usually ignore the non-additive genetic contribution. Thus, only narrow sense heritability estimates can be computed for GWASs. We therefore investigated in a case-study the effect of non-additive genetic variance on hybrid phenotypes in *Arabidopsis thaliana* and characterised the contribution of dominance to heterosis — that is the phenotypic superiority of progeny or vigour of a hybrid cross relative to their genetically distinct parents [Baranwal *et al.*, 2012; Falconer and Mackay, 1995] — as a potential source of

missing heritability [Seymour *et al.*, 2015]. For the analysis of this study we utilised the **easyGWASCore** framework and demonstrated its full potential. We found that a large proportion of the heritability can be attributed to non-additive genetic variance. Significant hits identified by the dominance deviation of the hybrid phenotype — that is the deviation of the hybrid phenotypes to its estimated mean phenotype of its two parents — could explain up to $\sim 30\%$ of the total genetic variance. In addition, we found that **SConES** could explain up to $\sim 77\%$ of the total genetic variance while at the same time only selecting between 10 and 324 SNPs per phenotype. In summary these results suggested that non-additive effects might be an important source to explain parts of the missing heritability.

APPENDIX A

Nomenclature

- n is the number of samples from a given population (e.g. in a given dataset)
- m is the number of genetic markers (SNPs) in a given dataset
- d is the number of fixed effects in a model. A fixed effect can be genetic markers and/or covariates
- λ, η, μ are often used as regularisation parameters
- p denotes a probability
- p_v denotes a p-value
- $a \in \{A, G, T, C\}$ is an allele and can be one of the four nucleotides A, G, T or C
- β_0 is the weight of the intercept in a regression model
- $\boldsymbol{\beta} \in \mathbb{R}^{n \times m+1}$ is a vector of regression weights including the intercept
- $\mathbf{g} = (a_i, \dots, a_n)^\top$ is a genetic marker with n alleles, where a_i is the allele of the i th sample/individual
- $\mathbf{y} = (y_i, \dots, y_n)^\top \in \mathbb{R}^n$ is a n -dimensional vector of phenotypic measurements, where y_i is the phenotype of the i th sample/individual
- $\mathbf{y}_c \in \mathbb{R}^n$ is a n -dimensional vector with quantitative phenotypic measurements
- $\mathbf{y}_b \in \{0, 1\}$ is a n -dimensional dichotomous/binary vector
- $\mathbf{M} = (g_1, \dots, g_m)$ is a $n \times m$ matrix of genetic markers
- $\mathbf{1} = (1_1, \dots, 1_n)^\top$ is a n -dimensional vector of ones
- \mathbf{I} is a $n \times n$ identity matrix
- $\mathbf{G} = (\mathbf{1}, \mathbf{g}_1, \dots, \mathbf{g}_m) \in \mathbb{R}^{n \times m+1}$ is a matrix containing the intercept of a regression model and a subset of m genetic markers

- $\mathbf{C} \in \mathbb{R}^{n \times d}$ is a matrix of fixed effects, such as covariates
- $\mathbf{K} \in \mathbb{R}^{n \times n}$ is a $n \times n$ symmetric positive semi-definite kinship matrix of genetic similarities between a set of n samples
- \mathbf{H} is a Hessian matrix
- \mathbf{W} is a diagonal matrix
- G is an arbitrary graph
- V are the vertices or nodes of graph G
- E are the edges between two vertices of a graph G
- \mathbf{A} is an adjacency matrix for a given graph G
- $\mathbf{c} \in \mathbb{R}^m$ is a vector of association scores
- $\mathbf{f} \in \{0, 1\}^m$ is an indicator vector for any genetic marker j , where $f_j = 1$ if the marker is selected and $f_j = 0$ if not
- \mathcal{S} is a set or subset of genetic markers
- \mathbf{L} is the Laplacian, where $\mathbf{L} = \mathbf{W} - \mathbf{A}$

APPENDIX B

Performance Evaluation Statistics

Various statistics can be derived from a confusion matrix to evaluate the performance of trained models or tools (Table B.1). In a typical classification task one tries to separate two different classes — often referred to as the positive and negative class — from each other and predict the correct class label for an unknown test point. Let \mathbf{P} be the set of all observed positive predictions and \mathbf{N} be the set of all observed negative prediction. Here, a test point is defined as a true positive (TP) if and only if the test point corresponds to the positive class and as true negative (TN) if and only if the test point corresponds to the negative class. Accordingly, a false positive (FP) is a

	Actual Positive	Actual Negative
Test Positive	True Positive (TP)	False Positive (FP), Type I Error
Test Negative	False Negative (FN), Type II Error	True Negative (TN)

Table B.1: Confusion matrix

actual negative test point that is classified to be positive and a false negative (FN) a positive test point classified to be a negative one. Based on this confusion matrix one can derive several statistics:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (\text{B.1})$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (\text{B.2})$$

$$\text{Recall/Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (\text{B.3})$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (\text{B.4})$$

$$\text{Negative Predictive Value (NPV)} = \frac{\text{TN}}{\text{TN} + \text{FN}} \quad (\text{B.5})$$

$$\text{F-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (\text{B.6})$$

$$\text{Matthews Correlation Coefficient (MCC)} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (\text{B.7})$$

In addition, one can assess the performance of a model by computing receiver operation characteristic curves (ROC-curves), that is the fraction of the TP over all positives $P = \text{TP} + \text{FN}$ (Sensitivity or Recall) against the fraction of the FP over all negatives $N = \text{TN} + \text{FP}$ (1-Specificity of False Positive Rate). Furthermore, the Precision-Recall curve (ROC-PR-curve) [Davis and Goadrich, 2006] is the fraction of the TP over all positives $P = \text{TP} + \text{FN}$ (Recall) against the fraction of the TP over all $\text{TP} + \text{FP}$

(Precision). To measure the performance, one can compute the area under the ROC or ROC-PR curves (AUC and AUC-PR, respectively). The area under the curve can take values between 0 and 1. A perfect classifier has an AUC and AUC-PR of 1. The AUC of a random classifier is 0.5.

APPENDIX C

General GWAS related terminology

C.1 Minor Allele Frequency

Minor allele frequency (MAF) is the frequency of the least common allele that occurs in a given genetic marker g of size n , where n is the number of samples (Figure C.1).

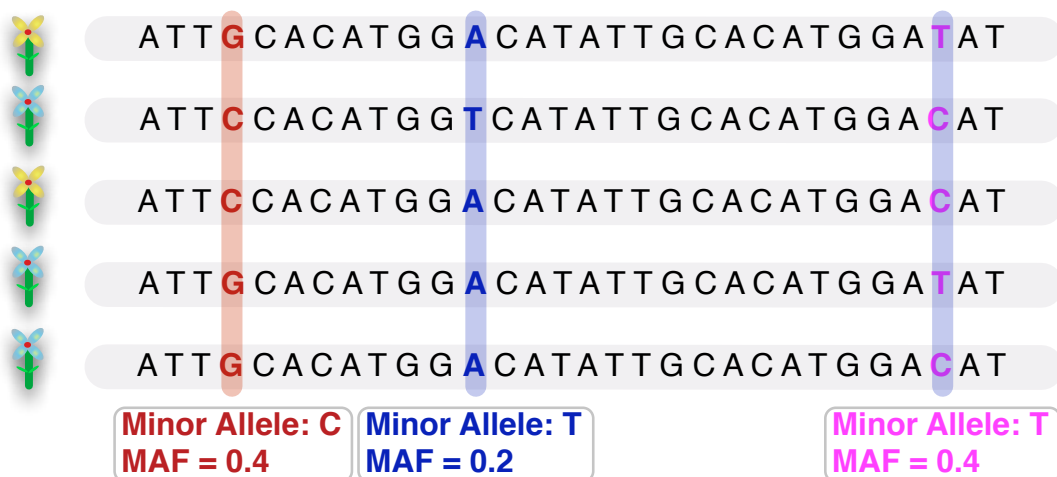


Figure C.1: Illustration of minor allele frequency. These genetic markers are illustrated together with the minor allele and the minor allele frequency (MAF).

C.2 Genotype Encoding

Genotype data can be encoded in different ways. The most popular encoding is an additive genotype encoding. Here, the major allele is encoded as 0, the heterozygous allele with 1 and the minor allele with 2. The recessive genotype encoding, however, encodes the major and heterozygous allele with 0 and the minor allele with 1. The dominant genotype encoding, encodes the major allele with 0 and the remaining two alleles (heterozygous and minor allele) with 1. Finally, the overdominant encoding encodes the major and minor allele as 0 and the heterozygous allele with 1.

APPENDIX D

easyGWASCore API and Python Command Line Interface Overview

D.1 The Application Programming Interface

In the following we list different methods and function of the `easyGWASCore` API. Methods are structured by the different abstraction layers as described in Chapter 2. The following overview is only an excerpt of the most important classes, methods and functions.

Layer 1 Modules, Classes and Methods

Class Name	Method/Function Name	Description
CCrossValidation	CCrossValidation(float seed[optional])	Crossvalidation Helper Class
	train_test_split(int n, float ratio)	Generate a train and test split for n samples with a certain ratio
	kFold(int k, int n)	Generate training and testing splits for n samples and k folds
	ShuffleSplit(int n, int k, int, ratio)	Generate k random splits for n samples with a certain ratio
	getTrainingIndices(int k)	Get training indices for set/fold k
	getTestingIndices(int k)	Get testing indices for set/fold k
CMathHelper	pinv(MatrixXd& input, MatrixXd* output)	Computes pseudo inverse of a input matrix and writes it to an output matrix
	int factorial(int n)	Returns the factorial: n!
	float beta(float x, float y)	Returns the value of the Beta function at x and y
	float lbeta(float x, float y)	Returns the log of the Beta function at x and y
	float erfinv(float p)	Returns the inverse of the error function
ArgSort	bool isOdd(int n)	Returns true if n is odd
	ArgSort(VectorXd& x) VectorXd getIndices()	(Constructor): Class to sort the input vector x in decreasing order Returns sorted indices for input vector
CMatrixHelper	MatrixXd sliceRowsMatrix(MatrixXd& X, VectorXd& indices)	Slice rows of matrix X at indices
	MatrixXd sliceColsMatrix(MatrixXd& X, VectorXd& indices)	Slice columns of matrix X at indices
	MatrixXd permuteVector(VectorXd& y, int p)	Permute vector y, p times

Table D.1: utils Module: Contains several helper methods for various different general tasks. Only the most important methods are shown.

Class Name	Method/Function Name	Description
CKernels	realizedRelationshipKernel(MatrixXd& X)	Returns the realised relationship kinship matrix for input matrix X

Table D.2: kernel Module: Contains Kernel related methods.

Class Name	Method/Function Name	Description
CBrentOptimizer	float solve(CBrentFunction* func, float lower, float upper, float epsilon, int max_it)	Root finding algorithm for a given function func

Table D.3: optimiser Module: Contains different optimisation methods.

Class Name	Method/Function Name	Description
CStats	float pearson_corr(VectorXd& v1, VectorXd& v2)	Computes Pearson's correlation coefficient between two vectors v1 and v2
	float pearson_pval(float r, int n, string tail="both")	Computes the p-value for a Pearson's r value for n samples
	MatrixXd principle_components(MatrixXd& X)	Returns principle components of X sorted by decreasing eigenvalues
	float varf(VectorXd& x)	Returns variance for vector x
	float stdf(VectorXd& x)	Returns standard deviation for vector x
	VectorXd std(MatrixXd& X, int dim)	Return vector with standard deviations of matrix X along dimension dim
CBeta	VectorXd mean(MatrixXd& X, int dim)	Return vector with mean values of matrix X along dimension dim
	float cdf(float x, float k)	Return CDF of Beta distribution for value x and k degrees of freedom
	float logcdf(float x, float k)	Returns logarithm of CDF
	float sf(float x, float k)	Return survival function
	float isf(float x, float k)	Return inverse of survival function
	float logsf(float x, float k)	Return logarithm of survival function
CChi2	float pdf(float x, float k)	Return probability distribution function
	float logpdf(float x, float k)	Return logarithm of probability distribution function
	float cdf(float x, float k)	Return CDF of χ^2 distribution for value x and k degrees of freedom
	float logcdf(float x, float k)	Returns logarithm of CDF
	float sf(float x, float k)	Return survival function
	float isf(float x, float k)	Return inverse of survival function
CGamma	float logsf(float x, float k)	Return logarithm of survival function
	float pdf(float x, float k)	Return probability distribution function
	float logpdf(float x, float k)	Return logarithm of probability distribution function
	float cdf(float x, float k)	Return CDF of Gamma distribution for value x and k degrees of freedom
	float logcdf(float x, float k)	Returns logarithm of CDF
	float sf(float x, float k)	Return survival function
CGaussian	float logsf(float x, float k)	Return logarithm of survival function
	float pdf(float x, float k)	Return probability distribution function
	float logpdf(float x, float k)	Return logarithm of probability distribution function
	float cdf(float x, float k)	Return CDF of Gaussian distribution for value x and k degrees of freedom
	float logcdf(float x, float k)	Returns logarithm of CDF
	float sf(float x, float k)	Return survival function
CFisherF	float isf(float x, float k)	Return inverse of survival function
	float logsf(float x, float k)	Return logarithm of survival function
	float pdf(float x, float k)	Return probability distribution function
	float logpdf(float x, float k)	Return logarithm of probability distribution function
	float cdf(float x, float k)	Return CDF of Fisher F distribution for value x and k degrees of freedom
	float logcdf(float x, float k)	Returns logarithm of CDF
CStudentT	float sf(float x, float k)	Return survival function
	float isf(float x, float k)	Return inverse of survival function
	float logsf(float x, float k)	Return logarithm of survival function
	float pdf(float x, float k)	Return probability distribution function
	float logpdf(float x, float k)	Return logarithm of probability distribution function
	float cdf(float x, float k)	Return CDF of Student T distribution for value x and k degrees of freedom

Table D.4: stats Module: Contains several statistical helper functions as well as different distribution function. Only the most important methods are shown.

Class Name	Method/Function Name	Description
CPlinkParser	readPEDFile(string fn, GWASData* data)	Read PLINK PED file from file fn and store everything a data container
	readMAPFile(string fn, GWASData* data)	Read PLINK MAP file from file fn and store everything a data container
	readPhenotypeFile(string& fn, GWASData* data)	Read PLINK phenotype file fn and store phenotypes in data container
CGWASDataIO	writeSummaryOutput(string& fn, GWASData& data, GWASResults& res)	Write results from GWAS to output file fn
	writeFilteredPlinkFile(string& fn, GWASData& data)	Write a filtered data matrix in PLINK format to disc
	GWASResults readGWASResults(string& fn)	Read results from a GWAS output file
	writeMetaResultsFile(string& fn, CMetaResults& res)	Write results from a meta-analysis to disc
CSconesIO	readSparseNetworkFile(string fn, GWASData* data)	Read a Sparse Network File into data container
	writeOutput(string fn, GWASData& data, VectorXd& indicator, float bl, float be)	Write Scones results to disc
CLogging	writeCMatrix(string& fn, MatrixXd& cmat, CSconesSettings& set)	Write matrix with all consistency indices to disc
	CLogging(string& fn[optinal])	Create log file
	log(string& mode, string& msg)	Write msg to file or print to screen

Table D.5: io Module: Different classes for data input/output. Only the most important methods are shown.

Layer 2 Modules, Classes and Methods

Class Name	Method/Function Name	Description
CLinearRegression	CLinearRegression()	Constructor
	CLinearRegression(bool intercept)	Alternative constructor to set intercept
	MatrixXd getCovarianceBetas()	Get covariance matrix betas
	VectorXd getStdBetas()	Get standard deviations betas
	getMSE()	Get mean-squared error
	getRMSE()	Get RMSE
	void fit()	Fit Regression (Different Arguments can be passed)
	void predict(VectorXd*,MatrixXd)	Get prediction for new data
CLogisticRegression	CLogisticRegression()	Constructor
	CLogisticRegression(bool intercept)	Alternative constructor to set intercept
	int getIterations()	Get number of iterations for the optimiser
	MatrixXd getCovarianceBetas()	Get covariance matrix betas
	VectorXd getStdBetas()	Get standard deviations betas
	getMSE()	Get mean-squared error
	getRMSE()	Get RMSE
	getYHat()	Get residuals
void fit()	Fit Regression (Different Arguments can be passed)	
void predict(VectorXd*,MatrixXd)	Get prediction for new data	
CLinearMixedRegression	CLinearMixedRegression()	Constructor
	CLinearMixedRegression(bool intercept)	Alternative constructor to set intercept
	MatrixXd getCovarianceBetas()	Get covariance matrix betas
	VectorXd getStdBetas()	Get standard deviations betas
	getMSE()	Get mean-squared error
	getRMSE()	Get RMSE
	getYHat()	Get residuals
	getLogDelta()	Get estimated ratio parameter δ
getLogSigma()	Get estimated noise variance σ	
void fit()	Fit Regression (Different Arguments can be passed)	
void predict(MatrixXd,MatrixXd)	Get prediction for new data	

Table D.6: regression Module: Contains different regression based methods. Only the most important methods are shown.

Class Name	Method/Function Name	Description
CombinedPvalues	float64 FisherMethod(VectorXd)	Combine p-values using Fisher's method
	float64 StoufferZ(VectorXd)	Use StoufferZ to combine p-values and return a z-score
	float64 StoufferPval(float64)	Compute p-value for z-scores
FixedEffectModel	float64 StoufferZWeighted(VectorXd, VectorXd, VectorXd)	Also weight p-values and add effect directions
	FixedEffectModel(VectorXd, VectorXd)	Fixed effect model for meta-analysis
	void process()	Run model
RandomEffectModel	float64 getPvalue()	Return p-value for meta-analysis
	RandomEffectModel(VectorXd, VectorXd)	Random effect model for meta-analysis
	void process()	Run model
	float64 getPvalue()	Return p-value for meta-analysis

Table D.7: meta Module: Contains different meta-analysis methods. Only the most important methods are shown.

Layer 3 Modules, Classes and Methods

Class Name	Method/Function Name	Description
LinearRegression	LinearRegression()	Linear Regression constructor for GWASs (different arguments are possible)
	void test_associations()	Test for associations
	void permutations()	Perform a permutation based association test
	void setPhenotype(VectorXd)	Set phenotype
	void setGenotype(MatrixXd)	Set genotype
	void setCovariates(MatrixXd)	Set matrix of covariates
	void setIntercept(bool)	Set intercept of model
	float64 getLogLikelihoodNullModel()	Get log likelihood of null model
	VectorXd getLogLikelihoodAlternativeModels()	Get log likelihood estimates from all alternative models for all SNPs
	VectorXd getPValues()	Get p-values
	VectorXd getPermutationPValue()	Get permutation based p-values
	MatrixXd getBetas()	Get β estimates
	MatrixXd getSEBetas()	Get standard errors of β estimates
	VectorXd getTestStatistics()	Get computed test statistics
	float64 getAIC()	Get AIC measure
	float64 getCAC()	Get cAIC measure
	float64 getBIC()	Get BIC measure
LogisticRegression	LogisticRegression()	Logistic Regression constructor for GWASs (different arguments are possible)
	void test_associations()	Test for associations
	void permutations()	Perform a permutation based association test
	void setPhenotype(VectorXd)	Set phenotype
	void setGenotype(MatrixXd)	Set genotype
	void setCovariates(MatrixXd)	Set matrix of covariates
	void setIntercept(bool)	Set intercept of model
	float64 getLogLikelihoodNullModel()	Get log likelihood of null model
	VectorXd getLogLikelihoodAlternativeModels()	Get log likelihood estimates from all alternative models for all SNPs
	VectorXd getPValues()	Get p-values
	VectorXd getPermutationPValue()	Get permutation based p-values
	MatrixXd getBetas()	Get β estimates
	MatrixXd getSEBetas()	Get standard errors of β estimates
	VectorXd getTestStatistics()	Get computed test statistics
	float64 getAIC()	Get AIC measure
	float64 getCAC()	Get cAIC measure
	float64 getBIC()	Get BIC measure
EMMAX	EMMAX()	EMMAX constructor for GWASs with population stratification correction (different arguments are possible)
	void test_associations()	Test for associations
	void permutations()	Perform a permutation based association test
	void setPhenotype(VectorXd)	Set phenotype
	void setGenotype(MatrixXd)	Set genotype
	void setCovariates(MatrixXd)	Set matrix of covariates
	void setK(MatrixXd)	Set kinship matrix for population structure correction
	void setIntercept(bool)	Set intercept of model
	void setREML(bool)	Use REML estimates
	void setBrent(bool)	Use Brent optimiser
	float64 computeVarianceExplainedNullModel(uint)	Compute variance explained by null model using a n-fold cross-validation
	float64 getHeritabilityEstimate()	Get heritability estimates
	float64 getGeneticVariance()	Get genetic variance
	float64 getNoiseVariance()	Get noise variance
	float64 getLogLikelihoodNullModel()	Get log likelihood of null model
	VectorXd getLogLikelihoodAlternativeModels()	Get log likelihood estimates from all alternative models for all SNPs
	VectorXd getPValues()	Get p-values
VectorXd getPermutationPValue()	Get permutation based p-values	
MatrixXd getBetas()	Get β estimates	
MatrixXd getSEBetas()	Get standard errors of β estimates	
VectorXd getTestStatistics()	Get computed test statistics	
float64 getAIC()	Get AIC measure	
float64 getCAC()	Get cAIC measure	
float64 getBIC()	Get BIC measure	
FaSTLMN	FaSTLMN()	FaSTLMN constructor for GWASs with population stratification correction (different arguments are possible)
	void test_associations()	Test for associations
	void permutations()	Perform a permutation based association test
	void setPhenotype(VectorXd)	Set phenotype
	void setGenotype(MatrixXd)	Set genotype
	void setCovariates(MatrixXd)	Set matrix of covariates
	void setK(MatrixXd)	Set kinship matrix for population structure correction
	void setIntercept(bool)	Set intercept of model
	void setREML(bool)	Use REML estimates
	void setBrent(bool)	Use Brent optimiser
	float64 computeVarianceExplainedNullModel(uint)	Compute variance explained by null model using a n-fold cross-validation
	float64 getHeritabilityEstimate()	Get heritability estimates
	float64 getGeneticVariance()	Get genetic variance
	float64 getNoiseVariance()	Get noise variance
	float64 getLogLikelihoodNullModel()	Get log likelihood of null model
	VectorXd getLogLikelihoodAlternativeModels()	Get log likelihood estimates from all alternative models for all SNPs
	VectorXd getPValues()	Get p-values
VectorXd getPermutationPValue()	Get permutation based p-values	
MatrixXd getBetas()	Get β estimates	
MatrixXd getSEBetas()	Get standard errors of β estimates	
VectorXd getTestStatistics()	Get computed test statistics	
float64 getAIC()	Get AIC measure	
float64 getCAC()	Get cAIC measure	
float64 getBIC()	Get BIC measure	

Table D.8: gwas Module for single trait GWASs: Contains different classes for single trait GWASs. Only the most important methods are shown.

Class Name	Method/Function Name	Description
	CScones()	Scones constructor for multi locus GWASs (different arguments are possible)
	void test_associations()	Test for associations
	void setSKATWeights(VectorXd)	Set different weights for SKAT to reweight SNP importance
CScones	VectorXd getIndicatorVector()	Get indicator vector with selected SNPs
	float64 getObjectivescore()	Get score from objective function
	float64 geBestLambda()	Get optimal lambda parameter
	float64 getBestEta()	Get optimal eta parameter
	MatrixXd getCMatrix()	Get matrix with all consistency/stability values for all etas and lambdas

Table D.9: gwas Module for multi trait GWASs: Contains different classes for SConES. Only the most important methods are shown.

D.2 The easyGWASCore Command Line Python Interface

All data Sub-Commands

Listing D.1: easyGWASCore data sub-commands

```

1 $: python python/easygwascore.py data -h
2
3 Convert Plink into easyGWASCore HDF5 file:
4 --plink2hdf5          Convert Plink into an easyGWASCore HDF5 file
5 --plink_data PLINK_DATA
6                       Prefix of Plink input files (*.ped, *.map)
7 --hout HOUT          Filename for HDF5 output file
8 --plink_phenotype PLINK_PHENOTYPE
9                       Plink input file or directory with plink input files
10                      (Optional)
11 --maf MAF           Remove SNPs with a population based minor allele
12                      frequency (MAF) smaller than specified (default=0)
13 --exclude_snps EXCLUDE_SNPS
14                      Remove a list of SNP identifiers (Optional)
15 --distinct_filter DISTINCT_FILTER
16                      Exclude SNPs that are not distinct or share a certain
17                      pattern more often than x (default: no filtering)
18
19 File Input Flags:
20 --hdata HDATA       Filename of HDF5 input file (needed for options
21                      {-encode, --vcf, --hdf5toplink})
22 --hfile HFILE       HDF5 result input file or directory with several input
23                      files (needed for options {-csv,--ld})
24
25 Convert HDF5 File into Plink files:
26 --hdf5toplink       Convert HDF5 file into PLINK files
27 --pout POUT         Path with file prefix to PLINK output folder
28
29 Convert HDF5 File into Plink files (SPLIT DATA INTO TRAIN/TEST):
30 --hdf5toplink_split Convert HDF5 file into PLINK files
31 --spout SPOUT       Path with file prefix to PLINK output folder
32 --ratio RATIO       Splitting Ratio Training:Testing Set (default=0.2, 80% Training , 20% Testing)
33
34 Encode data in HDF5 file:
35 --encode {additive,dominant,recessive,overdominant}
36                      Encode raw data matrix in HDF5 file and store encoded data in file
37
38 HDF5 to VCF File:
39 --vcf              Convert HDF5 file to VCF output file
40 --vout VOUT        Filename for VCF output file
41
42 Gene GFF File to SQLite Database:
43 --gff2sql          Store genes from GFF file in local SQLITE3 database
44 --gfile GFILE      GFF input file
45 --sqlout SQLOUT    SQL output file
46
47 HDF5 Result file to CSV File:
48 --csv             Write HDF5 Results to CSV output file
49 --cout COUT       Path to CSV output folder
50
51 HDF5 Result file to Gene Annotation Output:
52 --agene           Write HDF5 Results to an Annotated Gene Output File
53 --sqlfile SQLFILE SQL input file
54 --gout GOUT       Path to output folder
55 --topx TOPX       Write top x associations with genes to output file (default=1000)
56
57 Create Linkage-Disequilibrium Output for HDF5 Result Files:
58 --ld             Create Linkage-Disequilibrium Files
59 --ldout LDOUT     Path to output folder
60 --snp SELECTED_SNP SNP identifier that should be used for analysis
61                      (default: loop over all significantly associated SNPs)
62 --distance DISTANCE Distance in bp around the selected SNP (default=10000)
63 --r2-measure {excoffier_slarkin,pearson_r2,roger_huff}
64                      Choice of Linkage Disequilibrium measure (default:
65                      Excoffier-Slatkin)
66 --nhypothesis NR_HYPOTHESIS, -n NR_HYPOTHESIS
67                      Number of hypothesis to correct for (default: all
68                      markers)
69 --distinct, -d    Use distinct SNPs only to compute multiple hypothesis
70                      threshold
71 --ignore IGNORE   Ignore Phenotypes that contain a certain string (Optional)

```

All gwas Sub-Commands

Listing D.2: easyGWASCore gwas sub-commands

```

1 $: python python/easygwascore.py gwas -h
2 General parameters shared by all algorithms:
3 --out OUT Path to output directory (Required)
4 --hdata HDATA HDF5 File containing the genotype data (Required)
5 --maf MAF Remove SNPs with a minor allele frequency (MAF)
6 smaller than specified (default=0)
7 --encoding {additive,dominant,recessive,overdominant}
8 Encode Genotype with a different encoding (Default: additive)
9 --transform {sqrt,log10,boxcox,zeroMean,unitVariance}
10 Transform Phenotype (default: No transformation)
11 --homozygous Genotype is homozygous (default=False)
12 --phenotype_id PHENOTYPE_ID
13 Specify certain phenotype in HDF5 file. If not
14 specified loop over all phenotypes (default=all)
15 --algorithm {linear,logit,FaSTLMM,EMMAX,ttest,fisher,WCrT,MWUrT,linearperm,logitperm,EMMAXperm}
16 Select Algorithm
17 --threads THREADS If multiple phenotypes in file parallelise
18 computations (default=Available CPUs - 1)
19 --pcs PRINCIPLE_COMPONENTS
20 Number of Principle Components (default=0)
21 --pc_iterative Iterate through the number of 'pcs' (default=False)
22
23 Linear Mixed Model specific parameters:
24 --unique_snps Use unique SNPs only to compute kinship matrix (default=False)

```

All plot sub-commands

Listing D.3: easyGWASCore plot sub-commands

```

1 $: python python/easygwascore.py plot -h
2 optional arguments:
3 --hfile HFILE HDF5 result input file or directory with several input files
4 --csvfile CSVFILE CSV result input file or directory with several input files
5 --out OUT Path to output directory (Required)
6 --ifformat Image File Format (Default: png) {png,pdf,tiff}
7 --nogc Do not compute genomic control (GC) and display in plot
8 --phenotype_id PHENOTYPE_ID
9 Specify certain phenotype in HDF5 file. If not
10 specified loop over all phenotypes (default=all)
11
12 Common Plotting Parameters:
13 --distinct, -d Use distinct SNPs only to compute multiple hypothesis threshold
14 --notitle Add title to plots including Phenotype name and lambda value
15 --nhypothesis NR_HYPOTHESIS, -n NR_HYPOTHESIS
16 Number of hypothesis to correct for (default: all markers)
17 --nhypothesis2 NR_HYPOTHESIS2, -n2 NR_HYPOTHESIS2
18 Second Number of hypothesis to correct for (default: no markers)
19 --ignore IGNORE Ignore Phenotypes that contain a certain string (Optional)
20
21 Manhattan-Plots:
22 --manhattan Create a Manhattan plot
23
24 QQ-Plots:
25 --qqplot Create a Quantile-Quantile (QQ)-plot
26 --estpv Add the estimated theoretical distribution of p-values to the plot
27
28 LD-Plots:
29 --ldplot Create a Manhattan Linkage Plot
30 --hdata HDATA HDF5 File containing the genotype data (Required)
31 --snp SELECTED_SNP SNP identifier that should be used for analysis
32 (default: loop over all significantly associated SNPs)
33 --distance DISTANCE Distance in bp around the selected SNP (default=10000)
34 --r2-measure {excoffier,slatkin,pearson_r2,roger_huff}
35 Choice of Linkage Disequilibrium measure (default: Excoffier-Slatkin)
36 --sql_gene SQL_GENE Add gene annotations to plot. Requires a SQL file
37 generated from GFF file (see data manipulation commands) (Optional)
38 --pathogenicity_scores SFILE Add pathogenicity scores to LD plot (Optional)
39 --maf Remove SNPs with a population based minor allele frequency (MAF)
40 smaller than specified (default=0)

```

List of Figures

1.1	Illustration of a genome-wide association study: Simple illustration of a GWAS using a binary plant phenotype (plant flowers yellow vs. plant flowers blue) and three SNPs. The green SNP is associated with the phenotype whereas the others are not.	2
1.2	Illustration of an association with a non-causal SNP: Here, the causal SNP (red) was not sequenced. However, an indirect association with a linked SNP (blue) can be observed. . .	3
1.3	Combing the different puzzle pieces of a GWAS: To successfully perform a complete GWAS many different pieces have to be combined. The objective of this thesis is to contribute to different methods to explain larger parts of the missing heritability while facilitating the process of conducting a GWAS.	7
2.1	Illustration of the least squared estimator. Here, we try to minimise the sum of the squared training error, that is the sum of the distances between the observed data points (magenta points) and the predicted data points (blue crosses).	15
2.2	Probability of making at least one type-1 error. Here, we show the probability of making at least one type-1 error when testing m multiple hypothesis with respect to three different significance thresholds α	24
2.3	FWER vs. FDR vs. no correction. The further left a method the higher the probability of making more type-2 errors, whereas the further right the more type-1 errors.	25
2.4	Fixed effect model. The Fixed Effect Model assumes a common true effect of the genetic marker i across all studies k . The circle represents the true effect for study j . The square is the observed effect for study j and ϵ_{ij} is the random noise (sampling error) of the genetic marker i for study j . σ_{ij}^2 is the variance of the genetic marker i for study j . The triangle is the estimated common true effect across all k studies.	29
2.5	Random effect model. The Random Effect Model assumes a within-study and between-study variance for genetic marker i across all studies k . The circle represents the true effect θ_{ij} for study j . The square is the observed effect for study j , ϵ_{ij} is the random noise (sampling error) and τ_{ij} the true variation in effect sizes of the genetic marker i for study j . σ_{ij}^2 is the variance of the genetic marker i for study j . The triangle is the estimated mean of the population effect across all k studies.	31
2.6	Layers and modules of easyGWASCore: easyGWASCore is structured into three main layers. The first layer contains core modules that are needed by a variety of algorithms and methods. The second layer contains modules that represent a general class of algorithms and methods that can be applied to solve many different problems. The last layer contains modules tailored to certain tasks that mostly use methods or algorithms from the first two layers.	34
2.7	Manhattan plot and QQ-plot for the phenotype <i>avrPphB</i>: Purple points on the Manhattan Plot (left) indicate that these SNPs are significantly associated after correcting for multiple hypothesis using Bonferroni (red dashed line). The QQ-Plot (right) compares the observed distribution of p-values against the expected distributions using the negative logarithm of the p-values.	43
2.8	Linkage disequilibrium plot for SNP at position 4146714 on Chr 1: The magenta point is the focal SNP. LD is illustrated using different colours. The lower part of the plots gives information about the minor allele frequency of each SNP, as well as the genes in this region. .	44

2.9	Performance analysis of data processing and algorithmic runtime: Performance analysis of four GWAS algorithms from <code>easyGWASCore</code> with respect to data handling and actual algorithmic runtime. The number of genetic markers as well as the number of samples were varied for each algorithm.	45
2.10	Runtime comparison between <code>easyGWASCore</code> and the individual tools: For each tool we measured the total runtime in seconds and compared it to the <code>easyGWASCore</code> implementation (blue). We varied the number of genetic markers as well as the number of samples.	46
3.1	Venn-diagram showing the overlap between all five benchmark datasets: <i>VariBenchSelected</i> (10266 variants) is the part of <i>VariBench</i> not overlapping with <i>HumVar</i> nor <i>ExoVar</i> . <i>predictSNPSelected</i> (16098 variants) is the part of <i>predictSNP</i> not overlapping with <i>HumVar</i> , <i>ExoVar</i> nor <i>VariBench</i> . <i>SwissVarSelected</i> (12729 variants) is the part of <i>SwissVar</i> that does not overlap with <i>HumVar</i> , <i>ExoVar</i> , <i>VariBench</i> , nor <i>predictSNP</i>	50
3.2	Predictive performance of 10 popular pathogenicity prediction tools over five datasets: Evaluation of the ten different pathogenicity prediction tools (by AUC) over five datasets. The hatched bars indicate potentially biased results, due to the overlap (or possible overlap) between the evaluation data and the data used (by tool developers) for training the prediction tool.	51
3.3	Evaluation of type 2 circularity: Evaluation of the ten different pathogenicity prediction tools (by AUC) over five datasets. The hatched bars indicate potentially biased results, due to the overlap (or possible overlap) between the evaluation data and the data used (by tool developers) for training the prediction tool. The dotted bars indicate that the tool is biased due to type 2 circularity.	52
3.4	Dataset compositions: (a) Protein perspective: proportion of proteins containing only neutral variants (“neutral-only”), only pathogenic variants (“pathogenic-only”), and both types of variants (“mixed”). (b) Variant perspective: proportions, of variants in each of the three categories of proteins.	53
3.5	Fractions of variants for each dataset: Fractions of variants containing various ratios of pathogenic-to-neutral variants, binned into increasingly narrow bins, approaching balanced proteins. The open interval]0.0, 1.0[contains all mixed proteins (as in Figure 3.4b).	54
3.6	Performance of ten pathogenicity prediction tools according to protein pathogenic-to-neutral variant ratio: Evaluation of tool performance on subsets of <i>VariBenchSelected</i> , <i>predictSNPSelected</i> and <i>SwissVarSelected</i> , defined according to the relative proportions of pathogenic and neutral variants in the proteins they contain. “Pure” indicates variants belonging to proteins containing only one class of variant. [x, y] indicate variants belonging to mixed proteins, containing a ratio of pathogenic-to-neutral variants between x and y.]0.0, 1.0[therefore indicates all mixed proteins (the ratios of 0.0 and 1.0 being excluded by the reversed brackets). While <code>FATHMM-W</code> performs well or excellently on variants belonging to pure proteins (<i>VariBenchSelected</i> and <i>predictSNPSelected</i>), it performs poorly on those belonging to mixed proteins.	55
3.7	Variants from the selected datasets that are in identical or similar proteins in the proxy training datasets <i>HumVar</i>/<i>ExoVar</i>: Percentage of pathogenic and neutral variants that can be found in identical or similar proteins in <i>HumVar</i> / <i>ExoVar</i> . The x-axis shows different similarities between the proteins in the Selected dataset and the proteins in <i>HumVar</i> / <i>ExoVar</i> . The y-axis is the percentage of variants that can be found in identical or similar proteins. . . .	56
3.8	Comparison of the performance of two meta-predictors (Logit and Condel) and their component tools, across five datasets: Bar heights reflect AUC for each tool and tool combination. <code>Logit</code> and <code>Condel</code> are meta-predictors combining <code>MASS</code> , <code>PP2</code> , and <code>SIFT</code> . The “+” versions of <code>Logit</code> and <code>Condel</code> also include <code>FATHMM-W</code> . While effective in prediction, <code>FATHMM-W</code> (alone and in the <code>Logit+</code> and <code>Condel+</code> meta-predictors) is optimistically biased due to type 2 circularity.	57
3.9	Performance according to protein pathogenic-to-neutral variant ratio: Evaluation of <code>Condel</code> , <code>Condel+</code> , <code>Logit</code> and <code>Logit+</code> on subsets of <i>VariBenchSelected</i> , <i>predictSNPSelected</i> and <i>SwissVarSelected</i> defined according to the relative proportions of pathogenic and neutral variants in the proteins they contain.	58
3.10	Linkage disequilibrium plots including pathogenicity predictions: LD plots for a selected number of significantly associated SNPs. LD plots are enriched with pathogenicity predictions. Upper triangles indicate missense variants predicted to be pathogenic, whereas lower triangles represent benign missense variants.	61

4.1	Schematics of the easyGWAS architecture: Illustration of the internal architecture of easyGWAS including the hybrid database model and different task queues. Communication between the web-application and queues is established via the RabbitMQ message passing server. Task queues can be distributed over different computing nodes. The hybrid database can be accessed from the web-application, as well as from the different task queues. Users can link their personal Dropbox account to easyGWAS to integrate large genotype datasets in easyGWAS (Dropbox and the Dropbox logo are trademarks of Dropbox, Inc.).	66
4.2	Data repository view: The data repository consists of a public and private data section. The greenish menu on the left side allows the user to access different types of information, such as information about integrated species or phenotypes. The blueish menu in the lower left part offers methods for data management, such as data up- and download.	67
4.3	Data organisation schematic: Illustration of the data organisation. A user can have several data bags. A data bag contains several type of information about the GWAS data, e.g. the species, integrated datasets with their samples, phenotypes and covariates.	67
4.4	Detailed sample view: For each sample easyGWAS provides additional meta-information. Meta-information can be added or changed dynamically. Text in red ellipses are brief descriptions about certain functions.	68
4.5	Detailed phenotype view: For each phenotype easyGWAS provides a detailed view about different types of meta-information. Meta-information can be added or changed dynamically. Several statistics are shown about the distribution of the phenotype. Text in red ellipses are brief descriptions about certain functions.	69
4.6	Data download view: Download manager to download publicly available data.	69
4.7	GWAS dataset upload view: Form to integrate a new GWAS dataset into easyGWAS . Data in PLINK format has to be stored in a single ZIP file and uploaded to the personal Dropbox account. easyGWAS can then fetch the data from the personal Dropbox account.	70
4.8	The GWAS centre: The easyGWAS GWAS centre is structured into three main areas: (1) methods for performing GWASs and meta-analyses, (2) analyses and management options for study results and (3) management and sharing options for study projects.	71
4.9	Transformation and normalisation view of the easyGWAS GWAS wizard: Data distributions and Shapiro-Wilk test for the selected phenotypes are shown. Different normalisation techniques can be applied to normalise the data. The Shapiro-Wilk test and histogram are updated dynamically for different normalisation functions.	72
4.10	Screenshot of the meta-analysis summary page: The meta-analysis summary page is the final view of the meta-analysis wizard. The user can check if everything is selected correctly and submit the analysis.	73
4.11	Temporary history view: All submitted and finished experiments are initially stored in a temporary list. This list is automatically cleaned after 48h. Experiments can be either deleted or stored permanently.	73
4.12	Save experiments permanently: Form to save experiments permanently. Experiments are always grouped into an existing or new project. The names of the experiments can be changed.	74
4.13	GWAS result view: On the left side a brief summary of the experiment is displayed together with information about the top 10 associated hits. The right shows dynamic zoomable Manhattan plots. Different multiple hypothesis correction methods are available, as well as different options to dynamically adjust the Manhattan plots. A new sub-menu is shown in the top of the result view to navigate through different result views.	75
4.14	GWAS result summary: Summary of the experiment results. Detailed overview about all selected data sources and parameters.	76
4.15	Manhattan plot and QQ-plot for the original phenotype 4W: Results for the original 4W phenotype using all 119 samples in the <i>AtPolyDB</i> dataset. The phenotype is box-cox transformed and a MAF filter of 5% was applied. Only one hit was found to be significantly associated after Bonferroni correction.	79
4.16	Manhattan plot and QQ-plot for the phenotype 4W-AtPolyDB: Results for the 4W-AtPolyDB phenotype using a subset of 40 samples on the <i>AtPolyDB</i> dataset. The phenotype is box-cox transformed and a MAF filter of 5% was applied.	80
4.17	Manhattan plot and QQ-plot for the phenotype 4W-1001: Results for the original 4W-1001 phenotype using a subset of 79 samples on the 1,001 Genomes dataset. The phenotype is box-cox transformed and a MAF filter of 5% was applied.	80
4.18	Manhattan plot of meta-analysis: Meta-analysis results combining the summary statistics for the GWASs on the phenotypes 4W-AtPolyDB and 4W-1001.	80

4.19	Zoomed gene annotation plots: Zoomed in Manhattan plots downloaded from <code>easyGWAS</code> . Two SNPs are associated with two well-known genes <code>SVP</code> and <code>DOG1</code>	81
5.1	Extended graph for s/t min-cut. Graph is extended with a source s and sink t node. The source is connected with all nodes which association term $c_p > \eta$ and the sink is connected with all nodes which association term $c_p < \eta$	89
5.2	Three types of biological networks: a) Genomic sequence network: genetic markers adjacent on the genomic sequence are connected to each other. b) Gene membership network: adjacent markers and markers near the same gene are connected. c) Gene-interaction network: adjacent markers, markers near the same gene and markers near interacting genes are connected.	91
5.3	Runtime comparison between SConES, univariate linear regression, ncLASSO and graphLASSO: The left panel shows the runtime from 100 to 25k SNPs. The right panel shows the runtime up to 200k SNPs. After, three weeks, <code>graphLASSO</code> and <code>ncLASSO</code> had not finished running for 50k SNPs. The accelerated version of <code>ncLASSO</code> ran out of memory for more than 150k SNPs.	94
5.4	Evaluation of SConES on simulated data: Power and false discovery rate (FDR) of <code>SConES</code> , compared to state-of-the-art <code>LASSO</code> algorithms and a baseline univariate linear regression, in three different data simulation scenarios. Best methods are closest to the upper-left corner. Numbers denote the number of markers selected by the method.	95
5.5	Cross-validated predictivity of SConES: Predictivity is measured as Pearson's squared correlation coefficient between the actual phenotype and the predicted phenotype by a ridge-regression over the selected markers, compared to that of <code>LASSO</code> , <code>groupLASSO</code> , and <code>ncLASSO</code> . Horizontal bars indicate cross-validated <code>BLUP</code> predictivity.	98
5.6	Two scenarios to generate an unified network: a) If only one network exists for the multi-task learning approach, than the original network is duplicated and new edges (dashed lines) between shared vertices are added. b) Two networks which share vertices but have different edges are given. The unified network contains new edges (dashed lines) between shared vertices.	101
5.7	Runtime comparison: Runtime with respect to changes in number of tasks under fixed regularisation parameters.	104
5.8	Parameter sensitivity: Feature selection performance with respect to changes in regularisation parameters η , λ and μ . Note that x-axes for λ and η have logarithmic scales. The effect of changes in μ are reported for various feature-sharing scenarios. Two parameters η and λ behave identically independently of the amount of true causal features shared by the tasks and corresponding plots are therefore not reported.	105
5.9	Feature selection performance for simulated data: <code>MCC</code> (left column) should be maximised and <code>MSE</code> (right column) should be minimised. <code>CR</code> : the ranking of correlations (baseline); <code>LA</code> : <code>LASSO</code> ; <code>EN</code> : <code>Elastic Net</code> ; <code>GL</code> : <code>groupLASSO</code> ; <code>GR</code> : <code>ncLASSO</code> ; <code>AG</code> : <code>ancLASSO</code> ; <code>SC</code> : <code>Multi-SConES</code>	106
5.10	Feature selection performance in two tasks for simulated data: Only a fraction of the causal features are shared between the tasks. <code>CR</code> : the ranking of correlations (baseline); <code>LA</code> : <code>LASSO</code> ; <code>EN</code> : <code>Elastic Net</code> ; <code>GL</code> : <code>groupLASSO</code> ; <code>GR</code> : <code>ncLASSO</code> ; <code>AG</code> : <code>ancLASSO</code> ; <code>SC</code> : <code>Multi-SConES</code>	107
5.11	Data processing and algorithmic runtime evaluation: We varied the number of genetic markers, the number of samples and network densities. Solid lines represent data processing time, whereas dashed lines represent algorithmic runtime.	109
5.12	Runtime comparison between different implementations: Algorithmic runtime comparison with fixed parameters between different implementations of <code>SConES</code> . Experiments not finished after more than 20 days are truncated.	110
5.13	Runtime comparison of SConES including a grid-search: Comparison between the overall runtime of <code>easyGWASCore</code> and the original implementations in <code>Matlab</code> and <code>R</code> . Runtime includes data processing and the cross-validation with an internal grid-search.	110
5.14	Runtime comparison of Mutli-SConES: Runtime comparison of <code>Mutli-SConES</code> between <code>easyGWASCore</code> and <code>R</code> . Number of tasks are varied between one and 4. Computations are truncated after 20 days of runtime.	111
6.1	Illustration of diallel crossing schemes: a) In a full diallel crossing scheme, parents p are crossed in all possible combinations with each other. A total of p^2 crosses are possible. b) In a half diallel crossing scheme, parents are crossed excluding self-crosses and reciprocal crosses.	115
6.2	Illustration of experimental crossing scheme: Half diallel crosses (435 hybrids) including self-crosses between parents (30) and self-fertilisations of parents (30).	116
6.3	Cumulative distribution of pattern occurrences: Shows the number cumulative number of SNPs for different categories.	117
6.4	Distribution of SNPs within the first 20 categories: Distribution of SNPs across chromosomes for 20 different pattern occurrence categories.	117

6.5	Illustration of phenotypic components: Dominance deviation of a hybrid phenotype is the distance of the hybrid phenotype from the mid-parent value. The additive phenotypic component a is calculated as half the distance between the two parental phenotypes.	118
6.6	Dominance deviation histograms: Histograms show the distribution of the dominance deviation for all 10 phenotypes.	119
6.7	Heritability estimates: Broad sense H_b^2 and narrow sense h_n^2 heritability estimates including standard errors for all 10 phenotypes.	123
6.8	Manhattan plot and QQ-plot for estimated mean phenotypes (overdominant model): Manhattan and QQ-plots for three phenotypes with significantly associated SNPs when using an overdominant model and the estimated mean phenotypes. Two Bonferroni thresholds are shown. The standard one using all SNPs (red dashed line) and the stringent one (blue dashed line) correcting for 3 experiments (additive, overdominant mean and overdominant dominance deviation). Magenta points are significantly associated SNPs.	125
6.9	Manhattan plot and QQ-plot for dominance deviations (overdominant model): Manhattan and QQ-plots for three phenotypes with significantly associated SNPs when using an overdominant model and the derived dominance deviations d . Two Bonferroni thresholds are shown. The standard one using all SNPs (red dashed line) and the stringent one (blue dashed line) correcting for 3 experiments (additive, overdominant mean and overdominant dominance deviation). Magenta points are significantly associated SNPs.	126
6.10	Linkage disequilibrium plots: LD plots for significantly associated hits using an overdominant model and the dominance deviation of the phenotype <i>LTF</i>	127
6.11	Variance explained and broad sense heritability estimates: Variance Explained by all SNPs, all significant hits and <i>SConES</i> . Numbers above bars indicate the number of associated hits used to determine variance explained.	128
6.12	Power analyses of simulated phenotypes: a) Additive genotype encoding and simulated additive phenotype. b) Additive genotype encoding and simulated overdominant phenotype. c) Overdominant genotype encoding and simulated additive phenotype. d) Overdominant genotype encoding and simulated overdominant phenotype.	128
C.1	Illustration of minor allele frequency. There genetic markers are illustrated together with the minor allele and the minor allele frequency (MAF).	143

List of Tables

3.1	Overview of all benchmark datasets: These preprocessed and filtered datasets are used to evaluate the performance of different prediction tools.	49
3.2	Protein categories and variants per category: Overview about the total number of proteins per dataset and the composition of these datasets.	53
4.1	Available transformation methods: Overview of different methods to transform phenotypes. For each transformation method certain constrains are listed. The GWAS wizard determines on-the-fly which transformation method could be applied to which phenotype.	70
4.2	Available algorithms: Overview of all available algorithms in easyGWAS . The columns <i>Homozygous</i> and <i>Heterozygous</i> indicate whether the algorithm can be used with homozygous or heterozygous data, respectively. The columns <i>Binary</i> , <i>Continuous</i> and <i>Categorical</i> indicate whether the algorithm supports binary, continuous or categorical phenotypes. Additionally, the column <i>Covariates</i> indicates whether covariates can be added to the model. \triangle means that the model can be used with that type of data but it is not recommended.	71
4.3	Available meta-analysis algorithms: Overview of all available meta-analyses algorithms in easyGWAS . The columns <i>P-Value</i> and <i>Effect Size</i> indicate whether the algorithm needs p-values or effect sizes as input.	72
5.1	Comparison of tools: F-scores of SConES , compared to state-of-the-art LASSO algorithms and a baseline univariate linear regression (LR), in six different data simulation scenarios: The true causal SNPs are (i) randomly distributed; (ii) adjacent on the genomic sequence; (iii) near the same gene; (iv) near either of the same 2 connected genes; (v) near either of the same 3 connected genes; (vi) near either of the same 5 connected genes. Best performance in bold and second best in italics. <i>GS</i> : Genomic sequence network. <i>GM</i> : Gene membership network. <i>GI</i> : Gene interaction network.	96
5.2	Effect of removing network edges: Effect on the F-scores of SConES of removing a small fraction of the network edges. Results reported for SConES + GM in three different scenarios: The true causal markers are (ii) adjacent on the genomic sequence; (iii) near the same gene; (vi) near either of the same five connected genes.	96
5.3	Associations close to known candidate genes: Associations detected close to known candidate genes, for all flowering time phenotypes of <i>Arabidopsis thaliana</i> . We report the number of selected markers near candidate genes, followed by the total number of selected markers. Largest ratio in bold. <i>GS</i> : Genomic sequence network. <i>GM</i> : Gene membership network. <i>GI</i> : Gene interaction network.	99
5.4	Summary statistics: Averaged over the <i>Arabidopsis thaliana</i> flowering time phenotypes: average total number of selected markers (“#Markers”), average proportion of selected markers near candidate genes (“Near Candidate Genes”) and average number of different candidate genes recovered (“Candidate Genes Hit”). <i>GS</i> : Genomic sequence network. <i>GM</i> : Gene membership network. <i>GI</i> : Gene interaction network.	99
5.5	Fraction of markers deemed significantly associated with the phenotype: Comparison of markers identified by a LMM run on the full dataset to that selected by the other methods. We only report the phenotypes for which EMMAX returned at least one significant marker.	100

5.6	Results for multi-locus and multi-trait mapping: Results for different methods using between one and three correlated phenotypes. In case of more than one phenotype the multi-task version of the algorithm is used.	108
B.1	Confusion matrix	141
D.1	utils Module: Contains several helper methods for various different general tasks. Only the most important methods are shown.	145
D.2	kernel Module: Contains Kernel related methods.	145
D.3	optimiser Module: Contains different optimisation methods.	145
D.4	stats Module: Contains several statistical helper functions as well as different distribution function. Only the most important methods are shown.	146
D.5	io Module: Different classes for data input/output. Only the most important methods are shown.	146
D.6	regression Module: Contains different regression based methods. Only the most important methods are shown.	147
D.7	meta Module: Contains different meta-analysis methods. Only the most important methods are shown.	147
D.8	gwas Module for single trait GWASs: Contains different classes for single trait GWASs. Only the most important methods are shown.	148
D.9	gwas Module for multi trait GWASs: Contains different classes for SConES . Only the most important methods are shown.	149

Bibliography

- 1000 Genomes Project Consortium et al. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073, 2010.
- Richard J Abbott and Mioco F Gomes. Population genetic structure and outcrossing rate of *arabidopsis thaliana* (l.) heynh. *Heredity*, 62(3):411–418, 1989.
- Hervé Abdi. Bonferroni and sidak corrections for multiple comparisons. In *Encyclopedia of Measurement and Statistics.*, pages 103–107. Sage, 2007.
- Panagiotis Achlioptas, Bernhard Schölkopf, and Karsten Borgwardt. Two-locus association mapping in subquadratic time. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 726–734. ACM, 2011.
- Ivan A Adzhubei, Steffen Schmidt, Leonid Peshkin, Vasily E Ramensky, Anna Gerasimova, Peer Bork, Alexey S Kondrashov, and Shamil R Sunyaev. A method and server for predicting damaging missense mutations. *Nature methods*, 7(4):248–249, 2010.
- Ken Aho, DeWayne Derryberry, and Teri Peterson. Model selection for ecologists: the worldviews of aic and bic. *Ecology*, 95(3):631–636, 2014.
- Rie Kubota Ando and Tong Zhang. Learning on graph with laplacian regularization. *Advances in neural information processing systems*, 19:25, 2007.
- Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1): 25–29, 2000.
- Susanna Atwell, Yu S Huang, Bjarni J Vilhjálmsson, Glenda Willems, Matthew Horton, Yan Li, Dazhe Meng, Alexander Platt, Aaron M Tarone, Tina T Hu, et al. Genome-wide association study of 107 phenotypes in *arabidopsis thaliana* inbred lines. *Nature*, 465(7298):627–631, 2010.

- Yurii S Aulchenko, Stephan Ripke, Aaron Isaacs, and Cornelia M Van Duijn. GenABEL: an R library for genome-wide association analysis. *Bioinformatics*, 23(10):1294–1296, 2007.
- Chloé-Agathe Azencott, **Dominik Grimm**, Mahito Sugiyama, Yoshinobu Kawahara, and Karsten M Borgwardt. Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics*, 29(13):i171–i179, 2013.
- Erica G Bakker, Eli A Stahl, Christopher Toomajian, M Nordborg, M Kreitman, and J Bergelson. Distribution of genetic variation within and among local populations of *Arabidopsis thaliana* over its species range. *Molecular Ecology*, 15(5):1405–1418, 2006.
- David J Balding and Richard A Nichols. A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. In *Human Identification: The Use of DNA Markers*, pages 3–12. Springer, 1995.
- Vinay Kumar Baranwal, Venugopal Mikkilineni, Usha Barwale Zehr, Akhilesh K Tyagi, and Sanjay Kapoor. Heterosis: emerging ideas about hybrid vigour. *Journal of experimental botany*, 63(18):6309–6314, 2012.
- Jeffrey C Barrett, B Fry, JDMJ Maller, and Mark J Daly. Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, 21(2):263–265, 2005.
- Jonathan Baxter. A model of inductive bias learning. *J. Artif. Intell. Res.(JAIR)*, 12: 149–198, 2000.
- Jaroslav Bendl, Jan Stourac, Ondrej Salanda, Antonin Pavelka, Eric D Wieben, Jaroslav Zendulka, Jan Brezovsky, and Jiri Damborsky. Predictsnp: robust and accurate consensus classifier for prediction of disease-related mutations. *PLoS computational biology*, 10(1):e1003440, 2014.
- Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.
- Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.
- Joy Bergelson, Eli Stahl, Scott Dudek, and Martin Kreitman. Genetic variation within and among populations of *Arabidopsis thaliana*. *Genetics*, 148(3):1311–1323, 1998.
- Kirsten Bomblies, Levi Yant, Roosa A Laitinen, Sang-Tae Kim, Jesse D Hollister, Norman Warthmann, Joffrey Fitz, and Detlef Weigel. Local-scale patterns of genetic variability, outcrossing, and spatial structure in natural stands of *Arabidopsis thaliana*. *PLoS Genetics*, 6(3):e1000890, 2010.

- Michael Borenstein, Larry V Hedges, Julian Higgins, and Hannah R Rothstein. A basic introduction to fixed-effect and random-effects models for meta-analysis. *Research Synthesis Methods*, 1(2):97–111, 2010.
- Michael Borenstein, Larry V Hedges, Julian PT Higgins, and Hannah R Rothstein. *Introduction to meta-analysis*. John Wiley & Sons, 2011.
- George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.
- Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.
- Benjamin Brachi, Nathalie Faure, Matt Horton, Emilie Flahauw, Adeline Vazquez, Magnus Nordborg, Joy Bergelson, Joel Cuguen, and Fabrice Roux. Linkage and association mapping of arabidopsis thaliana flowering time in nature. *PLoS Genetics*, 6(5):e1000940, 2010.
- Richard P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425, 1971.
- AB Bruce. The mendelian theory of heredity and the augmentation of vigor. *Science*, pages 627–628, 1910.
- Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using diamond. *Nature methods*, 2014.
- Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 2015.
- Brendan Bulik-Sullivan, Po-Ru Loh, Hilary Finucane, Stephan Ripke, Jian Yang, Nick Patterson, Mark J Daly, Alkes L Price, Benjamin M Neale, Schizophrenia Working Group Psychiatric Genomics Consortium, et al. Ld score regression distinguishes confounding from polygenicity in genome-wide association studies. *bioRxiv*, page 002931, 2014.
- Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2002.
- Paul R Burton, David G Clayton, Lon R Cardon, Nick Craddock, Panos Deloukas, Audrey Duncanson, Dominic P Kwiatkowski, Mark I McCarthy, Willem H Ouwehand, Nilesh J Samani, et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, 2007.
- William S Bush and Jason H Moore. Genome-wide association studies. *PLoS computational biology*, 8(12):e1002822, 2012.

- Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009.
- Rita M Cantor, Kenneth Lange, and Janet S Sinsheimer. Prioritizing gwas results: a review of statistical methods and recommendations for their application. *The American Journal of Human Genetics*, 86(1):6–22, 2010.
- Jun Cao, Korbinian Schneeberger, Stephan Ossowski, Torsten Günther, Sebastian Bender, Joffrey Fitz, Daniel Koenig, Christa Lanz, Oliver Stegle, Christoph Lipfert, et al. Whole-genome sequencing of multiple arabidopsis thaliana populations. *Nature genetics*, 43(10):956–963, 2011.
- R Caruna. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 41–48, 1993.
- Eunyoung Chae, Kirsten Bomblies, Sang-Tae Kim, Darya Karelina, Maricris Zaidem, Stephan Ossowski, Carmen Martín-Pizarro, Roosa AE Laitinen, Beth A Rowan, Hezi Tenenboim, et al. Species-wide genetic incompatibility analysis identifies immune genes as hot spots of deleterious epistasis. *Cell*, 159(6):1341–1351, 2014.
- Daniel I Chasman, Markus Schürks, Verner Anttila, Boukje de Vries, Ulf Schminke, Lenore J Launer, Gisela M Terwindt, Arn MJM van den Maagdenberg, Konstanze Fendrich, Henry Völzke, et al. Genome-wide association study reveals three susceptibility loci for common migraine in the general population. *Nature genetics*, 43(7):695–698, 2011.
- George CK Chiang, Deepak Barua, Emily Dittmar, Elena M Kramer, Rafael Rubio de Casas, and Kathleen Donohue. Pleiotropy in the wild: the dormancy gene *dog1* exerts cascading control on life cycles. *Evolution*, 67(3):883–893, 2013.
- Liam H Childs, Jan Liseč, and Dirk Walther. Matapax: An online high-throughput genome-wide association study pipeline. *Plant physiology*, 158(4):1534–1541, 2012.
- Seoae Cho, Kyunga Kim, Young Jin Kim, Jong-Keuk Lee, Yoon Shin Cho, Jong-Young Lee, Bok-Ghee Han, Heebal Kim, Jurg Ott, and Taesung Park. Joint identification of multiple genetic variants via elastic-net variable selection in a genome-wide association analysis. *Annals of human genetics*, 74(5):416–428, 2010.
- BR Christie and VI Shattuck. The diallel cross: design, analysis and use for plant breeders. *Plant breeding reviews*, 9:9–36, 1992.
- Sung Chun and Justin C Fay. Identification of deleterious mutations within three human genomes. *Genome research*, 19(9):1553–1561, 2009.
- Pablo Cingolani, Adrian Platts, Le Lily Wang, Melissa Coon, Tung Nguyen, Luan Wang, Susan J Land, Xiangyi Lu, and Douglas M Ruden. A program for annotating

- and predicting the effects of single nucleotide polymorphisms, snpeff: Snps in the genome of drosophila melanogaster strain w1118; iso-2; iso-3. *Fly*, 6(2):80–92, 2012.
- Gregory M Cooper and Jay Shendure. Needles in stacks of needles: finding disease-causal variants in a wealth of genomic data. *Nature Reviews Genetics*, 12(9):628–640, 2011.
- Chas. B. Davenport. Degeneration, albinism and inbreeding. *Science*, pages 454–455, 1908.
- Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.
- Eugene V Davydov, David L Goode, Marina Sirota, Gregory M Cooper, Arend Sidow, and Serafim Batzoglou. Identifying a high fraction of the human genome to be under selective constraint using *gerp++*. *PLoS computational biology*, 6(12):e1001025, 2010.
- Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo del Angel, Manuel A Rivas, Matt Hanna, et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491–498, 2011.
- Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled clinical trials*, 7(3):177–188, 1986.
- B Devlin and Kathryn Roeder. Genomic control for association studies. *Biometrics*, 55(4):997–1004, 1999.
- Edward M East. Inbreeding in corn. *Rep Conn Agric Exp Stn*, 1907:419–428, 1908.
- Peter Elias, Amiel Feinstein, and Claude E Shannon. A note on the maximum flow through a network. *Information Theory, IRE Transactions on*, 2(4):117–119, 1956.
- Evangelos Evangelou and John PA Ioannidis. Meta-analysis methods for genome-wide association studies and beyond. *Nature Reviews Genetics*, 14(6):379–389, 2013.
- Warren J Ewens and Gregory Grant. Gene expression, microarrays, and multiple testing. *Statistical Methods in Bioinformatics: An Introduction*, pages 430–474, 2005.
- Laurent Excoffier and Montgomery Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular biology and evolution*, 12(5):921–927, 1995.

- Ludwig Fahrmeir, Thomas Kneib, Stefan Lang, and Brian Marx. *Regression: Models, methods and applications*. Springer Science & Business Media, 2013.
- DS Falconer and TFC Mackay. Introduction to quantitative genetics. *Longman*, 19(8): 1, 1995.
- Aasa Feragen, Jens Petersen, Dominik Grimm, Asger Dirksen, Jesper Holst Pedersen, Karsten Borgwardt, and Marleen de Bruijne. Geometric tree kernels: Classification of copd from airway tree geometry. *Information Processing in Medical Imaging - IPMI 2013*, 2013.
- Daniele L Filiault and Julin N Maloof. A genome-wide association study identifies variants underlying the arabidopsis thaliana shade avoidance response. *PLoS genetics*, 8(3):e1002589, 2012.
- Ronald Aylmer Fisher. Statistical methods for research workers. *Biological monographs and manuals*, 1934.
- Jonathan Flint and Eleazar Eskin. Genome-wide association studies in mice. *Nature Reviews Genetics*, 13(11):807–817, 2012.
- Lester R Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.
- Andre Franke, Dermot PB McGovern, Jeffrey C Barrett, Kai Wang, Graham L Radford-Smith, Tariq Ahmad, Charlie W Lees, Tobias Balschun, James Lee, Rebecca Roberts, et al. Genome-wide meta-analysis increases to 71 the number of confirmed crohn’s disease susceptibility loci. *Nature genetics*, 42(12):1118–1125, 2010.
- Tobias Freilinger, Verner Anttila, Boukje de Vries, Rainer Malik, Mikko Kallela, Gisela M Terwindt, Patricia Pozo-Rosich, Bendik Winsvold, Dale R Nyholt, Willebrordus PJ van Oosterhout, et al. Genome-wide association analysis identifies susceptibility loci for migraine without aura. *Nature genetics*, 44(7):777–782, 2012.
- Brooke L Fridley and Joanna M Biernacka. Gene set analysis of snp data: benefits, challenges, and future directions. *European Journal of Human Genetics*, 19(8):837–843, 2011.
- NEG Gilbert. Diallel cross in plant breeding. *Heredity*, 12(3):477–492, 1958.
- Andrew V Goldberg and Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.
- Abel González-Pérez and Nuria López-Bigas. Improving the assessment of the outcome of nonsynonymous snvs with a consensus deleteriousness score, condel. *The American Journal of Human Genetics*, 88(4):440–449, 2011.

- Julian Gough, Kevin Karplus, Richard Hughey, and Cyrus Chothia. Assignment of homology to genome sequences using a library of hidden markov models that represent all proteins of known structure. *Journal of molecular biology*, 313(4):903–919, 2001.
- Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.
- Dominik Grimm**, Jörg Hagmann, Daniel Koenig, Detlef Weigel, and Karsten Borgwardt. Accurate indel prediction using paired-end short reads. *BMC genomics*, 14(1):132, 2013.
- Dominik G Grimm** and Karsten M Borgwardt. A C/C++ Framework with Python Interfaces for Genome-Wide Association Studies. *Unpublished*, 2015.
- Dominik G Grimm**, Bastian Greshake, Stefan Kleeberger, Christoph Lippert, Oliver Stegle, Bernhard Schölkopf, Detlef Weigel, and Karsten M Borgwardt. easyGWAS: An integrated interspecies platform for performing genome-wide association studies. *arXiv preprint arXiv:1212.4788*, 2012.
- Dominik G Grimm**, Chloé-Agathe Azencott, Fabian Aicheler, Udo Gieraths, Daniel G MacArthur, Kaitlin E Samocha, David N Cooper, Peter D Stenson, Mark J Daly, Jordan W Smoller, Laramie E Duncan, and Karsten M Borgwardt. The Evaluation of Tools Used to Predict the Impact of Missense Variants Is Hindered by Two Types of Circularity. *Human mutation*, 36(5):513–523, 2015.
- Daniel F Gudbjartsson, G Bragi Walters, Gudmar Thorleifsson, Hreinn Stefansson, Bjarni V Halldorsson, Pasha Zusmanovich, Patrick Sulem, Steinunn Thorlacius, Arnaldur Gylfason, Stacy Steinberg, et al. Many sequence variants affecting diversity of adult human height. *Nature genetics*, 40(5):609–615, 2008.
- Gaël Guennebaud, Benoit Jacob, et al. Eigen v3, 2010.
- Melissa Gymrek, Amy L McGuire, David Golan, Eran Halperin, and Yaniv Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.
- Susan T Harbison, Akihiko H Yamamoto, Juan J Fanara, Koenraad K Norga, and Trudy FC Mackay. Quantitative trait loci affecting starvation resistance in drosophila melanogaster. *Genetics*, 166(4):1807–1823, 2004.
- Daniel L Hartl, Andrew G Clark, and Andrew G Clark. *Principles of population genetics*, volume 116. Sinauer associates Sunderland, 1997.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Linear methods for regression. In *The Elements of Statistical Learning*, pages 43–99. Springer, 2009a.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Model assessment and selection. In *The elements of statistical learning*, pages 219–259. Springer, 2009b.

- Ben John Hayes, Peter M Visscher, and Michael E Goddard. Increased accuracy of artificial selection by using the realized relationship matrix. *Genetics Research*, 91(01):47–60, 2009.
- Gibran Hemani, Athanasios Theocharidis, Wenhua Wei, and Chris Haley. Epigpu: exhaustive pairwise epistasis scans parallelized on consumer level graphics cards. *Bioinformatics*, 27(11):1462–1465, 2011.
- Charles R Henderson. Best linear unbiased estimation and prediction under a selection model. *Biometrics*, pages 423–447, 1975.
- Yosef Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, 1988.
- Yosef Hochberg and Yoav Benjamini. More powerful procedures for multiple significance testing. *Statistics in medicine*, 9(7):811–818, 1990.
- Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.
- Gerhard Hommel. A comparison of two modified bonferroni procedures. *Biometrika*, 76(3):624–625, 1989.
- Gerhard Hommel and Frank Krummenauer. Improvements and modifications of tarone’s multiple test procedure for discrete data. *Biometrics*, pages 673–681, 1998.
- Matthew W Horton, Angela M Hancock, Yu S Huang, Christopher Toomajian, Susanna Atwell, Adam Auton, N Wayan Muliyati, Alexander Platt, F Gianluca Sperone, Bjarni J Vilhjálmsson, et al. Genome-wide patterns of genetic variation in worldwide arabidopsis thaliana accessions from the regmap panel. *Nature genetics*, 44(2):212–216, 2012.
- Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.
- Tim JP Hubbard, Bronwen L Aken, S Ayling, Benoit Ballester, Kathryn Beal, Eugene Bragin, Simon Brent, Yuan Chen, Peter Clapham, Laura Clarke, et al. Ensembl 2009. *Nucleic acids research*, 37(suppl 1):D690–D697, 2009.
- International Human Genome Sequencing Consortium et al. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004.
- Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM, 2009.
- Randall C Johnson, George W Nelson, Jennifer L Troyer, James A Lautenberger, Bailey D Kessing, Cheryl A Winkler, and Stephen J O’Brien. Accounting for multiple

- comparisons in a genome-wide association study (gwas). *BMC genomics*, 11(1):724, 2010.
- Donald F Jones. Dominance of linked factors as a means of accounting for heterosis. *Genetics*, 2(5):466, 1917.
- Katherine W Jordan, Mary Anna Carbone, Akihiko Yamamoto, Theodore J Morgan, and Trudy FC Mackay. Quantitative genomics of locomotor behavior in *drosophila melanogaster*. *Genome Biol*, 8(8):R172, 2007.
- Tony Kam-Thong, Darina Czamara, Koji Tsuda, Karsten Borgwardt, Cathryn M Lewis, Angelika Erhardt-Lehmann, Bernhard Hemmer, Peter Rieckmann, Markus Daake, Frank Weber, et al. Epiblaster-fast exhaustive two-locus epistasis detection strategy using graphical processing units. *European Journal of Human Genetics*, 19(4):465–471, 2011.
- Tony Kam-Thong, Chloé-Agathe Azencott, Lawrence Cayton, B Ptz, André Altman, Nazanin Karbalai, PG Smann, B Schlkopf, B Mller-Myhsok, and Karsten M Borgwardt. Glide: Gpu-based linear regression for detection of epistasis. *Human heredity*, 73(4):220, 2012.
- Hyun Min Kang, Noah A Zaitlen, Claire M Wade, Andrew Kirby, David Heckerman, Mark J Daly, and Eleazar Eskin. Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709–1723, 2008.
- Hyun Min Kang, Jae Hoon Sul, Susan K Service, Noah A Zaitlen, Sit-yeek Kong, Nelson B Freimer, Chiara Sabatti, Eleazar Eskin, et al. Variance component model to account for sample structure in genome-wide association studies. *Nature genetics*, 42(4):348–354, 2010.
- Theofanis Karaletsos, Oliver Stegle, Christine Dreyer, John Winn, and Karsten M Borgwardt. Shapepheno: unsupervised extraction of shape phenotypes from biological image collections. *Bioinformatics*, 28(7):1001–1008, 2012.
- Seyoung Kim, Kyung-Ah Sohn, and Eric P Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, 2009.
- Andrew Kirby, Hyun Min Kang, Claire M Wade, Chris Cotsapas, Emrah Kostem, Buhm Han, Nick Furlotte, Eun Yong Kang, Manuel Rivas, Molly A Bogue, et al. Fine mapping in 94 inbred mouse strains using a high-density haplotype resource. *Genetics*, 185(3):1081–1095, 2010.
- Martin Kircher, Daniela M Witten, Preti Jain, Brian J O’Roak, Gregory M Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature genetics*, 46(3):310–315, 2014.

- Arthur Korte, Bjarni J Vilhjálmsson, Vincent Segura, Alexander Platt, Quan Long, and Magnus Nordborg. A mixed-model approach for genome-wide association studies of correlated traits in structured populations. *Nature genetics*, 44(9):1066–1071, 2012.
- Ludmila I Kuncheva. A stability index for feature selection. In *Artificial intelligence and applications*, pages 421–427, 2007.
- Heungsoon Felix Lee and Daniel R Dooly. Algorithms for the constrained maximum-weight connected graph problem. *Naval Research Logistics (NRL)*, 43(7):985–1008, 1996.
- Jeong Hwan Lee, Seong Jeon Yoo, Soo Hyun Park, Ildoo Hwang, Jong Seob Lee, and Ji Hoon Ahn. Role of *svp* in the control of flowering time by ambient temperature in arabidopsis. *Genes & Development*, 21(4):397–402, 2007.
- Seunghak Lee, Elango Cheran, and Michael Brudno. A robust framework for detecting structural variations in a genome. *Bioinformatics*, 24(13):i59–i67, 2008.
- Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient l_1 regularized logistic regression. *Proceedings of the National Conference on Artificial Intelligence*, 21(1):401, 2006.
- Guillaume Lettre, Anne U Jackson, Christian Gieger, Fredrick R Schumacher, Sonja I Berndt, Serena Sanna, Susana Eyheramendy, Benjamin F Voight, Johannah L Butler, Candace Guiducci, et al. Identification of ten loci associated with height highlights new biological pathways in human growth. *Nature genetics*, 40(5):584–591, 2008.
- Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.
- Caiyan Li and Hongzhe Li. Variable selection and regression analysis for graph-structured covariates with an application to genomics. *The annals of applied statistics*, 4(3):1498, 2010.
- Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.
- Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009a.
- Lanzhi Li, Kaiyang Lu, Zhaoming Chen, Tongmin Mu, Zhongli Hu, and Xinqi Li. Dominance, overdominance and epistasis condition the heterosis in two heterotic rice hybrids. *Genetics*, 180(3):1725–1742, 2008.

- Miao-Xin Li, Johnny SH Kwan, Su-Ying Bao, Wanling Yang, Shu-Leong Ho, Yong-Qiang Song, and Pak C Sham. Predicting mendelian disease-causing non-synonymous single nucleotide variants in exome sequencing studies. *PLoS genetics*, 9(1):e1003143, 2013.
- Ruiqiang Li, Yingrui Li, Xiaodong Fang, Huanming Yang, Jian Wang, Karsten Kristiansen, and Jun Wang. Snp detection for massively parallel whole-genome resequencing. *Genome research*, 19(6):1124–1132, 2009b.
- Ruiqiang Li, Chang Yu, Yingrui Li, Tak-Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. Soap2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966–1967, 2009c.
- Zhi-Kang Li, LJ Luo, HW Mei, DL Wang, QY Shu, R Tabien, DB Zhong, CS Ying, JW Stansel, GS Khush, et al. Overdominant epistatic loci are the primary genetic basis of inbreeding depression and heterosis in rice. i. biomass and grain yield. *Genetics*, 158(4):1737–1753, 2001.
- Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman. Fast linear mixed models for genome-wide association studies. *Nature Methods*, 8(10):833–835, 2011.
- Christoph Lippert, Jennifer Listgarten, Robert I Davidson, Jeff Baxter, Hoifung Poon, Carl M Kadie, and David Heckerman. An exhaustive epistatic snp association analysis on expanded wellcome trust data. *Scientific reports*, 3, 2013.
- Jennifer Listgarten, Carl Kadie, Eric E Schadt, and David Heckerman. Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences*, 107(38):16465–16470, 2010.
- Jennifer Listgarten, Christoph Lippert, Carl M Kadie, Robert I Davidson, Eleazar Eskin, and David Heckerman. Improved linear mixed models for genome-wide association studies. *Nature Methods*, 9(6):525–526, 2012.
- Jennifer Listgarten, Christoph Lippert, and David Heckerman. Fast-lmm-select for addressing confounding from spatial structure and rare variants. *Nature genetics*, 45(5):470–471, 2013.
- Jun Liu, Shuiwang Ji, and Jieping Ye. Slep: Sparse learning with efficient projections. *Arizona State University*, 6:491, 2009.
- Felipe Llinares-López, **Dominik G. Grimm**, Dean A. Bodenham, Udo Gieraths, Mahito Sugiyama, Beth Rowan, and Karsten Borgwardt. Genome-wide detection of intervals of genetic heterogeneity associated with complex traits. *Bioinformatics*, 2015a.
- Felipe Llinares-López, Mahito Sugiyama, Laetitia Papaxanthos, and Karsten M Borgwardt. Fast and memory-efficient significant pattern mining via permutation

- testing. *Proceedings of the 21st ACM SIGKDD international conference on Knowledge discovery and data mining*, 2015b.
- Po-Ru Loh, George Tucker, Brendan K Bulik-Sullivan, Bjarni J Vilhjalmsson, Hilary K Finucane, Daniel I Chasman, Paul M Ridker, Benjamin M Neale, Bonnie Berger, Nick Patterson, et al. Efficient bayesian mixed model analysis increases association power in large cohorts. *Nature Genetics*, 47(3):284–290, March 2015. ISSN 1061-4036. doi: 10.1038/ng.3190. URL <http://www.nature.com/ng/journal/v47/n3/full/ng.3190.html>.
- Quan Long, Fernando A Rabanal, Dazhe Meng, Christian D Huber, Ashley Farlow, Alexander Platzer, Qingrun Zhang, Bjarni J Vilhjalmsson, Arthur Korte, Viktoria Nizhynska, et al. Massive genomic variation and strong selection in arabidopsis thaliana lines from sweden. *Nature Genetics*, 45(8):884–890, 2013.
- LJ Luo, Z-K Lia, HW Mei, QY Shu, R Tabien, DB Zhong, CS Ying, JW Stansel, GS Khush, and AH Paterson. Overdominant epistatic loci are the primary genetic basis of inbreeding depression and heterosis in rice. ii. grain yield components. *Genetics*, 158(4):1755–1771, 2001.
- Michael Lynch, Bruce Walsh, et al. *Genetics and analysis of quantitative traits*, volume 1. Sinauer Sunderland, 1998.
- Trudy FC Mackay, Stephen Richards, Eric A Stone, Antonio Barbadilla, Julien F Ayroles, Dianhui Zhu, Sònia Casillas, Yi Han, Michael M Magwire, Julie M Cridland, et al. The drosophila melanogaster genetic reference panel. *Nature*, 482(7384):173–178, 2012.
- Reedik Mägi and Andrew P Morris. Gwama: software for genome-wide association meta-analysis. *BMC bioinformatics*, 11(1):288, 2010.
- Michele Magrane, UniProt Consortium, et al. Uniprot knowledgebase: a hub of integrated protein data. *Database*, 2011:bar009, 2011.
- Teri A Manolio. Bringing genome-wide association findings into clinical use. *Nature Reviews Genetics*, 14(8):549–558, 2013.
- Teri A Manolio, Francis S Collins, Nancy J Cox, David B Goldstein, Lucia A Hindorff, David J Hunter, Mark I McCarthy, Erin M Ramos, Lon R Cardon, Aravinda Chakravarti, et al. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, 2009.
- Jonathan Marchini, Peter Donnelly, and Lon R Cardon. Genome-wide strategies for detecting multiple loci that influence complex diseases. *Nature genetics*, 37(4):413–417, 2005.
- Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

- Mark I McCarthy, Gonçalo R Abecasis, Lon R Cardon, David B Goldstein, Julian Little, John PA Ioannidis, and Joel N Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature Reviews Genetics*, 9(5):356–369, 2008.
- Angela McGaughran, Christian Rödelsperger, **Dominik Grimm**, Jan M Meyer, Eduardo Moreno, Katy Morgan, Mark Leaver, Vahan Serobyanyan, Barbara Rakitsch, Tony Hyman, Karsten M Borgwardt, and Ralf J Sommer. Genome-wide profiles of diversification and genotype-phenotype association in nematodes from la réunion island. *Journal Publication Under Preparation*, 2015.
- Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, et al. The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome research*, 20(9):1297–1303, 2010.
- William McLaren, Bethan Pritchard, Daniel Rios, Yuan Chen, Paul Flicek, and Fiona Cunningham. Deriving the consequences of genomic variants with the ensembl api and snp effect predictor. *Bioinformatics*, 26(16):2069–2070, 2010.
- Paul Medvedev, Monica Stanciu, and Michael Brudno. Computational methods for discovering structural variation with next-generation sequencing. *Nature methods*, 6:S13–S20, 2009.
- Mónica Meijón, Santosh B Satbhai, Takashi Tsuchimatsu, and Wolfgang Busch. Genome-wide association study using cellular traits identifies a new regulator of root development in arabidopsis. *Nature genetics*, 46(1):77–81, 2014.
- Nicolai Meinshausen, Lukas Meier, and Peter Bühlmann. P-values for high-dimensional regression. *Journal of the American Statistical Association*, 2012.
- Michael L Metzker. Sequencing technologies—the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2010.
- TJ Morgan and TFC Mackay. Quantitative trait loci for thermotolerance phenotypes in drosophila melanogaster. *Heredity*, 96(3):232–242, 2006.
- Frederick Mosteller and R. A. Fisher. Questions and Answers. *The American Statistician*, 2(5):30, October 1948. ISSN 00031305. doi: 10.2307/2681650.
- Anaïs Mottaz, Fabrice PA David, Anne-Lise Veuthey, and Yum L Yip. Easy retrieval of single amino-acid polymorphisms and phenotype information using swissvar. *Bioinformatics*, 26(6):851–852, 2010.
- Preethy Sasidharan Nair and Mauno Vihinen. Varibench: a benchmark database for variations. *Human mutation*, 34(1):42–49, 2013.

- Mike A Nalls, Nathan Pankratz, Christina M Lill, Chuong B Do, Dena G Hernandez, Mohamad Saad, Anita L DeStefano, Eleanna Kara, Jose Bras, Manu Sharma, et al. Large-scale meta-analysis of genome-wide association data identifies six new risk loci for parkinson's disease. *Nature genetics*, 2014.
- Benjamin M Neale, Sarah E Medland, Stephan Ripke, Philip Asherson, Barbara Franke, Klaus-Peter Lesch, Stephen V Faraone, Thuy Trang Nguyen, Helmut Schäfer, Peter Holmans, et al. Meta-analysis of genome-wide association studies of attention-deficit/hyperactivity disorder. *Journal of the American Academy of Child & Adolescent Psychiatry*, 49(9):884–897, 2010.
- Pauline C Ng and Steven Henikoff. Sift: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13):3812–3814, 2003.
- John Novembre, Toby Johnson, Katarzyna Bryc, Zoltán Kutalik, Adam R Boyko, Adam Auton, Amit Indap, Karen S King, Sven Bergmann, Matthew R Nelson, et al. Genes mirror geography within europe. *Nature*, 456(7218):98–101, 2008.
- Guillaume R Obozinski, Martin J Wainwright, and Michael I Jordan. High-dimensional support union recovery in multivariate regression. In *Advances in Neural Information Processing Systems*, pages 1217–1224, 2008.
- Stephan Ossowski, Korbinian Schneeberger, Richard M Clark, Christa Lanz, Norman Warthmann, and Detlef Weigel. Sequencing of natural strains of arabidopsis thaliana with short reads. *Genome research*, 18(12):2024–2033, 2008.
- Marita Kruskopf Österberg, Oksana Shavorskaya, Martin Lascoux, and Ulf Lagercrantz. Naturally occurring indel variation in the brassica nigra coll gene is associated with variation in flowering time. *Genetics*, 161(1):299–306, 2002.
- Ju-Hyun Park, Sholom Wacholder, Mitchell H Gail, Ulrike Peters, Kevin B Jacobs, Stephen J Chanock, and Nilanjan Chatterjee. Estimation of effect size distribution from genome-wide association studies and implications for future discoveries. *Nature genetics*, 42(7):570–575, 2010.
- Paul DP Pharoah, Ya-Yu Tsai, Susan J Ramus, Catherine M Phelan, Ellen L Goode, Kate Lawrenson, Melissa Buckley, Brooke L Fridley, Jonathan P Tyrer, Howard Shen, et al. Gwas meta-analysis and replication identifies three new susceptibility loci for ovarian cancer. *Nature genetics*, 45(4):362–370, 2013.
- F Xavier Pico, Belén Méndez-Vigo, José M Martínez-Zapater, and Carlos Alonso-Blanco. Natural genetic variation of arabidopsis thaliana is geographically structured in the iberian peninsula. *Genetics*, 180(2):1009–1021, 2008.
- Sreekumar G Pillai, Dongliang Ge, Guohua Zhu, Xiangyang Kong, Kevin V Shianna, Anna C Need, Sheng Feng, Craig P Hersh, Per Bakke, Amund Gulsvik, et al. A genome-wide association study in chronic obstructive pulmonary disease (copd): identification of two major susceptibility loci. *PLoS genetics*, 5(3):e1000421, 2009.

- Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904–909, 2006.
- Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.
- Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt, and Oliver Stegle. It is all in the noise: Efficient multi-task gaussian process inference with structured residuals. In *Advances in Neural Information Processing Systems*, pages 1466–1474, 2013a.
- Barbara Rakitsch, Christoph Lippert, Oliver Stegle, and Karsten Borgwardt. A lasso multi-marker mixed model for association mapping with population structure correction. *Bioinformatics*, 29(2):206–214, 2013b.
- David E Reich, Michele Cargill, Stacey Bolk, James Ireland, Pardis C Sabeti, Daniel J Richter, Thomas Lavery, Rose Kouyoumjian, Shelli F Farhadian, Ryk Ward, et al. Linkage disequilibrium in the human genome. *Nature*, 411(6834):199–204, 2001.
- David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.
- Boris Reva, Yevgeniy Antipin, and Chris Sander. Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic acids research*, page gkr407, 2011.
- John D Rioux, Ramnik J Xavier, Kent D Taylor, Mark S Silverberg, Philippe Goyette, Alan Huett, Todd Green, Petric Kuballa, M Michael Barmada, Lisa Wu Datta, et al. Genome-wide association study identifies new susceptibility loci for crohn disease and implicates autophagy in disease pathogenesis. *Nature genetics*, 39(5):596–604, 2007.
- Stephan Ripke, Naomi R Wray, Cathryn M Lewis, Steven P Hamilton, Myrna M Weissman, Gerome Breen, Enda M Byrne, Douglas HR Blackwood, Dorret I Boomsma, Sven Cichon, et al. A mega-analysis of genome-wide association studies for major depressive disorder. *Molecular psychiatry*, 18(4):497–511, 2013.
- Alan R Rogers and Chad Huff. Linkage disequilibrium between loci with unknown phase. *Genetics*, 182(3):839–844, 2009.
- Schizophrenia Psychiatric Genome-Wide Association Study (GWAS) Consortium et al. Genome-wide association study identifies five new schizophrenia loci. *Nature genetics*, 43(10):969–976, 2011.

- Robert J Schmitz, Matthew D Schultz, Mark A Urich, Joseph R Nery, Mattia Pelizzola, Ondrej Libiger, Andrew Alix, Richard B McCosh, Huaming Chen, Nicholas J Schork, et al. Patterns of population epigenomic diversity. *Nature*, 495(7440):193–198, 2013.
- Jana Marie Schwarz, David N Cooper, Markus Schuelke, and Dominik Seelow. Mutationtaster2: mutation prediction for the deep-sequencing age. *Nature methods*, 11(4):361–362, 2014.
- Laura J Scott, Karen L Mohlke, Lori L Bonnycastle, Cristen J Willer, Yun Li, William L Duren, Michael R Erdos, Heather M Stringham, Peter S Chines, Anne U Jackson, et al. A genome-wide association study of type 2 diabetes in finns detects multiple susceptibility variants. *science*, 316(5829):1341–1345, 2007.
- Vincent Segura, Bjarni J Vilhjálmsson, Alexander Platt, Arthur Korte, Ümit Seren, Quan Long, and Magnus Nordborg. An efficient multi-locus mixed-model approach for genome-wide association studies in structured populations. *Nature genetics*, 44(7):825–830, 2012.
- Ku Chee Seng and Chia Kee Seng. The success of the genome-wide association approach: a brief story of a long struggle. *European Journal of Human Genetics*, 16(5):554–564, 2008.
- Ümit Seren, Bjarni J Vilhjálmsson, Matthew W Horton, Dazhe Meng, Petar Forai, Yu S Huang, Quan Long, Vincent Segura, and Magnus Nordborg. Gwapp: a web application for genome-wide association mapping in arabidopsis. *The Plant Cell Online*, 24(12):4793–4805, 2012.
- Danelle K. Seymour, Chae Eunyoung, **Dominik G. Grimm**, Carmen M. Pizzaro, François Vasseur, Barbara Rakitsch, Karsten M. Borgwardt, Daniel Koenig, and Detlef Weigel. The genetic architecture of non-additive hybrid phenotypes in *A. thaliana*. *In Preparation*, 2015.
- Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, pages 591–611, 1965.
- Hashem A Shihab, Julian Gough, David N Cooper, Peter D Stenson, Gary LA Barker, Keith J Edwards, Ian NM Day, and Tom R Gaunt. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden markov models. *Human mutation*, 34(1):57–65, 2013.
- George H Shull. The composition of a field of maize. *Journal of Heredity*, (1):296–301, 1908.
- George Harrison Shull. What is "heterosis"? *Genetics*, 33(5):439, 1948.
- Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.

- Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.
- Nadia Solovieff, Chris Cotsapas, Phil H Lee, Shaun M Purcell, and Jordan W Smoller. Pleiotropy in complex traits: challenges and strategies. *Nature Reviews Genetics*, 14(7):483–495, 2013.
- Erik LL Sonnhammer, Sean R Eddy, Richard Durbin, et al. Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins-Structure Function and Genetics*, 28(3):405–420, 1997.
- George F Sprague and Loyd A Tatum. General vs. specific combining ability in single crosses of corn. *Journal of the American Society of Agronomy*, 1942.
- Eli A Stahl, Soumya Raychaudhuri, Elaine F Remmers, Gang Xie, Stephen Eyre, Brian P Thomson, Yonghong Li, Fina AS Kurreeman, Alexandra Zhernakova, Anne Hinks, et al. Genome-wide association study meta-analysis identifies seven new rheumatoid arthritis risk loci. *Nature genetics*, 42(6):508–514, 2010.
- Peter D Stenson, Matthew Mort, Edward V Ball, Katy Shaw, Andrew D Phillips, and David N Cooper. The human gene mutation database: building a comprehensive mutation repository for clinical and molecular genetics, diagnostic testing and personalized genomic medicine. *Human genetics*, 133(1):1–9, 2014.
- John D Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):479–498, 2002.
- John D Storey. The positive false discovery rate: A bayesian interpretation and the q-value. *Annals of statistics*, pages 2013–2035, 2003.
- John D Storey. False discovery rate. In *International encyclopedia of statistical science*, pages 504–508. Springer Berlin Heidelberg, 2011.
- John D Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.
- Samuel A. Stouffer, Edward A. Suchman, Leland C. Devinney, Shirley A. Star, and Robin M. Williams Jr. *The American soldier: adjustment during army life. (Studies in social psychology in World War II, Vol. 1.)*, volume xii. Princeton Univ. Press, Oxford, England, 1949.
- Charles W Stuber, Stephen E Lincoln, DW Wolff, T Helentjaris, and ES Lander. Identification of genetic factors contributing to heterosis in a hybrid from two elite maize inbred lines using molecular markers. *Genetics*, 132(3):823–839, 1992.
- Mahito Sugiyama, Chloé-Agathe Azencott, **Dominik Grimm**, Yoshinobu Kawahara, and Karsten Borgwardt. Multi-task feature selection on multiple networks via maximum flows. In *Proc. of the 2014 SIAM Int’l Conf. on Data Mining (SDM’14)*, pages 199–207, 2014.

- Mahito Sugiyama, Felipe Llinares-López, Niklas Kasenburg, and Karsten M Borgwardt. Significant subgraph mining with multiple testing correction. *Proc. of the 2015 SIAM Int'l Conf. on Data Mining (SDM'15)*, pages 37–45, 2015.
- Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.
- RE Tarone. A modified bonferroni method for discrete data. *Biometrics*, pages 515–522, 1990.
- Aika Terada, Mariko Okada-Hatakeyama, Koji Tsuda, and Jun Sese. Statistical significance of combinatorial regulations. *Proceedings of the National Academy of Sciences*, 110(32):12996–13001, 2013.
- Janita Thusberg, Ayodeji Olatubosun, and Mauno Vihinen. Performance of mutation pathogenicity prediction methods on missense variants. *Human mutation*, 32(4):358–368, 2011.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- Eray Tuzun, Andrew J Sharp, Jeffrey A Bailey, Rajinder Kaul, V Anne Morrison, Lisa M Pertz, Eric Haugen, Hillary Hayden, Donna Albertson, Daniel Pinkel, et al. Fine-scale structural variation of the human genome. *Nature genetics*, 37(7):727–732, 2005.
- Robert Vaser, Swarnaseetha Adusumalli, Ngak Leng Sim, Sikic Mile, and Pauline C. Ng. SIFT 4G: Missense Predictions for Genomes. *In Preparation*, 2015.
- Mauno Vihinen. How to evaluate performance of prediction methods? measures and their interpretation in variation effect analysis. *BMC genomics*, 13(Suppl 4):S2, 2012.
- Mauno Vihinen. Guidelines for reporting and using prediction tools for genetic variation analysis. *Human mutation*, 34(2):275–282, 2013.
- Peter M Visscher. Sizing up human height variation. *Nature genetics*, 40(5):489–490, 2008.
- Peter M Visscher, Matthew A Brown, Mark I McCarthy, and Jian Yang. Five years of gwas discovery. *The American Journal of Human Genetics*, 90(1):7–24, 2012.
- Congmao Wang, Chang Liu, Damian Roqueiro, Dominik Grimm, Rebecca Schwab, Claude Becker, Christa Lanz, and Detlef Weigel. Genome-wide analysis of local chromatin packing in arabidopsis thaliana. *Genome research*, 25(2):246–256, 2015.

- Dong Wang, Kent M Eskridge, and Jose Crossa. Identifying qtls and epistasis in structured plant populations using adaptive mixed lasso. *Journal of agricultural, biological, and environmental statistics*, 16(2):170–184, 2011.
- Michael N Weedon, Hana Lango, Cecilia M Lindgren, Chris Wallace, David M Evans, Massimo Mangino, Rachel M Freathy, John RB Perry, Suzanne Stevens, Alistair S Hall, et al. Genome-wide association analysis identifies 20 loci that influence adult height. *Nature genetics*, 40(5):575–583, 2008.
- Joachim Weischenfeldt, Orsolya Symmons, Francois Spitz, and Jan O Korbel. Phenotypic impact of genomic structural variation: insights from and for human disease. *Nature Reviews Genetics*, 14(2):125–138, 2013.
- SJ Welham and R Thompson. Likelihood ratio tests for fixed model terms using residual maximum likelihood. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):701–714, 1997.
- Samuel S Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938.
- Cristen J Willer, Yun Li, and Gonçalo R Abecasis. Metal: fast and efficient meta-analysis of genomewide association scans. *Bioinformatics*, 26(17):2190–2191, 2010.
- Michael C Wu, Seunggeun Lee, Tianxi Cai, Yun Li, Michael Boehnke, and Xihong Lin. Rare-variant association testing for sequencing data with the sequence kernel association test. *The American Journal of Human Genetics*, 89(1):82–93, 2011.
- Jinhua Xiao, Jiming Li, Longping Yuan, and Steven D Tanksley. Dominance is the major genetic basis of heterosis in rice as revealed by qtl analysis using molecular markers. *Genetics*, 140(2):745–754, 1995.
- Jian Yang, Beben Benyamin, Brian P McEvoy, Scott Gordon, Anjali K Henders, Dale R Nyholt, Pamela A Madden, Andrew C Heath, Nicholas G Martin, Grant W Montgomery, et al. Common snps explain a large proportion of the heritability for human height. *Nature genetics*, 42(7):565–569, 2010.
- Jian Yang, S Hong Lee, Michael E Goddard, and Peter M Visscher. Gcta: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011.
- Kai Ye, Marcel H Schulz, Quan Long, Rolf Apweiler, and Zemin Ning. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, 25(21):2865–2871, 2009.
- Tjalling J Ypma. Historical development of the newton-raphson method. *SIAM review*, 37(4):531–551, 1995.

- Jianming Yu, Gael Pressoir, William H Briggs, Irie Vroh Bi, Masanori Yamasaki, John F Doebley, Michael D McMullen, Brandon S Gaut, Dahlia M Nielsen, James B Holland, et al. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature genetics*, 38(2):203–208, 2006.
- Dmitri V Zaykin, S Stanley Young, and Peter H Westfall. Using the false discovery rate approach in the genetic dissection of complex traits: a response to weller et al. *Genetics*, 154(4):1917–1918, 2000.
- Xiang Zhang, Fei Zou, and Wei Wang. Fastanova: an efficient algorithm for genome-wide association study. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 821–829. ACM, 2008.
- Xiang Zhang, Feng Pan, Yuying Xie, Fei Zou, and Wei Wang. Coe: a general approach for efficient genome-wide two-locus epistasis test in disease association study. In *Research in Computational Molecular Biology*, pages 253–269. Springer, 2009.
- Xiang Zhang, Shunping Huang, Fei Zou, and Wei Wang. Team: efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics*, 26(12): i217–i227, 2010a.
- Yu Zhang, Dit-Yan Yeung, and Qian Xu. Probabilistic multi-task feature selection. In *Advances in neural information processing systems*, pages 2559–2567, 2010b.
- Zhiwu Zhang, Elhan Ersoz, Chao-Qiang Lai, Rory J Todhunter, Hemant K Tiwari, Michael A Gore, Peter J Bradbury, Jianming Yu, Donna K Arnett, Jose M Ordovas, et al. Mixed linear model approach adapted for genome-wide association studies. *Nature genetics*, 42(4):355–360, 2010c.
- Keyan Zhao, María José Aranzana, Sung Kim, Clare Lister, Chikako Shindo, Chunlao Tang, Christopher Toomajian, Honggang Zheng, Caroline Dean, Paul Marjoram, et al. An arabidopsis example of association mapping in structured samples. *PLoS Genetics*, 3(1):e4, 2007.
- Keyan Zhao, Chih-Wei Tung, Georgia C Eizenga, Mark H Wright, M Liakat Ali, Adam H Price, Gareth J Norton, M Rafiqul Islam, Andy Reynolds, Jason Mezey, et al. Genome-wide association mapping reveals a rich genetic architecture of complex traits in *oryza sativa*. *Nature communications*, 2:467, 2011.
- Yang Zhou, Rong Jin, and Steven Hoi. Exclusive lasso for multi-task feature selection. In *International Conference on Artificial Intelligence and Statistics*, pages 988–995, 2010.
- Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2): 301–320, 2005.

James Zou, Christoph Lippert, David Heckerman, Martin Aryee, and Jennifer Listgarten. Epigenome-wide association studies without the need for cell-type composition. *Nature methods*, 11(3):309–311, 2014.

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