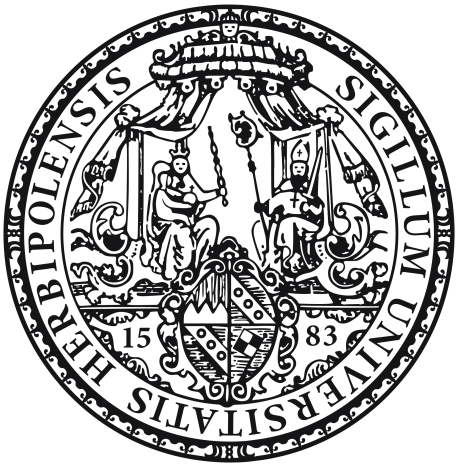
JULIUS-MAXIMILIANS UNIVERSITÄT

WÜRZBURG



Quantitative genetics: from genome assemblies to neural network aided omics-based prediction of complex traits

Doctoral Thesis at the Graduate School of Life Sciences

Section Integrative Biology

submitted by

Jan Alexander FREUDENTHAL

from **Lübeck, Schleswig-Holstein, Germany**

Submitted on: .....

Members of the Thesis Committee:

Chairperson: *Prof. Thomas Schmitt*

Primary Supervisor: *Prof. Arthur Korte*

Supervisor (Second): *Prof. Jörg Schultz*

Supervisor (Third): *Prof. Thomas Dandekar*

Date of Public Defence: .....

Date of Receipt of Certificates: .....

Affidavit

I hereby confirm that my thesis entitled *"Quantitative genetics - from genome assemblies to neural network aided omics based prediction of complex traits"* is the result of my own work. I did not receive any help or support from commercial consultants. All sources and or materials applied are listed and specified in the thesis. Furthermore, I confirm that this thesis has not yet been submitted as part of another examination process neither in identical nor in similar form.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, die Dissertation mit dem Titel *"Quantitative Genetik - von Genomassemblierungen bis zur Vorhersage von Phänotypischen Merkmalen mit Hilfe von Omics unterstüzten Neuronalen Netzwerken"* eigenständig, d.h. insbesondere selbständig und ohne Hilfe eines kommerziellen Promotionsberaters, angefertigt und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet zu haben. Ich erkläre außerdem, dass die Dissertation weder in gleicher noch in ähnlicher Form bereits in einem anderen Prüfungsverfahren vorgelegen hat.

"Wit beyond measure is man's greatest treasure"

Rowena Rawenclaw

Acknowledgements

Thanks, to Prof. Arthur Korte for hiring me and the opportunity to conduct research and write this thesis in his group, as well as the invaluable help.

Thanks, to Arthur, Prof. Jörg Schultz and Prof. Thomas Dandekar for the supervision and being part of the thesis committee.

Thanks, to Prof. Thomas Schmitt for being the chairperson of the thesis committee.

Thanks, to the co-authors of the chloroplast benchmarking project: Niklas Terhoeven, Simon Pfaff, Markus Ankenbrand and Frank Förster.

Thanks, to the co-authors of the GWAS-Flow project: Markus Ankenbrand, Dominik Grimm and of course Arthur.

Thanks, to the students who assisted in the genomic selection projects: Laura Steinmann, Roman Saiz and Florens Fischer as well as the other collaborators Markus Ankenbrand and Torsten Pook.

Thanks, to the German Federal Ministry of Education and Research (BMBF) and the German tax payers for funding the MAZE project within the scope of the funding initiative "Plant Breeding Research for the Bioeconomy" (Funding ID: 031B0195).

Thanks, to the MAZE project partners for the great collaboration in the past three years.

And lastly, thanks to everyone at the CCTB for creating a great working environment.

Contents

List of Figures

List of Tables

List of Abbreviations

Adadelta	**Ada**ptive **delta**
Adagrad	**Ada**ptive **Grad**ient Algorithm
Adam	**Ada**ptive **M**oment estimation
ANN	**A**rtificial **N**eural **N**etwork
API	**A**pplication **P**rogramming **I**nterface
AUC	**A**rea **U**nder the **C**urve
BL	**B**ayesian **L**asso
BLUE	**B**est **L**inear **U**nbiased **E**stimator
BLUP	**B**est **L**inear **U**nbiased **P**redictor
bp	**B**ase **P**air
BRR	**B**ayesian **R**idge **R**egression
CPU	**C**ore **P**rocessing **U**nit
DH	**D**oubled **H**aploid
DNA	**D**eoxyribo**N**ucleic **A**cid
EMMA	**E**fficient **M**ixed **M**odel **A**ssociations
FDR	**F**alse **D**iscovery **R**ate
FCL	**F**ully **C**onnected **L**ayer
GBLUP	**G**enomic **B**est **L**inear **U**nbiased **P**redictor
GD	**G**radient **D**escent
GEBC	**G**enomic **E**stimated **B**reeding **V**alues
GPL	**G**eneral **P**ublic **L**icense
GP	**G**enomic **P**rediction
GPU	**G**raphical **P**rocessing **U**nit

GRM	**G**enomic **R**elationship **M**atrix
GS	**G**enomic **S**election
GUI	**G**raphical **U**ser **I**nterface
GWAIS	**G**enome **W**ide **I**nteraction **A**ssociation **S**tudies
GWAS	**G**enome **W**ide **A**ssociation **S**tudies
HDF	**H**ierarchical **D**ata **F**ormat
HL	**H**idden **L**ayer
HPC	**H**igh **P**erformance **C**omputing
IR	**I**nverted **R**epeat
LCL	**L**ocally **C**onnected **L**ayer
LD	**L**inkage **D**isequilibrium
LMM	**L**inear **M**ixed **M**odel
LSC	**L**arge **S**ingle **C**opy
MAF	**M**inor **A**llele **F**requency
MCMC	**M**arkov **C**hain **M**onte **C**arlo
MLP	**M**ulti **L**ayer **P**erceptron
ML	**M**achine **L**earning
MSE	**M**ean **S**quare **E**rror
Nadam	**N**esterov-**a**ccelerated **A**daptive **M**oment estimation
NAG	**N**esterov **A**ccelerated **M**omentum
NCBI	**N**ational **C**enter for **B**iotechnological **I**nformation
QTL	**Q**uantitative **T**rait **L**ocus
ReLU	**Re**ctified **L**inear **U**nits
RKHS	**R**eproducing **K**ernel **H**ilbert **S**paces
RMSE	**R**oot **M**ean **S**quare **E**rror
RMSProp	**R**oot **M**ean **S**quare **Prop**agation
RNA	**R**ibo**N**ucleic **A**cid
ROC	**R**eceiver **O**perating **C**haracteristics

RSS	**R**esidual **S**um of **S**quares
SGD	**S**tochastic **G**radient **D**escent
SNP	**S**ingle **N**ucleotide **P**olymorphism
SRA	**S**equence **R**ead **A**rchive
SSC	**S**mall **S**ingle **C**opy
TRN	**TR**ai**N**ing subset
TST	Te**ST**ing subset
WGS	**W**hole **G**enome **S**equencing
XOR	e**X**clusive **OR**

1 General introduction

Plant breeding is a process that started as early as agriculture itself around 10,000 BC. Even after twelve millennia many of its aspects are still obscured behind complex genetics and our lack to thoroughly comprehend them. But never were the challenges imposed on breeding as vast as today. In the year 2050 the world's agriculture will be responsible for feeding nine to ten billion people GERLAND et al., 2014, next to other inflicted responsibilities of replacing fossil fuels with regenerate energy from plants, providing fibers for industrial textile production and pharmaceutical applications. Those demands will be met without continuing advances in breeding and in quantitative genetics.

According to WALLACE, RODGERS-MELNICK, and BUCKLER, 2018 the history of breeding can be divided into four main epochs that always utilized the technologies available to them in a specific era. In the beginning breeding did not exist as a succinct field of science and was accomplished by simple phenotypic selection by local farmers, which lead to dramatic changes in approximately 7000 cultivated crop species compared to their wild ancestors KHOURY et al., 2016.

The next era in plant breeding was sparked by the upcoming of new statistical methods and the rediscovery of Mendelian genetics in the late 19th and early 20th century, which in combination let to the development of quantitative genetics TSCHERMAK, 1900; FISHER, 1919; FISHER and MACKENZIE, 1923; FALCONER and MACKAY, 1996. Along with it came the discovery of inbreeding and inbreeding depression, schematic design of field trials, the concept of variance component analysis, hybrid breeding and others.

The third stage, the genomic era of plant breeding, began with the discovery

of possibilities to asses polymorphisms in the genomes, leading up to marker-assisted selection, linkage and QTL mapping. As marker arrays grew larger and sequencing costs declined dramatically, those methods were succeeded by the more sophisticated and precise whole genome regression and genome-wide association studies (GWAS) with high-density marker maps HAYES and GODDARD, 2001; KORTE and FARLOW, 2013.

Those technological advances allowed plant breeders to provide farmers with cultivars, which were able to feed the exponentially growing world's populations since the 1950s. However, like any century before, the 21th imposes great challenges on humankind. Climate change leads to different stresses in the environments and plant breeders need to adapt to the specific requirements on high-yielding cultivars, maybe quicker than ever before, as droughts and flooding occur more often around the world each year.

This, however, is complicated because of the 7000 cultivated plants in agricultural history only a few provide the major source of food today on a global scale, with most important being maize (*Zea mays*), wheat *Triticum aestivum* and rice *Oryza sativa*. Furthermore, during the course of breeding the elite cultivars have lost the majority of the genetic diversity of its ancestral wild populations WALSH and LYNCH, 2018b. To continuously adapt and improve crop plants all the methods of quantitative genetics, genomics and genome editing need to be combined in the modern age of Breeding 4.0 WALLACE, RODGERS-MELNICK, and BUCKLER, 2018.

Quantitative genetics is a multi-step process and requires high quality data of both genomes and traits. Figure 1.1 shows a flow diagram with the major processes involved in going from genome assemblies to neural network aided genomic prediction of complex traits for plant breeding and GWAS.

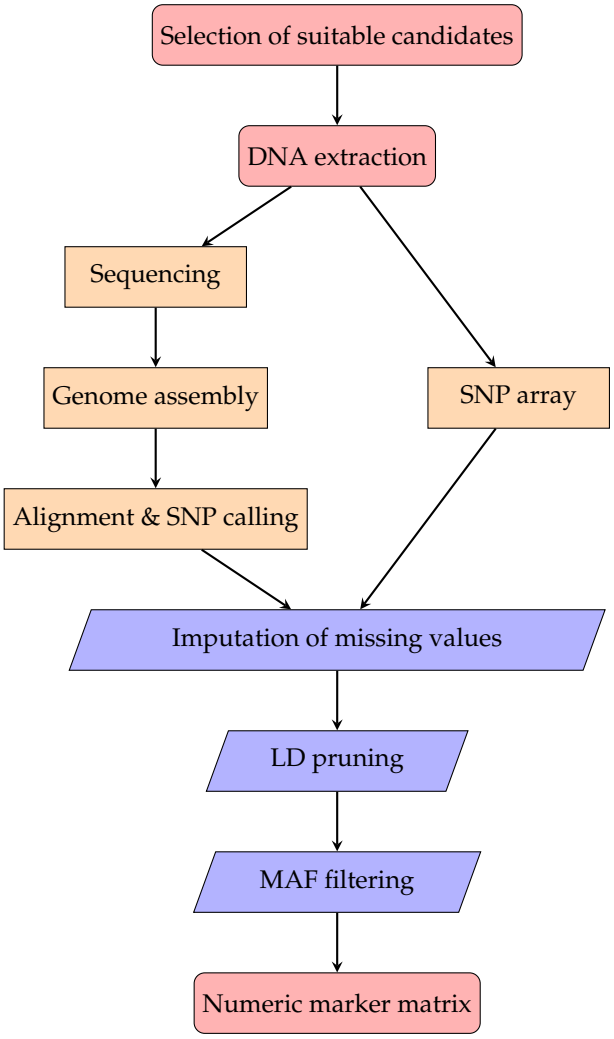


FIGURE 1.1: Schematic process of genotyping for quantitative genetics analyses with its crucial steps

Chapter 2 will focus on the first parts of figure 1.1: the assembly of genomes, which will be exemplified on plastid genomes and will elucidate some of the major obstacles presented, when starting with raw DNA reads and advancing towards genomic data, which is suitable for quantitative methods like GWAS and genomic selection. Section 3 will introduce a novel tool to perform large-scale GWAS using modern software and computing resources, which allows to perform those analyses on large scales in reasonable amounts of time.

Chapter 4 will give in depth introductions to machine learning and the complex architecture of quantitative traits and brings them in the context of plant breeding

and will further elucidate how those techniques can be used to continue improving plant germplasms for modern agriculture via genomic selection. Genomic selection is a process during which plants are not only selected and assessed based on their phenotypic appearances, single markers or pedigree relatedness to other individuals, but mainly on their genomic features HAYES and GODDARD, 2001.
In the final chapter figure 1.1 will be recapitulated and the main obstacles in the process will be thoroughly decomposed and explained how this study aids towards providing solutions for some of them.

2 Benchmarking of chloroplast genome assembly tools

This chapter is oriented on FREUDENTHAL et al., 2019b, which as has been published on the preprint server bioRχiv and submitted for peer review. Only the parts from the publication, which the author majorly contributed to are included. If not cited otherwise the plots were designed and generated by the author of this thesis.

2.1 Introduction

2.1.1 Motivation

Some organelles like mitochondria and chloroplasts contain their own genetic information from which they are able to synthesize certain proteins independent of the nucleus genome. Evolutionary this developed during endosymbiosis, a process which underlying theory seeks to explain how eukaryotic cells formed from prokaryotes MERESCHKOWSKY, 1905; KUTSCHERA and NIKLAS, 2005. This widely acknowledged hypothesis explains how in the early evolution of eukaryotes other cells were incorporated, which ultimately became organelles. The most likely precursors of today's chloroplasts were photosynthetic bacteria or similar organisms ARCHIBALD, 2015. This process left its traces in the structure of chloroplast genomes until today, which resemble more closely prokaryotic genomes than that of its eukaryotic host cells. A typical chloroplast genome consists of circular DNA with a size between 120 kbp to 160 kbp PALMER, 1985, while plant core

genomes are linear, organized in chromosomes and larger by multiple orders of magnitudes.

The first chloroplasts have been sequenced as early as 1986 and were isolated from *Marchantia polymorpha* and *Nicotiana tabacum* OHYAMA et al., 1986; SHINOZAKI et al., 1986. Complete reviews on the structure of chloroplast genomes were authored by GREEN, 2011 and WICKE et al., 2011. Chloroplast genomics is widely applied in evolutionary studies aiding to elucidate the processes involved in endosymbiosis and the development of photosynthetic plants MARTIN et al., 2002; XIAO-MING et al., 2017. Over the course of natural adaptation, the plastid genome has been reduced in size through endosymbiotic gene transfer, a form of horizontal gene transfer, where fractions of plastid genomes are incorporated in the core genome MARTIN et al., 2002; DEINER et al., 2017. This mechanism of evolution is still ongoing an can be observed *in vitro* BOCK, 2017; FUENTES et al., 2014; STEGEMANN and BOCK, 2009.

In the case of *Arabidopsis thaliana*, this process resulted in 14 % of the core genome's genes previously being located on the chloroplast, while 100-120 genes remain on the chloroplast itself WICKE et al., 2011, which by far would not suffice to allow the chloroplast to function independently of its host cell. The fact that organelle genomes are much smaller and highly conserved with a large gene content leads to polymorphisms being more likely to cause functional changes in physiological processes. Another difference between organelle and core genomes is that single chloroplasts contain up to hundreds of copies of its own genome KUMAR, OLDENBURG, and BENDICH, 2014; BENDICH, 1987. Considering that photosynthetic active cells again contain multiple chloroplasts means that the number of chloroplast genomes therefore is considerably higher than the number of core genomes per cell.

Structurally, chloroplast genomes are made up of four distinct regions. Two inverted repeats (IR), IR_A and IR_B, ranging from 10 kbp to 76 kbp in size that divide the circular genome into two regions: the large single copy (LSC) and the small

single copy (SSC) as shown in figure 2.1 PALMER, 1985.

Taking into account that the majority of assembly tools has been designed to assemble linear core genomes, the structure of chloroplast genomes is an obstacle for many assembly pipelines to overcome. This holds true especially for solving and aligning the IRs correctly WANG et al., 2018.

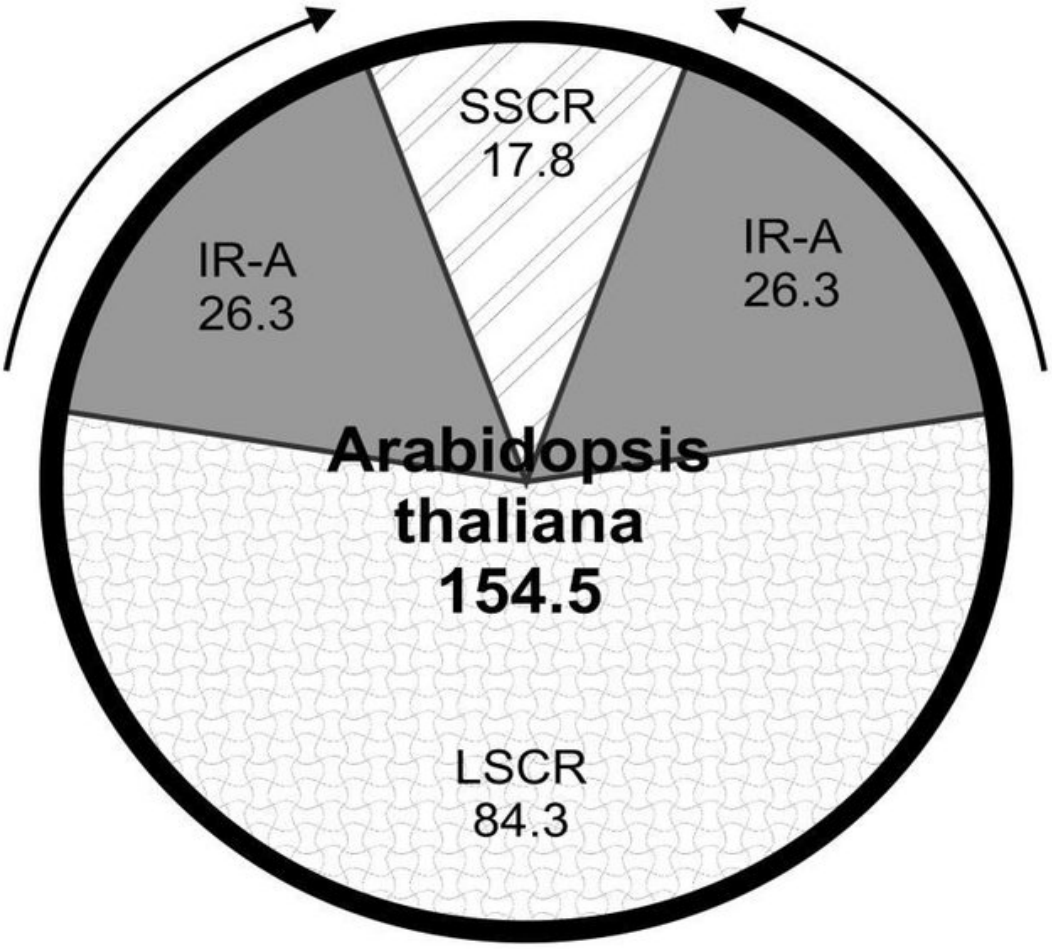


FIGURE 2.1: Structure of the chloroplast genome of *A. thaliana* with small single copy region (SSCR) and large single copy region (LSCR). The number denotes the length of the genome and its parts in kilo base pairs (kbp). Graphic from OLEJNICZAK et al., 2016

Another difficulty for the assembly is heteroplasmy, which describes the phenomenon of co-existence of multiple versions of the chloroplast's genome in a single organism and even single cells of that respective organism. Heteroplasmy

complicates genome assemblies and ongoing from there the downstream analyses CORRIVEAU and COLEMAN, 1988; CHAT et al., 2002. The underlying evolutionary mechanisms behind heteroplasmy are not fully understood and existing fitness advantages fueling heteroplasmy cannot be explained satisfactory by standard evolutionary methods SCARCELLI et al., 2016.

Derived from a multitude of plant genome projects, there is a large variety of databases, containing short read data for species without assembled organelle genomes available, e.g. NCBI's sequence read archive (SRA) LEINONEN et al., 2010. Because most plant DNA extraction protocols applied to procure raw input for sequencing use green leaf tissue as their basis, they also contain a large amount of plastid DNA, providing a valuable source for organelle genome assembly pipelines. In the course of this chapter, the performance of these pipelines will be assessed.
Having larger numbers of assembled and annotated chloroplast genomes publicly available will be beneficial for evolutionary studies and are a useful addition to bar-coding and super-barcoding COISSAC et al., 2016, aside from other biotechnological applications DANIELL et al., 2016. To obtain those there is a variety of tools available. In the course of this chapter the availability, usability and overall performance of seven of those assembly pipelines will be assessed. Ultimately, the newly gained insights will be utilized to attempt to assemble more than 100 chloroplasts *de novo*.

2.1.2 Extraction of chloroplast reads from whole genome data and general assembly workflow

There is large array of strategies to assemble chloroplast genomes from raw sequencing data TWYFORD and NESS, 2017. In general the process of chloroplast genome assembly involves three steps:

(i) extraction of plastid reads from the whole-genome sequencing (WGS) data

(ii) assembly of the plastid genome

(iii) solving the circular structure of the genome with the IRs.

There are three approaches to address step (i): the first one is to map all reads to a reference chloroplast VINGA et al., 2012, which works reasonably well if there is one available for the same or at least a closely related species. The second approach is to make use of the much larger coverage of chloroplast DNA compared to core DNA with a k-mer analysis CHAN and RAGAN, 2013. This is for example done by `chloroExtractor`, one of the tools used in this study ANKENBRAND et al., 2018. The third way to accomplish plastid DNA extraction is to combine both methods as done by `NOVOPlasty` DIERCKXSENS, MARDULYN, and SMITS, 2017. Figure 2.2 shows the general workflow of chloroplast assembly tools with the bifurcation at step (ii).

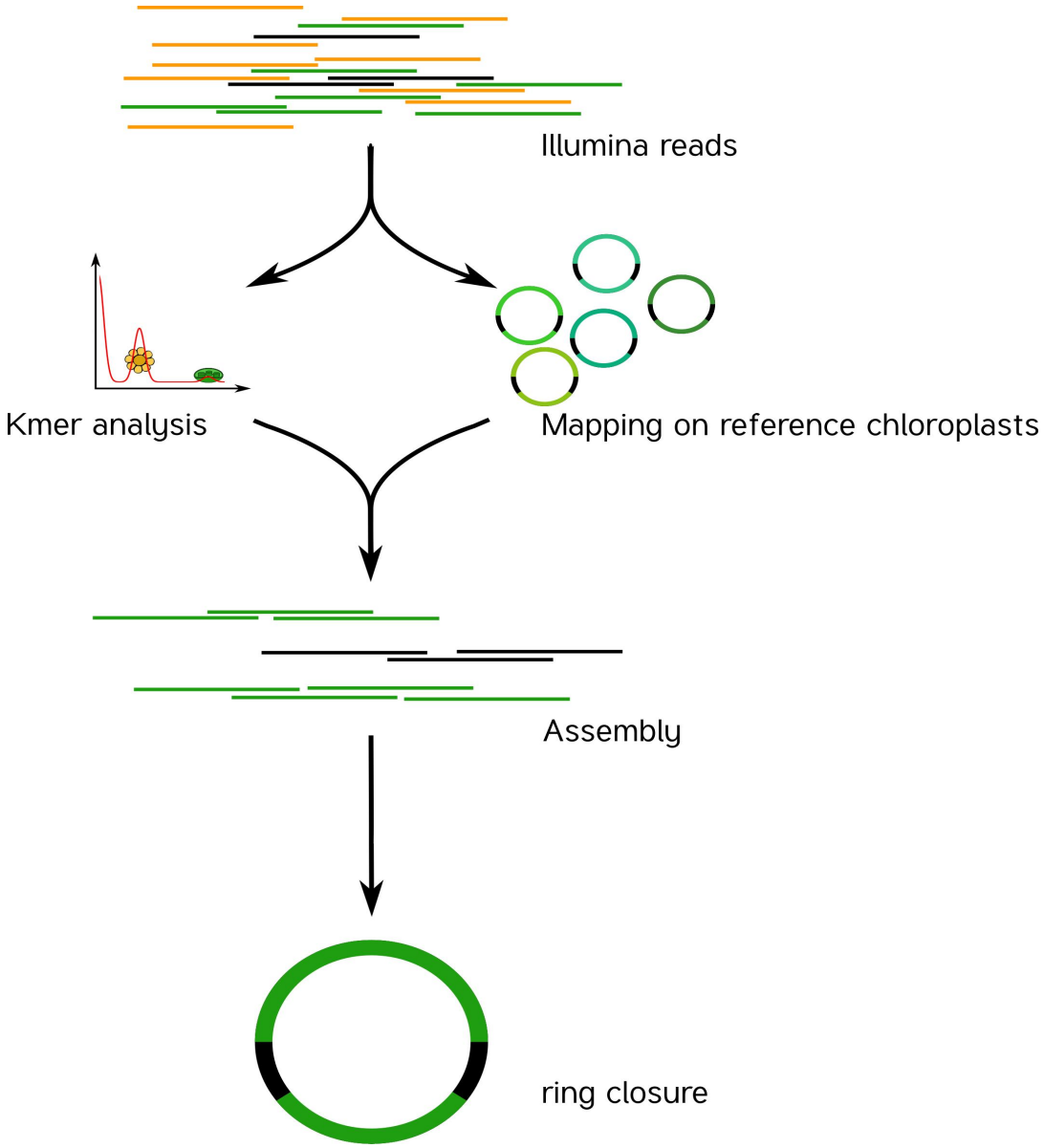


FIGURE 2.2: Standard workflow of chloroplast genome assembly. Graphic from ANKENBRAND et al., 2018

Purpose and scope of benchmarking the landscape of chloroplast assembly tools

The purpose of this study is to provide insights into the landscape of chloroplast assembly tools, to recommend best practices for organelle genome assemblies and to contribute *de novo* assemblies for many species and families without a reference chloroplast available so far to the scientific community.

2.2 Material and methods

2.2.1 Methods

Data and code availability

All the source code and data used is publicly available under the terms of the MIT-License. The source code has been published on github *GitHub Repository for Benchmark Project* and archived on zenodo FÖRSTER and ANKENBRAND, 2019 . The docker images are available on dockerhub *Docker Hub Group for Benchmark Project*.

Tools

To be included into this study, the software, including the source code, must be publicly available. The study was further restricted to paired-end Illumina data sets as their sole input source because they were abundantly available for this benchmark. The only technical requirement was being able to assemble chloroplast genomes from those paired-end Illumina reads. The other requirements were dictated by reproducibility. The software must be open-source and available under the terms of a liberal software license and the software must be able to be operated from a command line, since GUI-only tools are not suited for highly repetitive, automated analyses. In total there were seven tools that met those conditions:

(i) `chloroExtractor` ANKENBRAND et al., 2018

(ii) `Chloroplast assembly protocol` SANCHO et al., 2018

(iii) `GetOrganelle` JIN et al., 2018

(iv) `ORG.Asm` COISSAC et al., 2016

(v) `IOGA` BAKKER et al., 2016

(vi) `Fast-Plast` MCKAIN and AFINIT, 2017

(vii) `NOVOPlasty` DIERCKXSENS, MARDULYN, and SMITS, 2017

Standardization and reproducibility

Along with the study, easy and ready-to-use versions of all the involved programs, working with standardized input, were published. For this purpose `docker` containers MERKEL, 2014 were implemented. To work with the containers in a closed HPC environment they were transformed into related `singularity` containers KURTZER, SOCHAT, and BAUER, 2017. To apply the programs users simply need to provide two files: one for the forward reads (`forward.fq`) and one for the reverse reads (`reverse.fq`) and run the containers without any need for further configuration or installation besides `docker` or `singularity` itself, which can be easily done on all popular operating systems. Both files are required to be in FASTQ format. Besides the individual output files recording the process of the respective program, all programs write the assembly products into files called `output.fa` in FASTA format. For the quantitative and consistency measurements the singularity containers were run on the Julia HPC-cluster of the University of Würzburg using the SLURM workload manager JETTE, YOO, and GRONDONA, 2002. All runs for all assemblies were set with a time limit of 48 hours. This was necessary because some assemblers e.g. `IOGA`, if they did not finish after at least 12 hours, showed the tendency not to finish, even after weeks of running.

2.2.2 Data

Three different data sets were used:

(i) simulated data from *A. thaliana* chloroplasts

(ii) real data with known reference chloroplast to rate the success of the assemblies

(iii) novel data sets from NCBI's SRA without a known reference chloroplast to apply the gained knowledge to the *de novo* assembly of more than 100 chloroplasts.

Simulated data

To allow full control over all the parameters involved we started with the simulated data. In the present case the data's input parameters, thought to be influential on the outcome, were: the read length, the ratio between chloroplast and core genome reads as well as the total size of the data set. The data simulations were based on real data from the TAIR10 genome of *A. thaliana* BERARDINI et al., 2015 and spawned using `seqkit` SHEN et al., 2016. Core to chloroplast ratios simulated were: 0:1, 1:10, 1:1000 and 1:1000, with read length of 150 and 250 bp. The artificial data consisted either of 2 million read pairs or the full data available. The simulation process was documented and the code and the data are available on github and zenodo ANKENBRAND and FÖRSTER, 2019.

Real data set

Real data was selected from the SRA database. Table 2.1 lists the search parameters that had to be met for a plant to be included in the study from SRA.

TABLE 2.1: Data selection criteria for real data sets from SRA

Choice	Option	Explanation
Organism	green plants	include only photosynthetic plants e.g. no algae
Strategy	wgs	only data from wgs projects included
Platform	Illumina	include only paired-en Illumina reads
Properties	biomol DNA	include only biomol. DNA samples (e.g. no RNA)
Layout	paired	exclude single-end reads
Selection	random	
Access	public	only publicly available data included

In total this resulted in 369 data sets representing a broad variety of the plant kingdom with many different families and genera included.

Novel data sets

To assess the performance of assemblies without a published chloroplast on `CpBase` *CpBase* 105 data sets were selected from SRA. It was emphasized that the chosen read libraries were as distant as possible related to the next relatives with a reference chloroplast, related as possible in taxonomic terms according to NCBI *NCBI Taxonomy*. This was achieved by a phlyogenetic analysis of the accessible data sets on SRA by Frank Förster described in FREUDENTHAL et al., 2019b.

2.2.3 Evaluation

Quantitative

Where applicable, each assembly from each assembler was compared to their respective reference genome by alignment using `minimap2` LI, 2018. Based on those alignments scores were calculated following equation 2.1 from 0 to 100, with 100 being a perfect score.

$$score = \frac{1}{4} \cdot \left(cov_{ref} + cov_{qry} + min \left\{ \frac{cov_{qry}}{cov_{ref}}, \frac{cov_{ref}}{cov_{qry}} \right\} + \frac{1}{n_{contigs}} \right) \cdot 100 \qquad (2.1)$$

Four different metrics contributed equally to the final score:

(i) the coverage of the assembled genome compared to the reference genome cov_{ref} as an estimate for the completeness

(ii) the vice versa case cov_{qry} as a measure for the correctness of the assembly

(iii) the success of resolving the IR correctly, estimated by the difference from the reference and the newly assembled genome $min \left\{ \frac{cov_{qry}}{cov_{ref}}, \frac{cov_{ref}}{cov_{qry}} \right\}$

(iv) the number of total contigs were weighted as $\frac{1}{n_{contigs}}$ giving a chloroplast with one contig the optimal score

While it is difficult to evaluate the success or failure of assemblies on a continuous scale, equation 2.1 allows for objective and unbiased measurements. SNPs or other small variants do not influence the outcome of the score because they are more likely due to in-species variation between plastid genomes and not caused be the assembly itself. Even if the latter is true it would be difficult to determine.

Consistency

For any given bioinformatical application consistency is a desired trait. Software ideally should repeatedly yield the same output when provided with the same input and assembly tools are no exception. To evaluate the reproducibility of the seven tools all the 369 real data sets were assembled and scored twice with each assembler. The correlations between the first and the second run's scores were used as the measure for the robustness of a program.

2.3 Results

2.3.1 Simulated data

The simulated data sets were assembled and scored with all the tools as described above. Figure 2.3 shows a tile plot with the results displaying a color scale from orange over light green to dark green representing the scores from 0 to 100. Blank spaces indicate the failure to produce any output in the given time frame of 48 hours.

While at first sight there is no clear correlation between the input data sets and the score, it is clearly visible that there are grave differences between the assemblers. Two programs, namely `Chloroplast assembly protocol` and `IOGA`, failed to correctly assemble a single chloroplast's genome. `IOGA` even fails to provide an output at all for the majority of the data sets. While those two stand out as negative examples, `Fast-Plast` and `GetOrganelle` stand out as positive examples, perfectly or nearly perfectly assembling all the data sets, with `GetOrganelle` surpassing the performance of `Fast-Plast`. In the middle of the filed are `chloroExtractor`, `ORG.Asm` and `NOVOPlasty` performing reasonably well, but sometimes lacking to solve the IRs and the circular structure.

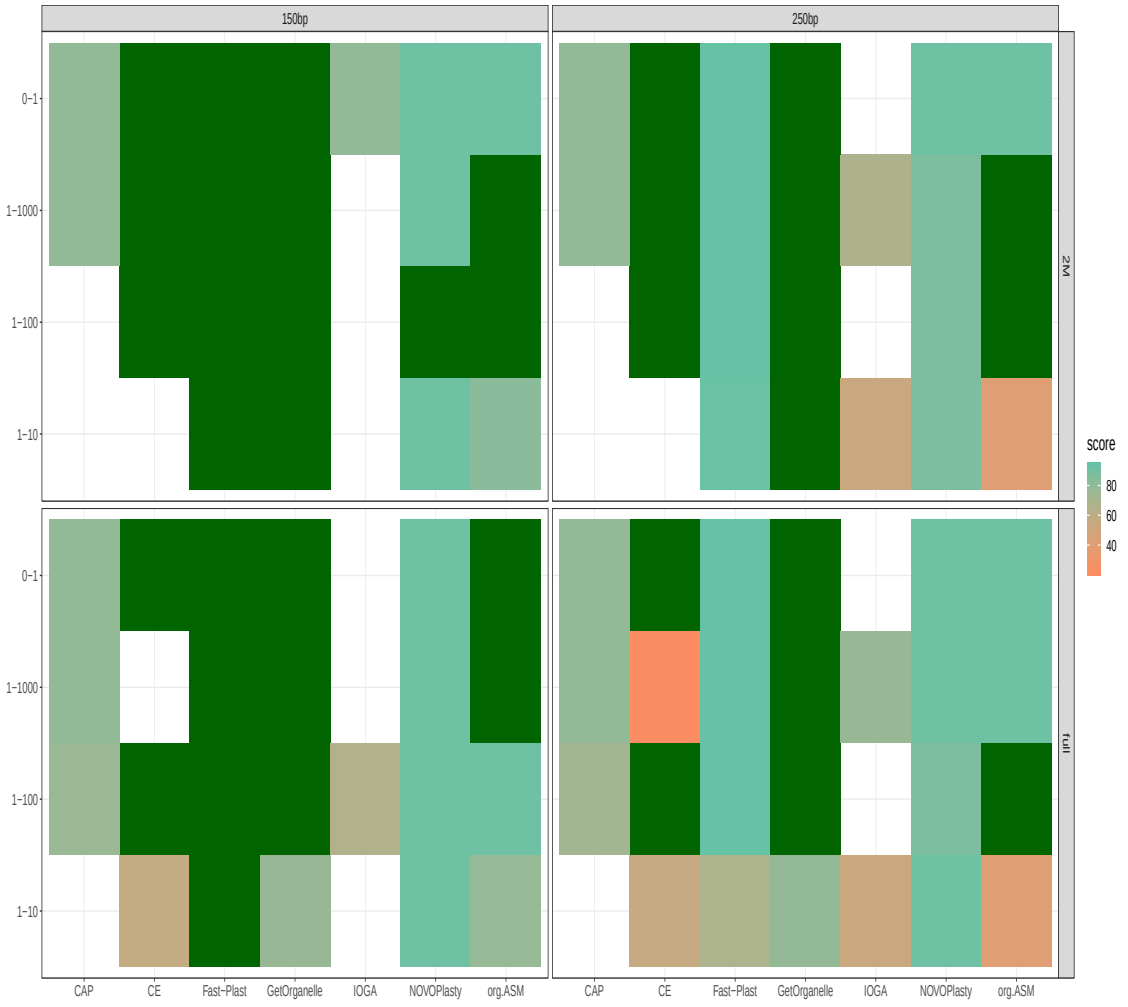


FIGURE 2.3: Results of assemblies executed with simulated data sets. The tile colors from orange to green indicate the score from 0 to 99. Dark green tiles point to scores >99. Blank tiles point to assemblies, which failed to provide an output file. The axis on the left shows the ratio between nucleic and plastid DNA, the one on the right the size of the data sets. On the top the read length in base pairs (bp) is given and in the bottom the seven different assemblers

There is a significant difference between the performance of the assemblers in general. The varying input parameters, however, do not have as grave an influence as the choice of the assembler. While `Fast-Plast` deals with the shorter reads of 150 bp much better than with the longer reads of 250 bp, the scores of the other assemblers do not seem to be influenced by the read length. There is no difference between the full and the subsampled data sets. And while all assemblers appear to be more challenged by low chloroplast to core genome ratios of

1:10, beyond a ratio of 1:100 it does not affect the quality of the assemblies. Table 2.2 shows all the individual results for all data sets and assemblers. For the fields with no entry the respective assembler failed to provide an output.

TABLE 2.2: Scores of assemblies of simulated data, with CAP = `Chloroplast assembly protocol`; CE = `chloroExtractor`; FP = `Fast-Plast`; GO = `GetOrganelle`; NP = `NOVOPlasty`; oA = `ORG.Asm`; length in base pairs (bp).

Set	Length	Ratio	CAP	CE	FP	GO	IOGA	NP	oA
full	150	0-1	79.10	100.00	99.48	100.00		91.52	100.00
2M	150	0-1	79.10	100.00	99.72	100.00	79.10	91.52	91.50
full	150	1-10		56.44	100.00	76.98		91.52	78.00
2M	150	1-10			99.97	100.00		91.52	82.72
full	150	1-100	75.72	100.00	99.48	100.00	66.09	91.52	91.50
2M	150	1-100		100.00	99.47	100.00		100.00	100.00
full	150	1-1000	79.10		99.72	100.00		91.52	100.00
2M	150	1-1000	79.10	100.00	99.72	100.00		91.52	100.00
full	250	0-1	79.10	100.00	93.82	100.00		91.52	91.50
2M	250	0-1	79.10	100.00	93.83	100.00		91.52	91.50
full	250	1-10		54.98	68.45	78.89	52.71	91.52	40.20
2M	250	1-10			93.00	100.00	52.67	87.40	40.20
full	250	1-100	72.81	100.00	93.82	100.00		87.40	100.00
2M	250	1-100		100.00	93.83	100.00		87.40	100.00
full	250	1-1000	79.10	21.30	93.83	100.00	76.96	91.52	91.50
2M	250	1-1000	79.10	100.00	93.83	100.00	67.55	87.40	100.00

2.3.2 Real data sets

Table 2.3 summarizes the results from the assemblies of 369 data sets with the seven assemblers. Similar to the scores of the previous section there is a significant difference between the tools. Likewise `GetOrganelle` is the most successful assembler by a large margin with 210 of 369 chloroplast genomes perfectly assembled. It completely fails to provide output for only 9 data sets, resulting in a median score >99. Contrary `Chloroplast assembly protocol` and `IOGA` both failed to completely assemble a single genome. The performance of `Fast-Plast` is reasonably well in comparison, completing approximately half as many genomes as `GetOrganelle` and being the only other tool whose average score surpasses 90.

Similar to the trials with the simulated data in chapter 2.3.1 `chloroExtractor`, `NOVOPlasty` and `ORG.Asm` are in the middle of the field.

TABLE 2.3: Median scores of chloroplast genome assemblers with inter-quartile range (IQR) and the number of perfect scores (n_perfect) compared to the total number of assemblies (n_tot) providing an output

Assembler	Median	IQR	n_perfect	n_tot
`Chloroplast assembly protocol`	45.25	50.19	0	369
`chloroExtractor`	56.55	71.50	14	369
`Fast-Plast`	92.80	23.59	113	369
`GetOrganelle`	99.83	20.94	210	360
`IOGA`	71.10	11.21	0	338
`NOVOPlasty`	75.95	48.69	58	369
`ORG.Asm`	67.35	91.69	46	348

Figure 2.4 emphasizes the large differences between the assemblers shown in table 2.3. The swarm plots show distinct bands for some assemblers e.g. `NOVOPlasty` and `ORG.Asm`, suggesting that multiple assemblies fail to be solved into a single contig genome at a certain point. As thoroughly discussed in section 2.4, solely from the swarm plot, it is debatable if all the tools should be recommended to be used for the purpose they were designed for.

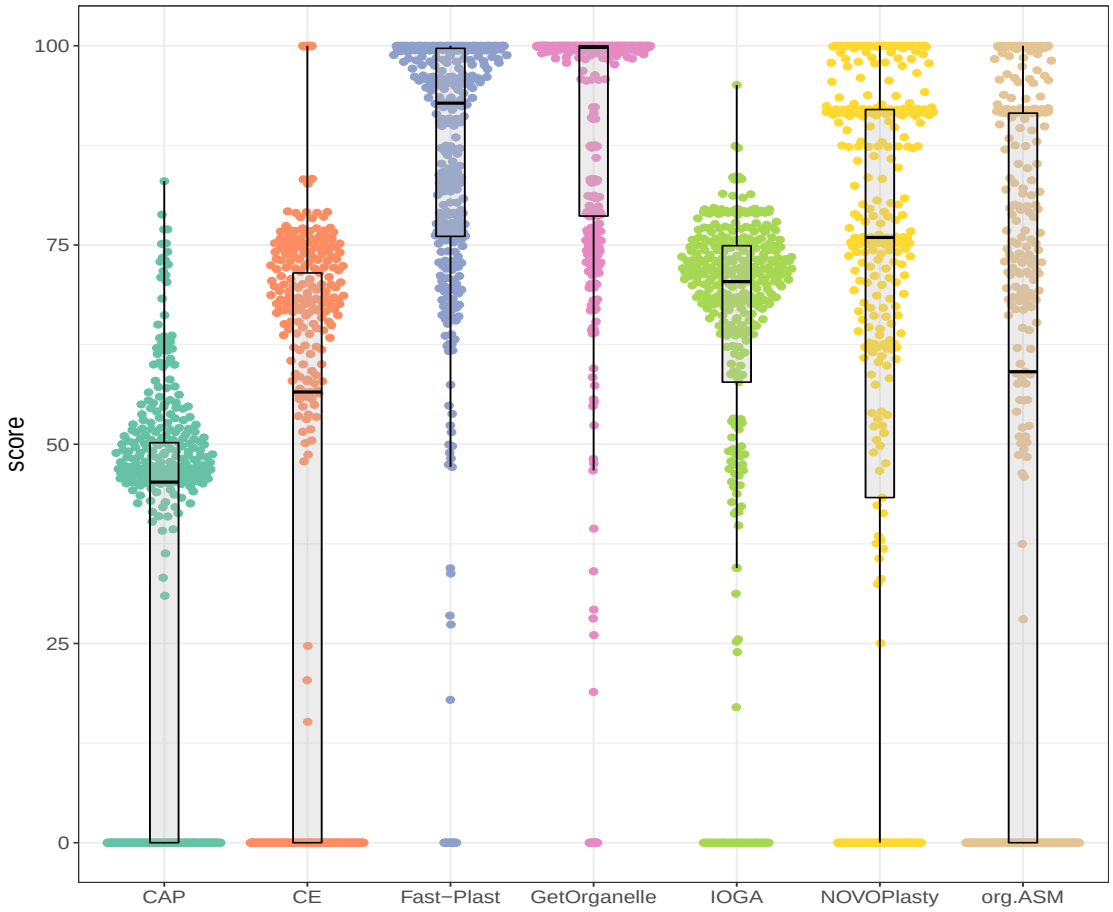


FIGURE 2.4: Box and swarm plots depicting the results from scoring of the assemblies for the real data sets as calculated by equation 2.1

2.3.3 Consistency

Consistency testing was done by re-running every assembly for the real data sets and comparison of the two scores. `chloroExtractor` was the only tool that was 100% consistent over both runs.The consistency plot (figure 2.5) for `Fast-Plast` and `NOVOPlasty` results in arrowhead shaped plots, with differences between the first and second run appearing in assemblies with the highest scores. All other assemblers appear to produce the same output in the two runs, except if either run failed to complete the assembly at all. This is less pronounced for `Chloroplast assembly protocol` and `GetOrganelle` and is a grave issue for `ORG.Asm` and `IOGA`.

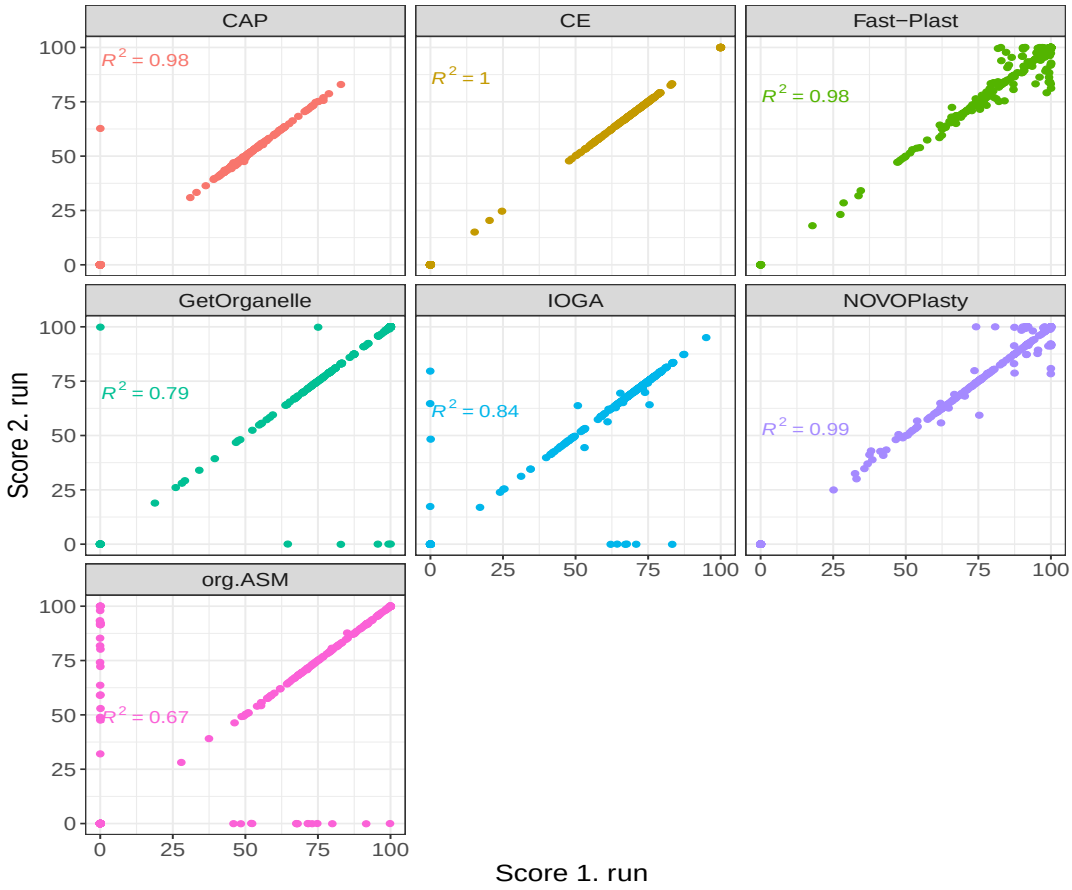


FIGURE 2.5: Swarm plots depicting the results from the scoring shown in 2.1 for two independent runs for each assembler on each of the data sets

2.3.4 Novel assemblies

The final assessment in the evaluation of the assemblers was to test them on novel data sets without a published chloroplast. This step is important for two reasons: (i) it is possible that certain tools perform well on known chloroplasts because they have knowledge of their structure, which would lead to a lack of generalization on unknown genomes. (ii) To apply and test the gained insights with the goal of providing the scientific community with a larger variety of published chloroplast genomes.

As in previous evaluations the most successful assembler was `GetOrganelle`, with 49 out of 105 novel data sets completely assembled.

Lacking a reference genome for alignment, the success had to be defined differently and equation 2.1 was not suitable to evaluate the novel assemblies. Metrics influencing the score of the novel assemblies were the number of contigs, solving the IRs and the size of the SSC and LSC (figure 2.6). This, known to the author, might be biased and not true for all chloroplasts and assumes that all chloroplast genomes evolved according to the general structure described in chapter 2.1. Figure 2.6 compares the results of the assemblies with at least one successful assembly.

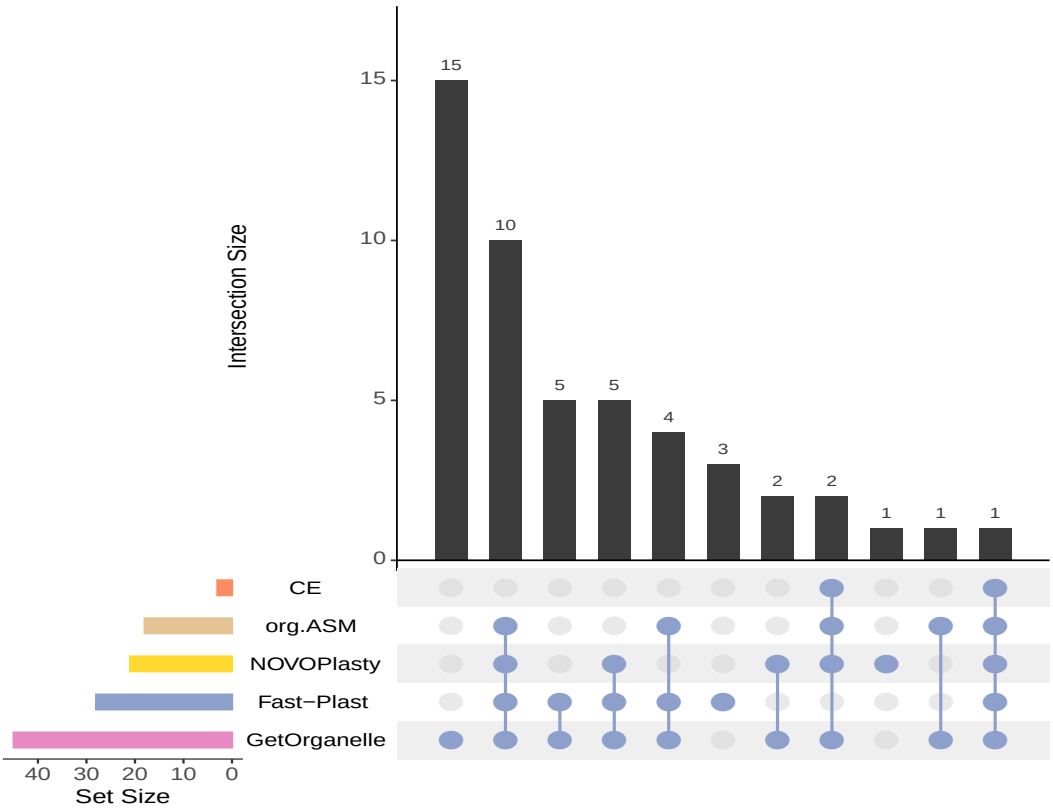


FIGURE 2.6: Upset plot comparing the success (single contig, length $\geq$ 130 kpb, IR $\geq$ 17 kbp) rates of the different assemblers for the novel data sets. The colored, horizontal bar plot show the total amount of successful assemblies for each assemblers. The black, vertical barplot the size of the intersection indicated by the dots in the middle

2.4 Discussion

The study presented in this chapter so far consists of two goals:

(i) to assess the overall performance of a variety of tools designed specifically for the assembly of circular chloroplast genomes from paired-end Illumina reads and

(ii) to *de novo* assemble a variety of yet unpublished chloroplast genomes from existing data.

To accomplish the first goal 16 simulated and 369 real data sets were used adding up to a total of 5166 assemblies for the real data sets and 112 for the simulated data, along 735 assemblies for the novel data sets, thus underlying the statistical powers of this benchmarking study.
The most successful tools were `GetOrganelle` and `Fast-Plast`, which are recommended to be used complementary because, as shown in figure 2.7, they succeed for most data sets compared to other assemblers and accomplish to satisfactory assemble chloroplast genomes where the other fails. If both of them fail it might be worthwhile to repeat the runs because other results could be expected as shown in the scatter plots of figure 2.5, especially `Fast-Plast` might be able to improve the previously reached sore. Only if both of them fail it might be, even though improbable, possible that `NOVOPlasty` performs a successful assembly. The other assemblers should be used with caution. While `chloroExtractor` might be good for a quick overview due to its relatively low demand in computational time FREUDENTHAL et al., 2019b; `Chloroplast assembly protocol`, `ORG.Asm` and `IOGA` are not recommended to be used as the primary tools in organelle genome assembly projects.

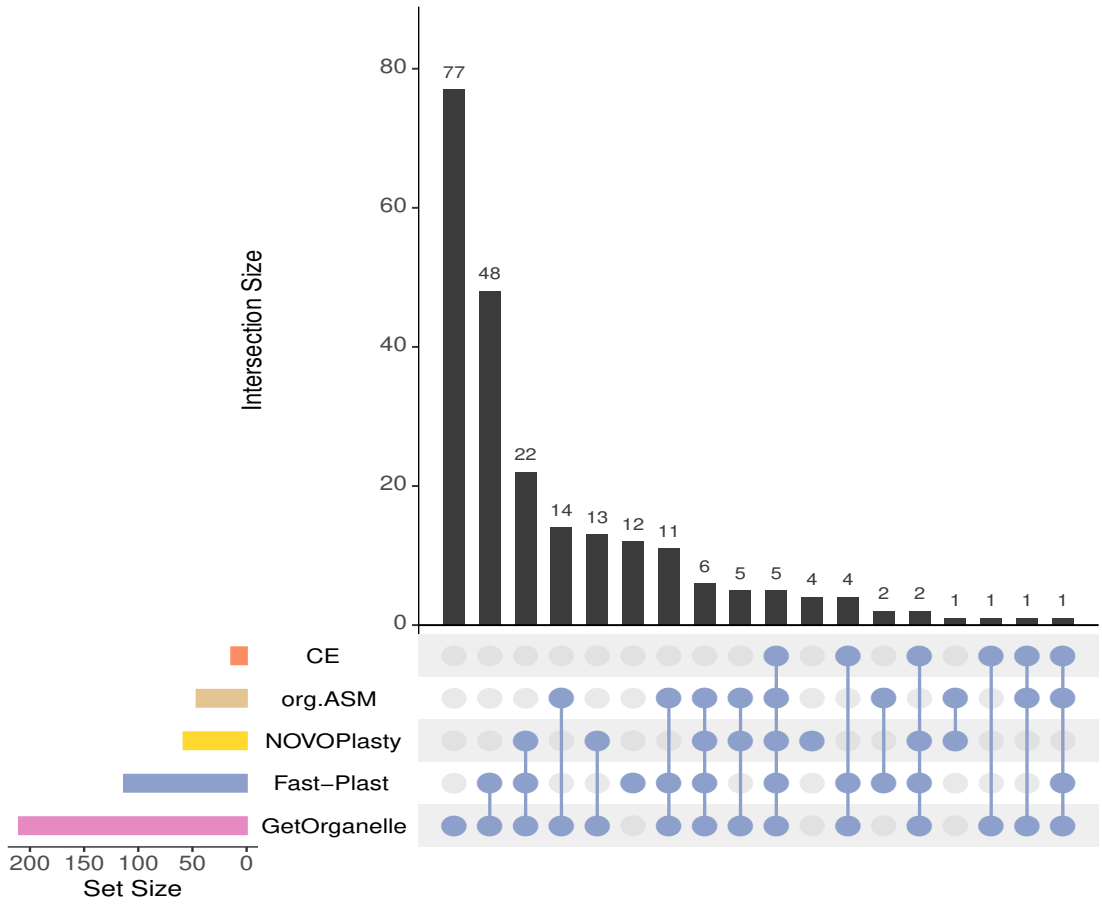


FIGURE 2.7: Upset plot showing the intersections of success rates between assemblers. A successful assembly was defined by a score >99 according to equation 2.1. The colored, horizontal barplot indicate the total number of successful assemblies for an individual tool. The black, vertical barplot gives the magnitude of the intersection between the assemblers indicated by the dots in the middle. Therefore the first and second vertical bars are to be interpreted as follows: 77 data sets were only successfully assembled by `GetOrganelle`, likewise 48 genomes were assembled completely by `GetOrganelle` and `Fast-Plast` and so on.

It might be possible that overall performance of a specific tool might change significantly by fine tuning the input parameters of the tool, which was purposely

not done in the scope of the present study because this study was designed to mimic the behavior of end-users and not developers of such tools. It was assumed that users with little experience in bioinformatics are inclined to use the basic configurations of such a tool.

While there are huge differences between all assemblers, they are presented with the same challenges and the bottlenecks are similar for all of them. However, the success rate of passing those differs. Figure 2.8 shows the alignment of the genomes, assembled with the seven tools, of *Oryza brachyantha*, a grass distantly related to cultivated rice *Oryza sativa*, and the respective reference genome. For the need of a linear representation of the circular genome the convention is to present chloroplast genomes in the order LSC - IRa - SSC -IRb. *O. brachyantha* was chosen because multiple tools successfully or at least almost assembled the full genome. `Chloroplast assembly protocol` is singled out, which only managed to assemble a few fragments on the SCC and the IRs on many contigs. A common mistake is to return three contigs as `IOGA` did. They represent the LSC the SSC and one IR, but failed to resolve those regions into a one circular contig. `GetOrganelle` and `Fast-Plast` were able to reproduce the structure of the reference, while `chloroExtractor` flipped the LSC and `NOVOPlasty` and `ORG.Asm` were not able to construct the single contig into the conventional structure. All of these are common mistakes appearing more or less rare in all the assemblers. This could be a good starting point for the developers to further improve their tools. In this example all but `Chloroplast assembly protocol` were able to construct all the parts of the chloroplast's genome. The most common error was to fail to resolve the structure of the genome into a circular, one contig version.

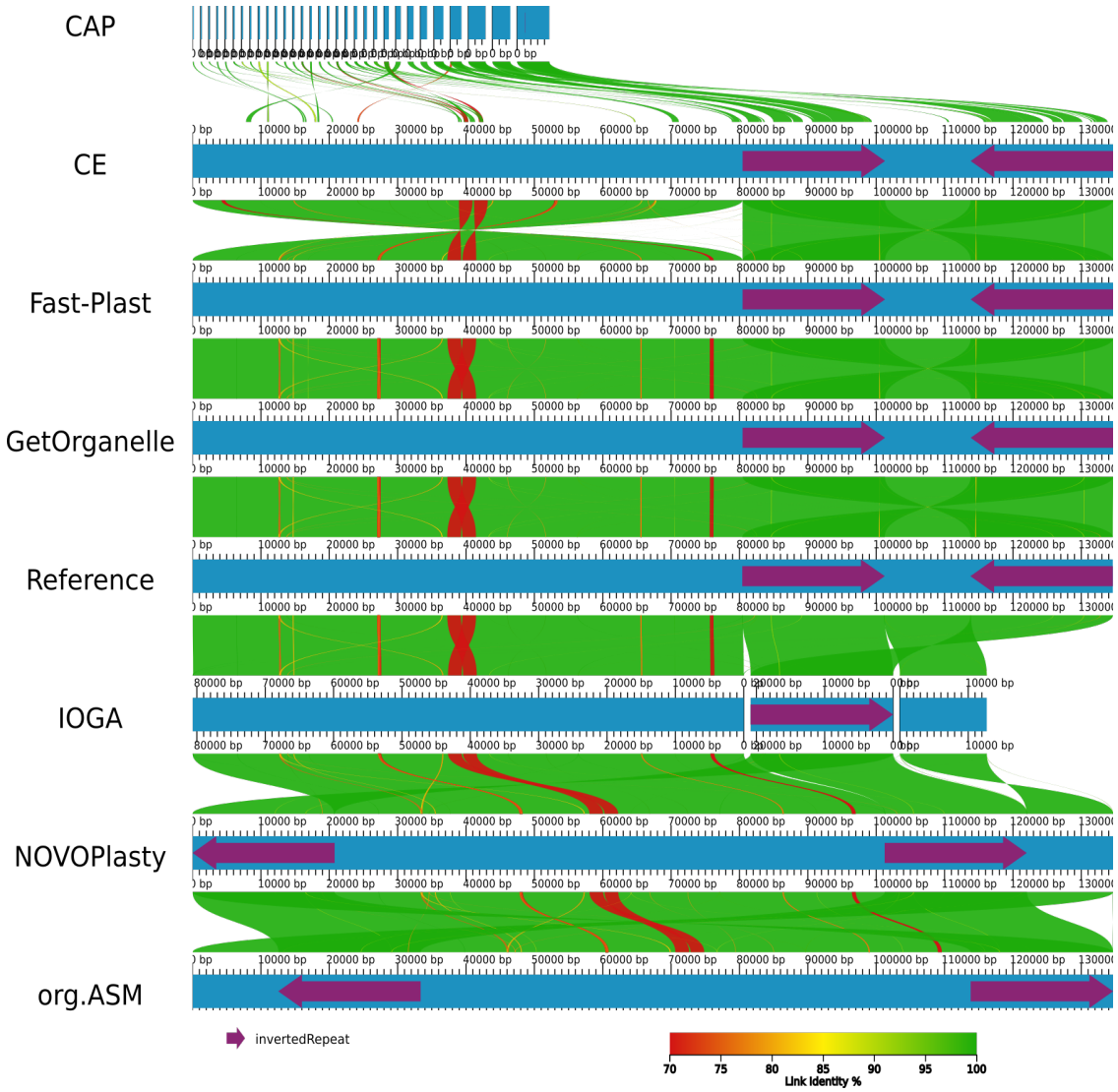


FIGURE 2.8: AliTV plot ANKENBRAND et al., 2017 from FREUDENTHAL et al., 2019b showing the alignments of *Oryza brachyantha* chloroplast genomes for all seven assemblers. Regions in adjacent assemblies are connected with colored ribbons. The color codes of for the similarity between regions. The purple arrows indicate the IR regions

2.5 Conclusion & outlook

Organelle genomics is a promising field in plant genetics. As described in section 2.1 chloroplast genomes are well-suited for applications in evolutionary sciences, taxonomy and barcoding applications. Alike for its mother branch core

genomics, for comparative chloroplast genomics it is just as crucial to obtain high quality genomes. The quality is mainly influenced by two factors: the quality of the genome sequencing protocol and the quality of the assembly. As shown the latter varies massively between tools and not all tools are recommend equally based on the conclusions drawn from the experiments described above. All tools have room for improvement. This is not meant to criticize the respectable work of the developers, but to encourage them to further develop tools and publish them under terms of liberal software licenses for the greater benefit of the entire scientific community.

3 GWAS-Flow a GPU-accelerated software for large-scale genome-wide association studies

The following chapter has been published in a similar version on the bioRχiv preprint server FREUDENTHAL et al., 2019a and has been submitted for peer review. The experiments and the software were designed and conducted by the author. The manuscript has been prepared by the author, with minor corrections from Prof. Arthur Korte & Prof. Dominik Grimm. All authors approved of the final manuscript.

3.1 Introduction

Genome-wide association studies, pioneered in human genetics HIRSCHHORN and DALY, 2005, have become the predominant method to detect associations between phenotypes and the genetic variations present in a population, in the last decade. Understanding the genetic architecture of traits and mapping the underlying genomic polymorphisms is of paramount importance for successful breeding, both in plants and animals, as well as for studying the genetic risk factors of diseases. Over the last decades the costs for genotyping have been reduced dramatically. Early GWAS consisted of a few hundred individuals, which have been phenotyped and genotyped on a couple of hundreds to thousands of genomic markers. Nowadays marker densities for many species easily exceed millions of genomic

polymorphisms. Albeit commonly SNPs are used for association studies, standard GWAS models are flexible to handle different genomic features as input.
The *Arabidopsis* 1001 genomes project features for example 1135 sequenced *A. thaliana* accessions with over 10 million genomic markers that segregate in the population ALONSO-BLANCO et al., 2016. Other genome projects also yielded large amounts of genomic data for a substantial amount of individuals, as exemplified in the 1000 genomes project for humans SIVA, 2008, the 2000 yeast genomes project or the 3000 rice genomes project LI, WANG, and ZEIGLER, 2014. Thus, there is an increasing demand for GWAS models that can analyze these data in a reasonable time frame.
One critical step of GWAS is to determine the threshold at which an association is termed significant. Classically the conservative Bonferroni threshold is used, which accounts for the number of statistical tests that are performed, while many recent studies try to set significance thresholds that are based on the false-discovery rate (FDR) STOREY and TIBSHIRANI, 2003. An alternative approach is to determine permutation-based thresholds CHE et al., 2014. Permutation-based thresholds estimate the significance by shuffling phenotypes and genotypes before each GWAS run, thus any signal left in the data should not have a genetic cause, but might represent model mis-specifications or uneven phenotypic distributions. Typically this process is repeated hundreds to thousands of times and will lead to a distinct threshold for each phenotype analyzed TOGNINALLI et al., 2017. The computational demand of permutation-based thresholds is immense, as per analysis not one but at least hundreds of GWAS need to be performed. Here the main limitation is the pure computational demand. Thus, faster GWAS models could easily make the estimation of permutation-based thresholds the default choice.

3.2 Methods

3.2.1 GWAS model

The GWAS model used for `GWAS-Flow` is based on a fast approximation of the linear-mixed-model described in KANG et al., 2010; ZHANG et al., 2010, which estimates the variance components σ_g and σ_e only once in a null model that includes the genetic relationship matrix but no distinct genetic markers. These components are thereafter used for the tests of each specific marker. Here, the underlying assumption is that the ratio of these components stays constant, even if distinct genetic markers are included into the GWAS model. This holds true for nearly all markers and only markers, which posses a big effect will alter this ratio slightly, where now σ_g would become smaller compared to the null model. Thus, the p-values calculated by the approximation might be a little higher (less significant) for strongly associated markers.

3.2.2 The GWAS-Flow software

The `GWAS-Flow` software was designed to provide a fast and robust GWAS implementation that can easily handle large data and allows to perform permutations in a reasonable time frame. Traditional GWAS implementations that are implemented using Python VAN ROSSUM and DRAKE JR, 1995 or R R CORE TEAM, 2019 cannot always meet these demands. We tried to overcome those limitations by using TensorFlow, a multi-language machine learning framework published and developed by Google ABADI et al., 2015. GWAS calculations are composed of a series of matrix computations that can be highly parallelized and easily integrated into the architecture provided by TensorFlow. Our implementation allows both the classical parallelization of code on multiple processors (CPUs) and the use of graphical processing units (GPUs).

`GWAS-Flow` is written using the Python TensorFlow API. Data import is done with

pandas McKinney, 2010 and/or *HDF5* for Python Collette, 2013. Preprocessing of the data (e.g filtering by minor Allele count (MAC)) is performed with *numpy* Oliphant, 2006. Variance components for residual and genomic effects are estimated with a function slightly altered from the Python package *limix* Lippert et al., 2014. The GWAS model is based on the following linear mixed model (LMM) that takes into account the effect of every marker with respect to the kinship:

$$Y = \beta_0 + X_i\beta_i + u + \epsilon, u \sim N(0, \sigma_g K), \epsilon \sim N(0, \sigma_e I) \tag{3.1}$$

From this LMM the residual sum of squares for marker i are calculated as described in 3.2

$$RSS_i = \sum Y - (X_i\beta_0 + I_i\beta_1) \tag{3.2}$$

The residuals are used to calculate a p-value for each marker according to an overall F-test, which compares the model including a distinct genetic effect to a model without this genetic effect:

$$F = \frac{RSS_{env} - R1_{full}}{\frac{R1_{full}}{n-3}} \tag{3.3}$$

Apart from the p-values that derive from the F-distribution, `GWAS-Flow` also reports summary statistics, such as the estimated effect size (β_i) and its standard error for each marker.

3.2.3 Calculation of permutation-based thresholds for GWAS

To calculate a permutation-based threshold essentially n repetitions ($n \geq 100$) are computed of the GWAS on the same data with the sole difference that before each GWAS phenotypic values are randomized. Thus any correlation between the phenotype and the genotype will be broken and indeed for over 90% of these analyses the estimated pseudo-heritability is close to zero. On the other hand, the

phenotypic distribution will stay unaltered by this randomization. Hence any remaining signal in the GWAS has to be of a non-genetic origin and could be caused by e.g. model mis-specifications. Now the lowest p-value (after filtering for the desired minor allele count) is taken for each permutation and the 5% lowest value is set as the permutation-based threshold for the GWAS.

3.2.4 Benchmarking

For benchmarking of `GWAS-Flow` data from the *Arabidopsis* 1001 Genomes Project ALONSO-BLANCO et al., 2016 was used. The genomic data used were subsets of the full data set containing between 10,000 and 100,000 markers. Subsets that exceed 100,000 markers were not included because there is a linear relationship between the number of markers and the computational time demanded, as all markers are tested independently. Phenotypic data for flowering time at ten degrees (FT10) for *A. thaliana* was used, downloaded from the AraPheno database SEREN et al., 2016. Down- and up-sampled sets were generated to obtain phenotypes for sets between 100 and 5000 accessions. For each set of phenotypes and markers 10 permutations were run to assess the computational time necessary.

All analyses have been performed with a custom R script that has been used previously TOGNINALLI et al., 2017, `GWAS-Flow` using either a CPU or a GPU architecture and *GEMMA* ZHOU and STEPHENS, 2012. *GEMMA* is a fast and efficient implementation of the mixed model that is broadly used to perform GWAS. All calculations were run on the same machine using 16 i9 virtual CPUs. The GPU version ran on an NVIDIA Tesla P100 graphic card. Additionally to the analyses of the simulated data, the times required by *GEMMA* and both `GWAS-Flow` implementations for $>$ 200 different real data sets from *A. thaliana* were compared, which also have been downloaded from the AraPheno database SEREN et al., 2016 and have been analyzed with the available fully imputed genomic data set of ca. 10 million markers, filtered for a minor allele count greater five.

3.3 Results

The two main factors influencing the computational time for GWAS are the number of markers incorporated in such an analysis and the number of different accessions, while the latter has an approximate quadratic effect in classical GWAS implementations ZHOU and STEPHENS, 2012. Figure 3.1 shows the time demand as a function of the number of accessions used in the analysis with 10,000 markers. Exponential increases in the time demand are clearly visible for the custom R implementation, as well as for the CPU-based `GWAS-Flow` implementation and *GEMMA*. The `GWAS-Flow` implementations and *GEMMA* clearly outperform the R implementation in general. For a smaller number of accessions `GWAS-Flow` is slightly faster then *GEMMA*.

For the GPU-based implementation the increase in run-time with larger sample sizes is much less pronounced. While for small ($< 1,000$ individuals) data there is no benefit compared to running `GWAS-Flow` on CPUs or running *GEMMA*, the GPU-version clearly outperforms the other implementations if the number of accessions increases.

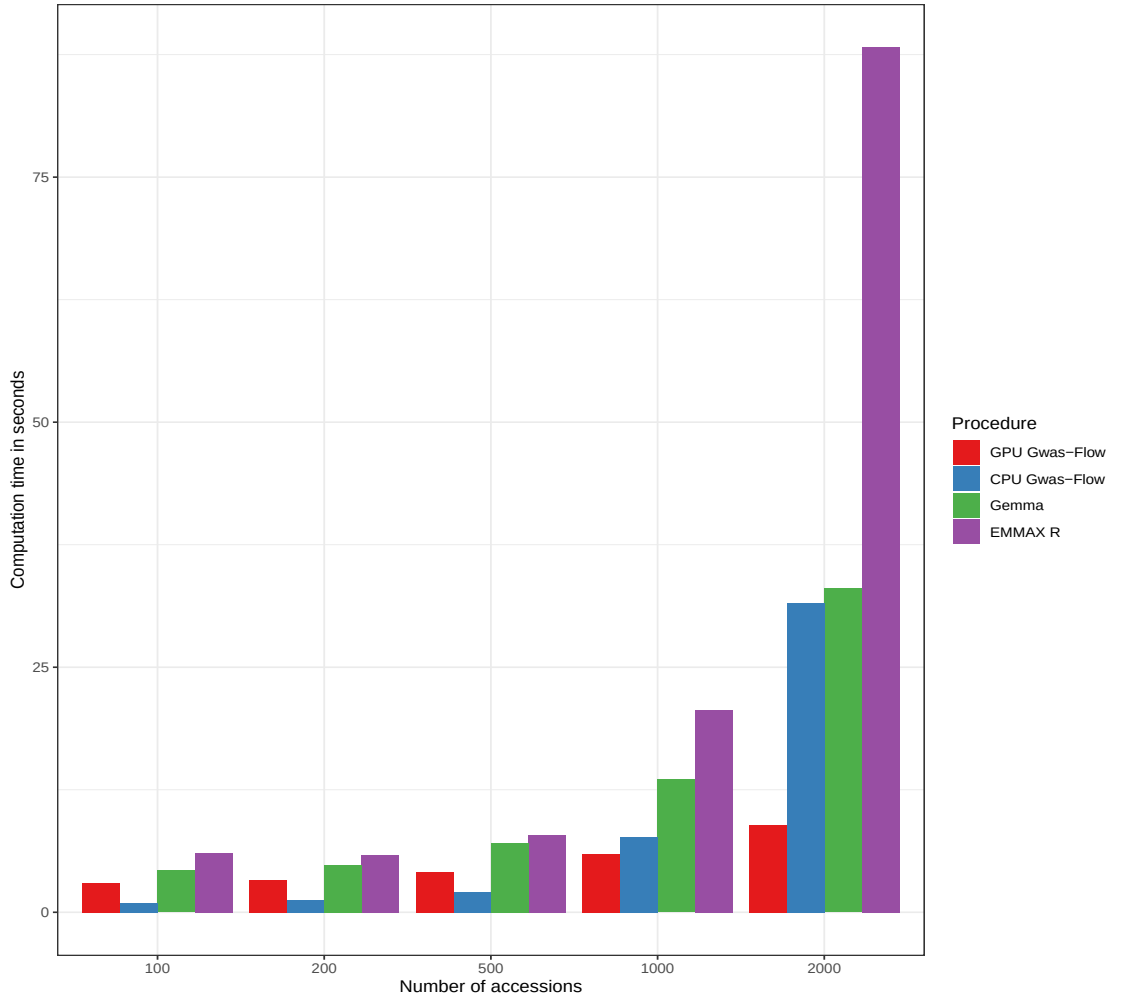


FIGURE 3.1: Computational time as a function of the number of accessions with 10000 markers each.

Figure 3.2 shows the computational time in relation to the number of markers with a fixed population of 2000 accessions for the two different `GWAS-Flow` implementations. Here, a linear relationship is visible in both cases.

To show the performance of `GWAS-Flow` not only for simulated data, both implementations were also run on more than 200 different real data sets of *A. thaliana*. Figure 3.3 shows the computational time demands for all analyses comparing both `GWAS-Flow` implementations to *GEMMA*. Here, the CPU-based `GWAS-Flow` performs comparable to *GEMMA*, while the GPU-based implementation outperforms both if the number of accessions is above 500. Importantly all obtained

GWAS results (p-values, beta estimates and standard errors of the beta estimates) are nearly (apart from some mathematical inaccuracies) identical between the three different implementations.

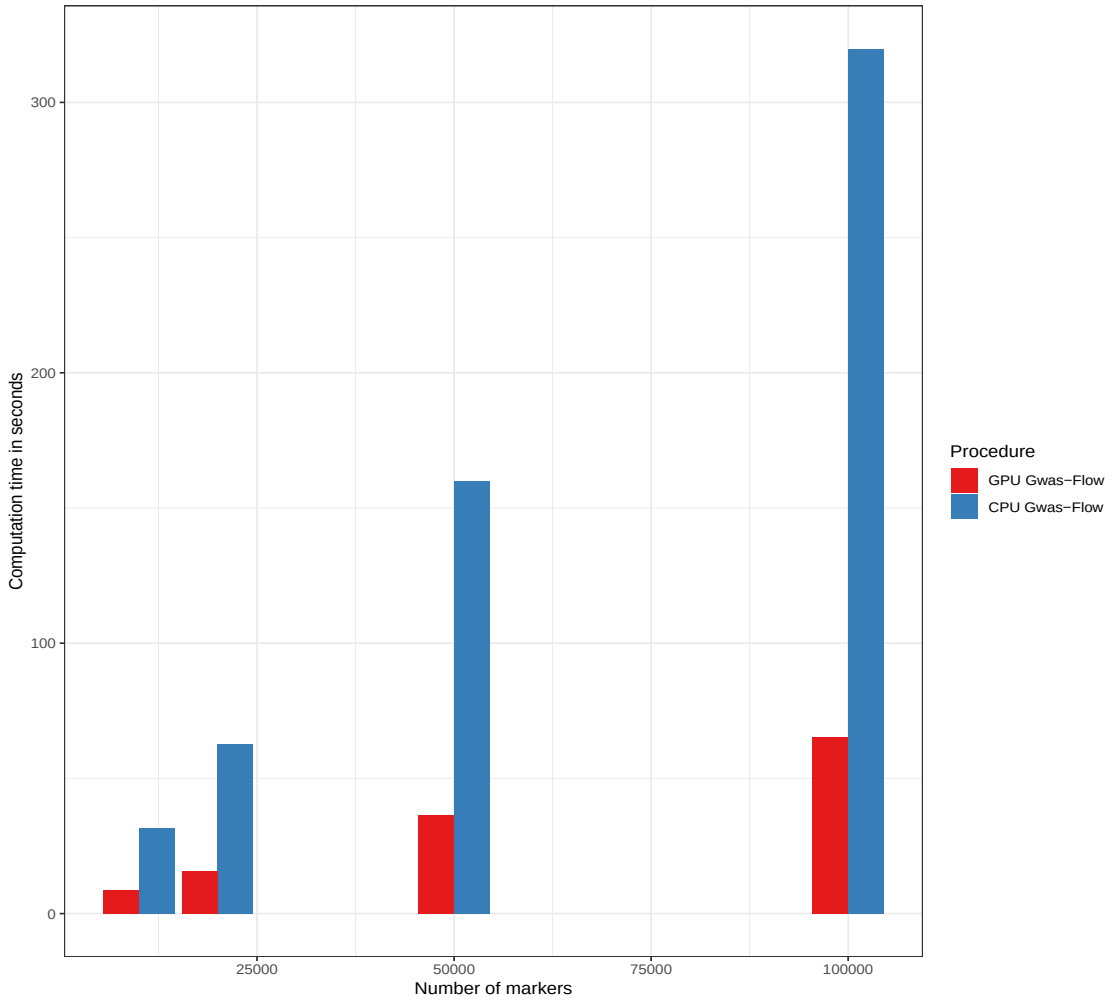


FIGURE 3.2: Computational time as a function of the number of genetic markers with constantly 2000 accessions for both `GWAS-Flow` versions

3.4 Discussion

To cope with the increasing computational demand in analyzing large GWAS data sets, recent developments of computational architecture and software were utilized to develop `GWAS-Flow`. With `GWAS-Flow` both a CPU- and a GPU-based

version of the classical linear mixed model, commonly used for GWAS, is provided. Both implementations outperform custom R scripts on simulated and real data. While the CPU-based version performs nearly identical compared to *GEMMA*, the GPU-based implementation outperforms both, if the number of individuals, which have been phenotyped, increases. For analyzing big data, here the main limitation would be the RAM of the GPU, but as the individual test for each marker is independent, this can be easily overcome programmatically.

The presented `GWAS-Flow` implementations are markedly faster compared to custom GWAS scripts and even outperform efficient fast implementations like *GEMMA* in terms of speed. This readily enables the use of permutation-based thresholds, as with `GWAS-Flow` hundred permutations can be performed in a reasonable time frame even for big data. Thus, it is possible for each analyzed phenotype to create a specific, permutation-based threshold that might present a more realistic scenario. Importantly the permutation-based threshold can be easily adjusted to different minor allele counts, generating different significance thresholds depending on the allele count. This could help to distinguish false and true associations even for rare alleles.

`GWAS-Flow` is a versatile and fast software package and currently is and will remain under active development to make the software more versatile. This includes e.g. to reach compatibility with TensorFlow v2.0.0 and to enable more data input formats, such as PLINK PURCELL et al., 2007. The whole framework is flexible, so it is easy to include predefined co-factors e.g to enable multi-locus models SEGURA et al., 2012 or account for multi-variate models like the multi-trait mixed model KORTE et al., 2012.

Standard GWAS are good in detecting additive effects with comparably large effect sizes, but lack the ability to detect epistatic interactions and their influence on complex traits MCKINNEY and PAJEWSKI, 2012; KORTE and FARLOW, 2013. To catch the effects of these gene-by-gene or SNP-by-SNP interactions, a variety of genome-wide association interaction studies (GWAIS) have been developed,

thoroughly reviewed in RITCHIE and VAN STEEN, 2018. Here, `GWAS-Flow` might provide a tool that enables to test the full pairwise interaction matrix of all SNPs. Although this would be a statistic nightmare, it now would be computationally feasible.

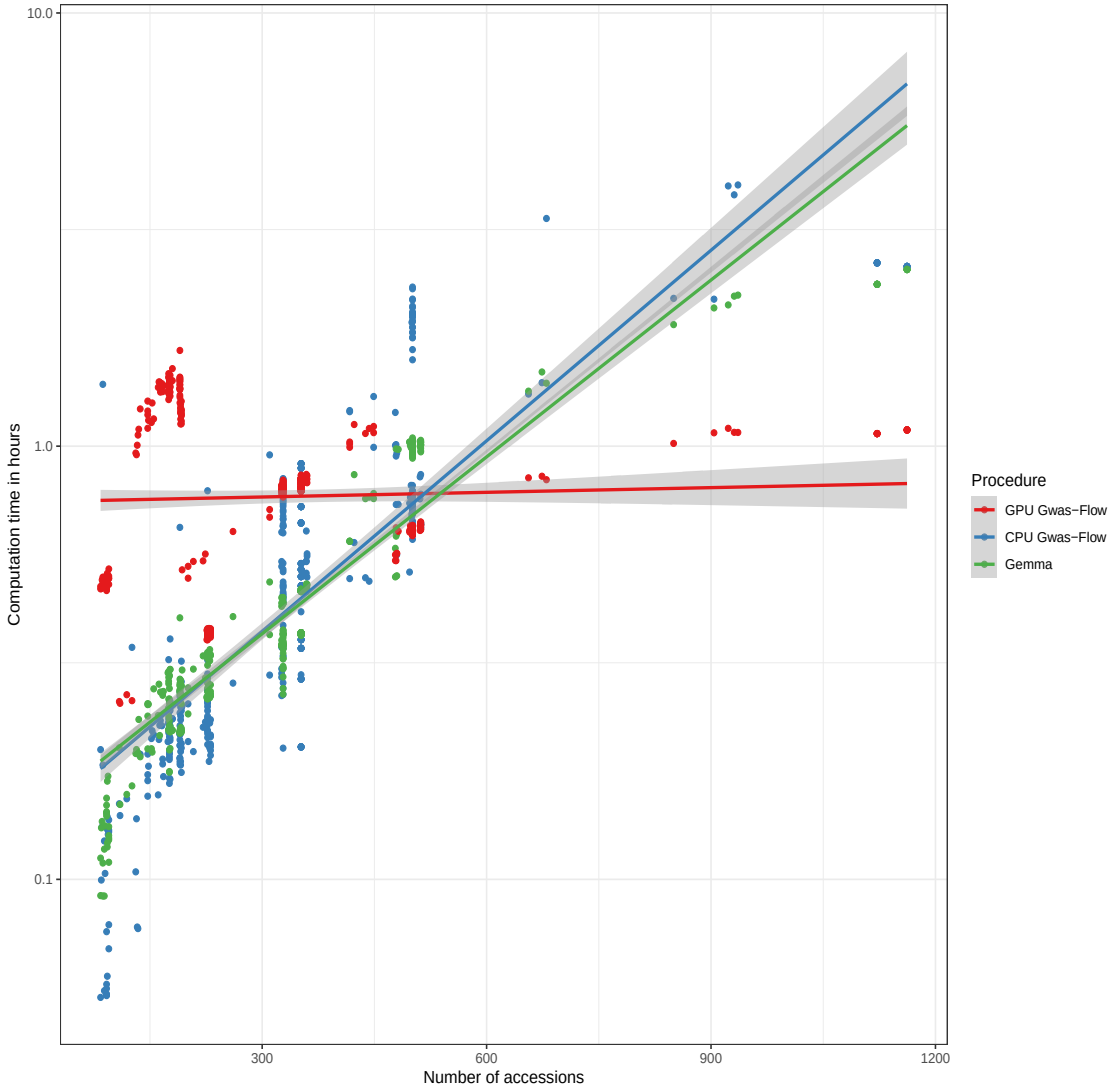


FIGURE 3.3: Comparison of the computational time for the analyses of $>$ 200 phenotypes from *Arabidopsis thaliana* as a function of the number of accessions for *GEMMA* and the CPU- and GPU-based version of `GWAS-Flow`. GWAS was performed with a fully imputed genotype matrix containing 10.7 M markers and a minor allele filter of MAC $>$ 5

4 Genomic prediction of phenotypic values of quantitative traits using artificial neural networks

4.1 Introduction to machine learning

4.1.1 The basic perceptron model

While machine learning, neural networks and deep learning became essential tools for many applications only in more recent years, their mathematical principals date back to the early 1950s and 1960s. Figure 4.1 schematically shows the basic perceptron model as proposed by Rosenblatt, one of the founders of machine learning, as the set of related statistical algorithms would be defined today. Rosenblatt designed his perceptron to mimic the information flow in biological nervous systems ROSENBLATT, 1961.

This basic perceptron, which contrary to perceptrons used nowadays, does not have an embedded activation function, takes n binary inputs $x_1, x_2....x_n$ and produces a single, likewise binary, output y after being processed. To achieve this Rosenblatt introduced the concept of weights, which determine a certain input's relative importance to the outcome of the output $w_1, w_2...w_n$. The output y is determined by the weighted sum of the weights $\sum_i w_i x_i$. If a certain threshold value is met the neuron is either activated and outputs 1 or not activated resulting

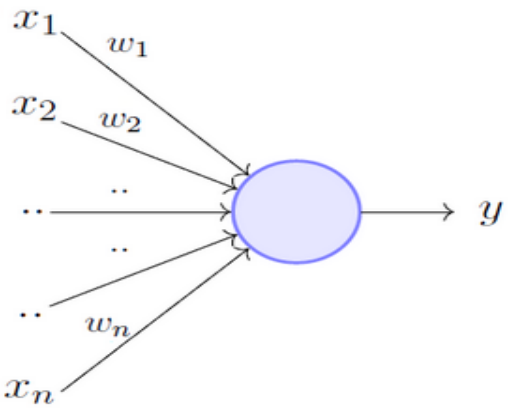


FIGURE 4.1: Basic perceptron model as proposed by Rosenblatt ROSENBLATT, 1961

in and output of 0. This is algebraically represented in equation 4.1:

$$0 = \text{if} \sum_{i}^{n} w_i x_i - \theta \leq 0 \tag{4.1a}$$

$$1 = \text{if} \sum_{i}^{n} w_i x_i - \theta > 0 \tag{4.1b}$$

Next to the weights w_n and the inputs x_n, a third term θ is introduced in equation 4.1, which represents the activation threshold. A single perceptron is a linear classifier and can only be trained on linearly separable functions and can be applied, as shown by ROSENBLATT, 1961, to solve simple logical operations as AND, OR and NOT. The basic perceptron fails, however, due to non-linearity to perform XOR operations, which was proven by MARVIN and SEYMOUR, 1969. This discovery led to a near stop in the research of artificial neural networks in the 1970s. That time period is now often referred to as the first AI-winter. Another reason that massively hindered the applications and research of machine learning during that span was the, compared to modern times, incredibly small amount of computational power available NGUYEN and WIDROW, 1990.

More complex decision making, like solving XOR problems, requires more complex structures than a single perceptron can provide. Continuing the trend of mimicking human neural networks, multiple artificial neurons were stacked into layers and these layers were connected to each other allowing communication between the many perceptrons in such a network. Figure 4.2 schematically shows the basic structure of an artificial neural network, now harboring three types of layers.

(i) the input layer

(ii) one or more hidden layers

(iii) the output layer, which in this case only consists of one neuron

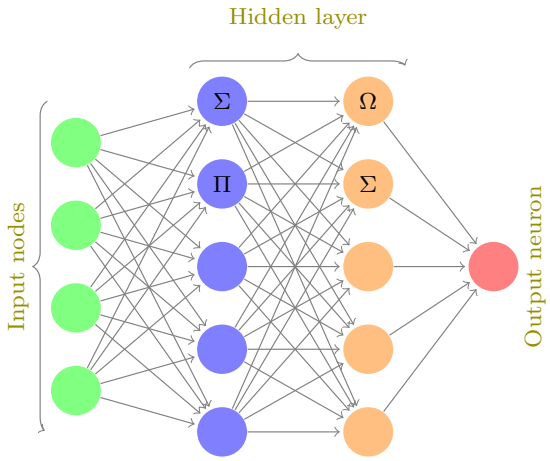


FIGURE 4.2: Schematic layout of a simple multi-layer perceptron

In the sample layout of figure 4.2 the neurons in the first column weigh the inputs and pass the gathered information to the neurons in the second layer. In the case above, neurons in the first layer are connected to all neurons on the second layer. Such layers are referred to as fully-connected layers (FLC) and their resulting networks are often called multi-layer perceptrons (MLP) or fully-connected networks. This architecture enables the network to perform more complex calculations resulting in more abstract decision making than single neurons or single

layer architectures.

There are other architectures, where neurons in the previous layer are only connected with neighboring neurons in the succeeding layers. Those are known as locally-connected layers (LCL). Related to them are convolutional layers, which share weights between selected neurons, building convolutional neural networks (CNN) LeCun et al., 1999.

4.1.2 Activation functions

The neurons discussed so far are only capable of outputting binary results, depending on whether threshold values are being reached or not. For more complex estimations it is desirable that small changes in the input also result in small changes of the output. This requirement cannot be easily met with binary outputs. Activation functions for a given node provide more sophisticated rules for the output in accordance to their inputs Žilinskas, 2006.

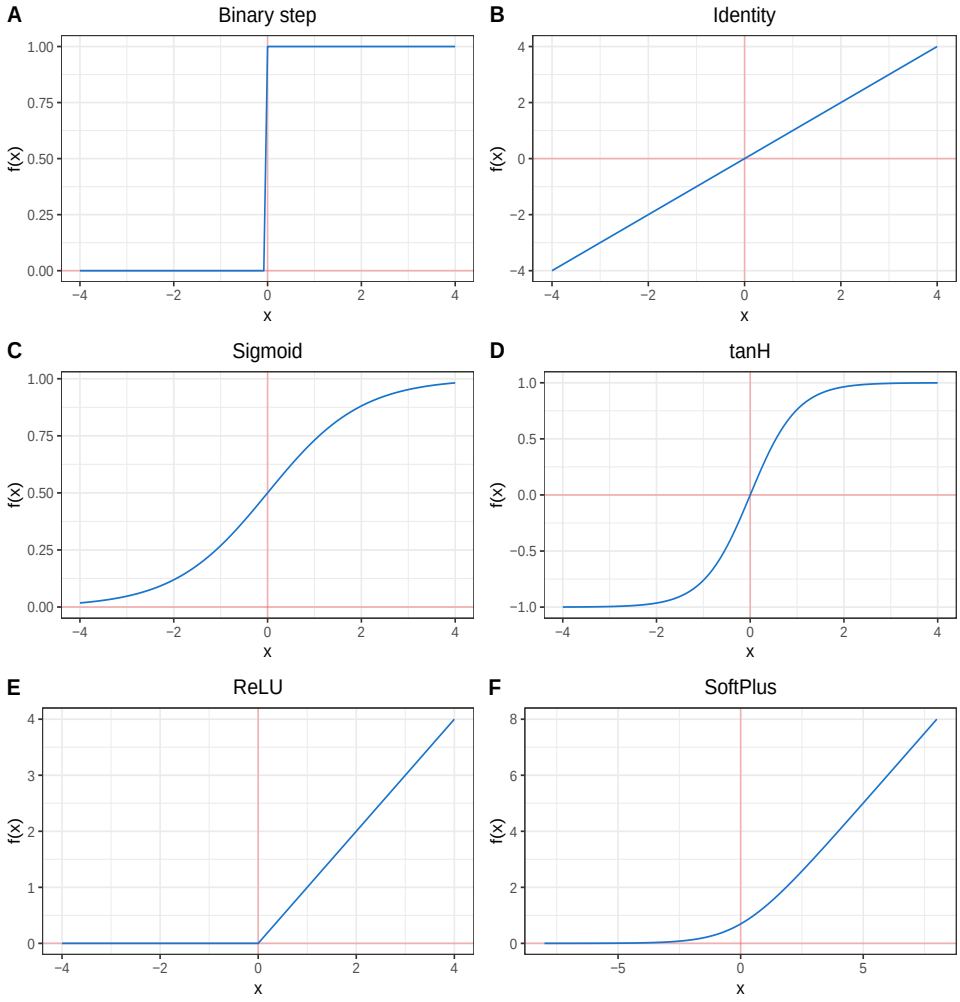


FIGURE 4.3: Popular activation functions used in neural networks.
A Binary step activation function
B Identity activation function
C Sigmoid or logistic activation function
D tangens hyperbolicus activation function
E rectified linear units activation function
F SoftPlus activation function

Figure 4.3 shows six of the most commonly used activation functions WARNER and MISRA, 1996. The simplest one, the binary step activation (**A**) was already introduced (see equation 4.2), whose properties have been discussed along the perceptron model. All other activation functions produce continuous outputs from given inputs.

Next to the binary step function any mathematical function is able to serve as an

activation function in neural nets, starting with a simple identity function (equation 4.3, figure 4.3 **B**). The sigmoid function (equation 4.4, figure 4.3 **C**) and tanh (equation 4.5, figure 4.3 **D**), when $x \to \infty$ or $x \to -\infty$ have similar properties as the binary function, but produce continuous output around threshold values of 0.

$$f(x) = \sigma(x) = \begin{cases} 0 \; for \; x < 0 \\ 1 \; for \; x \geq 0 \end{cases} \tag{4.2}$$

$$f(x) = \sigma(x) = x \tag{4.3}$$

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.4}$$

$$f(x) = \sigma(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.5}$$

$$f(x) = \sigma(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases} \tag{4.6}$$

$$f(x) = ln(1 + e^{x}) \tag{4.7}$$

Rectified linear units (ReLU) (equation 4.6) and SoftPlus (equation 4.7) share similar properties as well, the latter one being a smoothed version of ReLU. Rectifiers as activation functions have been introduced in the 2000s HAHNLOSER et al., 2000 and have since then overtaken all others as the most popular activation functions in neural networks and deep learning LECUN, BENGIO, and HINTON, 2015. They have proven to be superior in many deep-learning applications over sigmoid or logistic functions. One of the advantages leading to the superiority of ReLUs is that with randomly initialized weights only half of the ReLU neurons are activated at start compared to tanh and sigmoid activation GLOROT, BORDES,

and BENGIO, 2011. All activation functions shown in figure 4.3, but the binary step function, share one common property: a small change of the input weight will result in small changes of the output, while a small change of the input for the binary step function leads to either no or a complete change of the output, except for ReLU when $x < 0$. This property is, as described below, is an important prerequisite for networks being able to learn.

4.1.3 Gradient descent algorithm

Let a network alike the one shown in figure 4.2 be designed for the classification of an arbitrary, binary phenotype like petal color, being blue or not, with $x_1 \dots x_4$ on the input layers being genetic markers as features. The output layer displays values from 0 to 1 giving the probability of the petals being blue or not. To quantify how well the network performs on predicting the color of the petals a loss function is applied SCHMIDHUBER, 2015.

There is a large variety of different loss functions available for neural networks like mean squared error (MSE), root mean squared error (RMSE) and cross-entropy among others. In general MSE and RMSE are used for regression problems, with the latter being less popular, and cross-entropy also called log-loss is used for binary or multi-class classification settings JANOCHA and CZARNECKI, 2017. Since all problems presented in due course are regression problems that use MSE as their loss function this will be the only loss function further elaborated upon. MSE or the quadratic loss function can be written as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.8}$$

Equation 4.8 shows the MSE function, which is the sum of the squares of the differences of all the predicted $\hat{y}_i$ and the real values y_i. The same function can be

rewritten with the previously used terminology of weights and biases in equation 4.8 with $L(w, b)$ as the loss.

$$L(w, b) = \frac{1}{2n} \sum_{x} \|y(x) - y|^2. \tag{4.9}$$

With w and b as the collection of all the weights and the biases in the network used to optimize the function $y(x)$. Giving the quadratic nature of the function $L(w, b)$ will always be non-negative. If $L(w, b) \rightarrow 0$ the loss is minimal, meaning that the real and predicted values are close together and the network found weights and biases that approximate the output well.
A widely used algorithm to find the optimum of a loss function by finding its global minimum is gradient descent (GD) BOTTOU, 1991. The idea behind GD or other optimization algorithms is to start with randomly initialized weights and biases and repeatedly move them in direction Δw and Δb. This results in a change of the loss function, which can be represented using partial derivatives as shown in equation 4.10.

$$\Delta L = \frac{\partial L}{\partial w} \Delta w + \frac{\partial L}{\partial b} \Delta b \tag{4.10}$$

Ideally ΔL is negative and the optimization algorithm found Δw and Δb that lead to a reduction of the loss. To simplify this problem let Δd be the vector of changes: $\Delta d = (\Delta w, \Delta b)^T$ and ∇L the vector of the partial derivatives as in equation 4.11.

$$\nabla L = \left(\frac{\partial L}{\partial w}, \frac{\partial L}{\partial b} \right)^T \tag{4.11}$$

Having defined ∇L and Δd the term 4.10 can be simplified to:

$$\Delta C = \nabla L * \Delta d \tag{4.12}$$

Now the task of gradient descent is to find Δd that results in ΔC being negative as shown in equation 4.13

$$\Delta d = -\eta \nabla L \tag{4.13}$$

Here η is a small positive decimal number, commonly referred to as the learning rate, which usually, but not exclusively, ranges from 0.1 to 0.001. Having found a way to ensure that ΔL is always negative according to equation 4.13 it can be utilized to repeatedly update the gradient ∇L over time steps T. To make the gradient descent algorithm efficient the learning rate η has to be chosen correctly. If η is too large the gradient ΔL possibly ends up being larger than zero leading to an increase of the loss. If the learning rate is too small convergence will either take too long or not take place at all BERGSTRA et al., 2011. In practical machine learning approaches different learning rates are tested. There are also a variety of algorithmic approaches available to select the learning rate. Equation 4.11 only accounts for two inputs features but it can be generalized to compute n inputs as shown in equation 4.14.

$$\nabla L = \left(\frac{\partial L}{\partial w_1}, \ldots, \frac{\partial L}{\partial w_n} \right)^T \tag{4.14}$$

Equation 4.15 shows the gradient descent how it is used to repetitively update the weights and biases to optimize the loss function $L(w, b)$ with w and b as the weight and bias matrices and the learning rate η. In machine learning each iterational update of the network is often called epoch or training epoch.

$$w = w_i - \eta \frac{\partial}{\partial w} L(w) \tag{4.15a}$$

$$b = b_i - \eta \frac{\partial}{\partial b} L(b) \tag{4.15b}$$

Substituting the partial differentials with ∇L equation 4.15 a and b likewise simplifies to:

$$w = w_i - \eta \nabla L \tag{4.16}$$

4.1.4 Optimizers

The previous section introduced the concept of gradient descent, an algorithm to minimize the loss function of the weights and biases of a neural network. All other optimizers introduced in the following chapter are either variations or extensions of the basic gradient descent algorithm shown in equation 4.15.
One disadvantage of gradient descent is that if the data set grows larger the demand in memory for computation increases exponentially. Taking into consideration that machine learning is a popular method in big data applications this is a serious drawback.
Methods to solve that issue are stochastic gradient descent and mini batch gradient descent. The idea behind the latter is to randomly divide the entity of the training data into subsamples called mini batches BOTTOU and BOUSQUET, 2008. The network is then trained iteratively with all mini batches. The batch size has a significant influence on the accuracy and the training speed of the network and is a hyperparameter, which has to be tuned by iteratively testing different settings. If the batch size is one, mini batch GD is also referred to as stochastical gradient descent (SGD).
During the optimization process optimizers can descent into local minima of the loss function without being able to overcome them to reach the global minimum. An algorithm extending GD to accelerate the search of the global minimum is momentum, which allows the GD to speed up when the loss is decreasing and to carry on even when the loss function $L(w, b)$ is temporally increasing. This

is achieved by accounting for the gradient of the previous step in the calculation of the current step. This concept was introduced by POLYAK, 1964 and repopularized alongside the introduction of backpropagation learning by RUMELHART, HINTON, and WILLIAMS, 1988 an algorithm to efficiently update the weights and biases.

$$w = w_i - \eta \nabla L + \alpha \Delta w \tag{4.17}$$

Equation 4.17 shows how the momentum is mathematically represented in GD to update the weights w or likewise the biases. The delta of the weights multiplied by an coefficient α is the momentum. α usually ranges from 0.1 to 0.9 and is another parameter to be tuned for successful training. If the momentum is too small the GD will not be able to overcome local minima and if α is too large the loss functions tends to oscillate without ever finding an optimum LECUN, BENGIO, and HINTON, 2015.

For both the momentum and the learning rate it is impractical to maintain the same level during all training epochs. After each epoch the loss function is either closer or further away from its global minimum and depending on the distance to that minimum it is desirable to have larger or smaller learning rates and momenta. This can be achieved with naive approaches, for example using a step function to gradually decrease those values after each iteration or to utilize algorithmic approaches MICHIE, SPIEGELHALTER, and TAYLOR, 1994. There is a large variety of optimizers trying to find optimal values for α and η and till today this field is under active research GOODFELLOW, BENGIO, and COURVILLE, 2016. Popular among those are: RMSprop HINTON, SRIVASTAVA, and SWERSKY, 2012; Nesterov momentum DOZAT, 2016; Adadelta ZEILER, 2012; Adagrad RUDER, 2016 and Adam KINGMA and BA, 2014, with Adam being the most popular optimizer today.

Nesterov momentum is a slight change to the normal momentum algorithm, capable of having huge impacts in practical applications because it helps avoiding oscillations around the minimum by using intermediate information to adapt the momentum.
RMSProp - root mean square propagation - is a method aiming to adapt the learning rate algorithmically by choosing η for each iteration. Lastly, the wide-spread Adam optimizer combines both of the features of momentum and RMSProp and adapts the learning rate as well as the momentum iteratively KINGMA and BA, 2014.

4.1.5 Regularization parameters and overfitting

A common problem in machine learning is to over parameterize the model on the training data and losing the ability to generalize on validation data. This issue occurs because neural networks have hundreds of thousand of free parameters to be trained. Deeper neural networks even have billions or trillions of parameters. If training of the neural net continues for enough epochs eventually the loss function will approach a minimum. As $L(w, b) \to 0$ the initially drawn conclusion could mislead to assuming that training was successful. However, when trying to apply the trained network not only to the training data set (TRN) but to a testing data set (TST) or validation set (VAL) the loss of TST is very large and the accuracy of prediction of TST is accordingly small. This phenomenon is known as overfitting and a lot of fine tuning of hyperparameters is devoted to minimizing this effect TETKO, LIVINGSTONE, and LUIK, 1995. Figure 4.4 visualizes this during training GOODFELLOW, BENGIO, and COURVILLE, 2016 of a neural network.

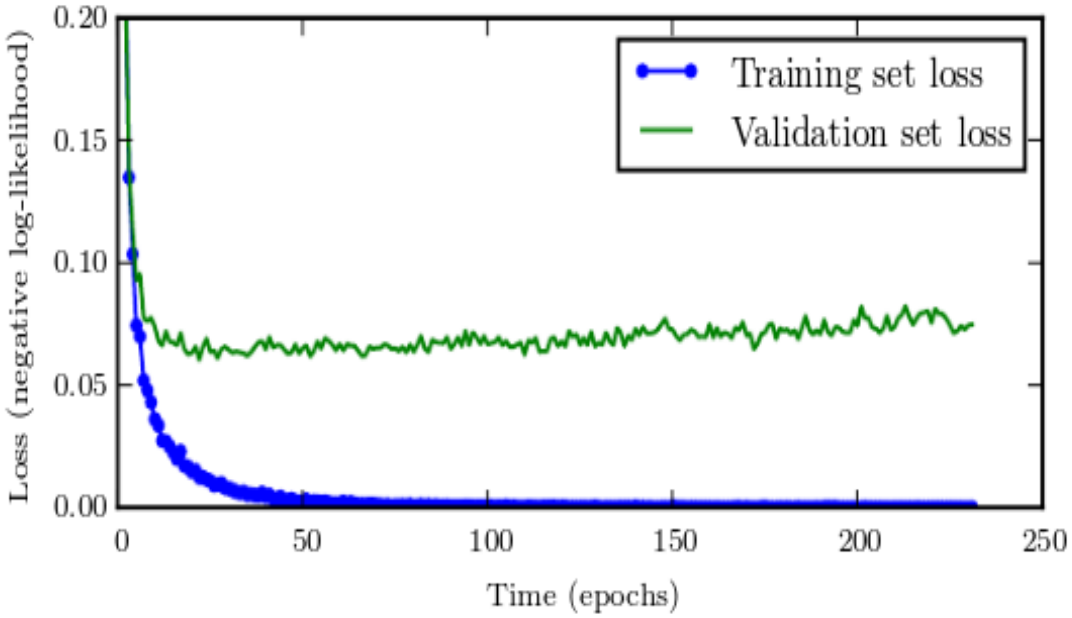


FIGURE 4.4: Learning curves showing how a loss function changes during training in the training and validation data set. While the training loss approaches 0 the validation loss starts increasing after hitting a minimum. This effect is due to overfitting on the training data set. Figure from GOODFELLOW, BENGIO, and COURVILLE, 2016.

Cross-validation

A method that is used in basically every training of neural networks is splitting up the data in multiple subsets. More specifically in a training set (TRN) and a testing set (TST). The training set is used to minimize the loss functions and its success is evaluated by comparison of the predicted values $\hat{y}$ and the real values y in TST. For all neural nets in this study Pearson's correlation coefficient was chosen as performance metric, calculated according to equation 4.18 SOPER et al., 1917.

$$\rho(y, \hat{y}) = \frac{cov(y, \hat{y})}{\sigma_y \sigma_{\hat{y}}} \tag{4.18}$$

There are other popular performance metrics, especially for classification problems, like AUC (area under the curve) and ROC (receiver operating characteristics), which evaluate the success of learning by weighing sensitivity and specificity.

In cross-validation compared to single validation the initial data set is split into TRN and TST multiple times, e.g. five times with a ratio of 80:20, and each TRN-TST pair is evaluated individually. Sometimes it is necessary to use a third subset - the validation data set. Because hyperparameter tuning is performed with the TRN and TST sets, a third portion of the data is needed to asses whether the neural network is able to generalize on global data or not.

4.2 Introduction to quantitative genetics and genome-based predictions

4.2.1 On the nature of quantitative traits

According to the omnigenic model, which is an extension of the polygenic model, proposed by BOYLE, LI, and PRITCHARD, 2017 and thoroughly reviewed in TIMPSON et al., 2018, all traits or phenotypic values are influenced by a great number or even all genes in the genome. This therefore results in traits following gradual statistical distributions instead of being binned in classes or binary.

Intuitively, this might be contradicting to the theoretical foundation of modern genetics - Mendel's three laws. They were derived from observations, which were mainly influenced by one locus. Using one of Mendel's phenotypes as an example - the round or wrinkled surfaces of peas (*Pisum sativum*)- an assessment of a couple of thousand peas would inevitably lead to the conclusion that from the "roundest" to the "wrinkliest" pea any gradual step between those two classes is possible and observable.

Mendel's third law of independent segregation also only holds true under certain assumptions. The simplest one being that the traits under investigation have to be located on different linkage groups, otherwise the seven traits used in Mendel's initial studies would not have segregated independently. The odds of seven randomly selected traits being on seven different linkage groups are rather small, especially taking into account that the genome of the *P. sativum* consists of only seven chromosomes itself KALO et al., 2004. Mendel most likely knew about traits not following his own laws, as well as being aware of the quantitative nature of traits, such as the constitution of a pea's surface or the color of its petals. However, being the pioneer of a then rather unexplored field of science, some of whose big questions we fail to satisfactory answer today, he did not have the resources or the knowledge to explain traits that were not "mendeling".

Initially thought to be contradicting to Mendel's ideas, Darwin proposed the concept of evolution due to natural selection, which introduced the idea of traits following gradual distributions DARWIN, 1859. This contrast led to a long lasting debate in the scientific community in the early 1900s between the Mendelians and the biometricians, who believed in the quantitative nature of continuous traits. This conflict has eventually been solved by Fisher's fundamental work published in 1919 FISHER, 1919. His theories combined the then in all fields of science popular research on statistical distributions and genomics. He mathematically proved that traits influenced by many genes, with randomly-sampled alleles, follow a continuous normal distribution in a population.

While this combined the ideas of Mendel and of the biometricians it opened another long debated question of effect sizes and the overall genetic architecture of complex traits. While in the theory of monogenic traits the effect size of a single gene on the trait is 100%, with an increasing number of genes influencing a complex trait the *per se* contribution of single genes has to decrease with an increasing number of loci determining the value of a given trait. Until the 1990s it has been believed that complex traits are predominantly controlled by few genes with a large to medium effect size, while others supposedly have a minimal influences ZHANG et al., 2018.

With the upcoming popularity of GWAS as the favored method to decipher genetic architectures of traits, it became clear that the majority of effect sizes are tiny (< 1%), while there are very few loci, which have a moderate effect on the phenotypic variance of a population with around 10% or less STRINGER et al., 2011; KORTE and FARLOW, 2013. This nature of quantitative traits presents great challenges to animal GODDARD and HAYES, 2009 and plant breeding WÜRSCHUM, 2012 in further improving crop or livestock performances, as well as complicating the decomposition of genomic causes for diseases like schizophrenia or autism in human medicine DE RUBEIS et al., 2014; PURCELL et al., 2014.

While the complex nature of the architecture of quantitative traits provides enough

challenges as is, all traits are also influenced by the environment surrounding the individual.
Therefore the distribution of trait values in a given population can be expressed as the addition of the variances of its genetic and the environmental effects 4.19.

$$\sigma_P = \sigma_G + \sigma_E \tag{4.19}$$

The genomic and the environmental effects do not only influence the phenotypic variance directly, but the environment also has an influence on gene expression, methylation of DNA bases etc. and therefore the equation 4.19 extends by the variance of the gene-environment interactions σ_{GxE} to equation 4.20 LYNCH and WALSH, 1998; WALSH and LYNCH, 2018b.

$$\sigma_P = \sigma_G + \sigma_E + \sigma_{GxE} \tag{4.20}$$

Equation 4.20 shows the decomposition of the phenotypic variance. To thoroughly understand the complex genetic architectures of traits the genetic variance needs to be decomposed further in its additive, dominance and epistatic components as in equation 4.21.

$$\sigma_G = \sigma_A + \sigma_D + \sigma_I \tag{4.21}$$

The additive effects are caused by single, for this model mostly homozygous, loci while the variance due to dominance effects is caused by heterozygous loci with their resulting interactions being full-, over- , co- or underdominant. Lastly, the interaction effects are a result of two or more genes only having an impact if they co-occur in a certain state. The resulting variance is commonly known as the gene-gene interaction or epistasis FALCONER and MACKAY, 1996.
Since possible interactions in a genome can appear between additive or dominant or a combination of those loci, the variance due to interaction effects σ_I can

be further dissembled into the variances resulting from additive-additive σ_{AA} dominant-dominant σ_{DD} and additive-dominant σ_{AD} interactions as shown in equation 4.22.

$$\sigma_I = \sigma_{AxA} + \sigma_{DxD} + \sigma_{AxD} \tag{4.22}$$

Knowledge of the variance components involved in the expression of a trait in a given population leads up to the estimation of the total influence of all genetic variances and the environmental variance on the phenotypic distribution. This concept is called heritability.
The heritability of a trait H^2 accounts for the proportion of the phenotypic variance controlled by the total genetic variance as shown in equation 4.23. This is also referred to as the broad sense heritability because all genetic effects, including additive, dominance and epistatic effects, are considered BROOKER, 1999.

$$H^2 = \frac{\sigma_A + \sigma_D + \sigma_I}{\sigma_P} \tag{4.23}$$

The concept of narrow-sense heritability 4.24 is similar to the broad-sense heritability, but only the additive genetic effects are included in the equation. This differentiation is important for natural and artificial selection and thus is commonly used in evolutionary genomics and breeding. Because in diploid species each parent only passes down a single allele of a given locus, dominance effects or interaction effects are not commonly inherited from one parent. Therefore mainly the additive genetic effects of a parent influence its offspring. While the dominance and epistatic variances are controlled by the combination of the parents FALCONER and MACKAY, 1996, WALSH and LYNCH, 2018b.

$$h^2 = \frac{\sigma_A}{\sigma_P} \tag{4.24}$$

4.3 Artificial selection in plant and animal breeding in the genomics era

4.3.1 Introduction to genomic selection

Genomic prediction has been applied to almost all relevant crop and model species. This includes among others:

A.thaliana; SHEN et al., 2013; HU et al., 2015.

Alfalfa (*Medicago sativa*) LI and BRUMMER, 2012; ANNICCHIARICO et al., 2015; LI et al., 2015; BIAZZI et al., 2017; HAWKINS and YU, 2018.

Barley (*Hordeum vulgare*) ZHONG et al., 2009; OAKEY et al., 2016; NEYHART, LORENZ, and SMITH, 2019.

Cassava (*Manihot esculenta*) ELIAS et al., 2018a; ELIAS et al., 2018b.

Cauliflower (*Brassica oleracea spp.*) THORWARTH, YOUSEF, and SCHMID, 2018.

Cotton (*Gossiypium spp.*) GAPARE et al., 2018.

Maize (*Zea mays*) RINCENT et al., 2012; WINDHAUSEN et al., 2012; TECHNOW, BÜRGER, and MELCHINGER, 2013; RIEDELSHEIMER et al., 2013; GUO et al., 2013; PEIFFER et al., 2014; TECHNOW et al., 2014; LEHERMEIER et al., 2014; OWENS et al., 2014; MONTESINOS-LÓPEZ et al., 2015; BUSTOS-KORTS et al., 2016a; KADAM et al., 2016; SCHOPP et al., 2017a;SCHOPP et al., 2017b; SOUSA et al., 2017; BRAUNER et al., 2018; SCHRAG et al., 2018; MOEINIZADE et al., 2019; ALLIER et al., 2019.

Potato (*Solanum tuberosum*) ENCISO-RODRIGUEZ et al., 2018; ENDELMAN et al., 2018.

Rape seed (*Brassica napus*) SNOWDON and INIGUEZ LUY, 2012; WÜRSCHUM, ABEL, and ZHAO, 2014; QIAN, QIAN, and SNOWDON, 2014; JAN et al., 2016; LUO et al., 2017; WERNER et al., 2018.

Rice (*Oryza sativa*) XU, 2013; GRENIER et al., 2015; HASSEN et al., 2018; MOMEN et al., 2019.

Rye (*Secale cerale*) BERNAL-VASQUEZ et al., 2014; WANG et al., 2014; AUINGER et al., 2016; MARULANDA et al., 2016; BERNAL-VASQUEZ et al., 2017.

Sugar beet (*Beta vulgaris*), WÜRSCHUM et al., 2013; BISCARINI et al., 2014.
Sugar cane (*Saccharum officinarum*) GOUY et al., 2013.
Soybean (*Glycine max*) JARQUIN, SPECHT, and LORENZ, 2016; XAVIER, MUIR, and RAINEY, 2016; STEWART-BROWN et al., 2019.
Switchgrass (*Panicum virgatum*) RAMSTEIN et al., 2016; POUDEL et al., 2019; RAMSTEIN and CASLER, 2019.
Wheat (*Triticum aestivum*) THAVAMANIKUMAR, DOLFERUS, and THUMMA, 2015; LOPEZ-CRUZ et al., 2015; SUKUMARAN et al., 2016; BUSTOS-KORTS et al., 2016b; GIANOLA et al., 2016; CROSSA et al., 2016; RINCENT et al., 2018; NORMAN et al., 2018; BELAMKAR et al., 2018; OVENDEN et al., 2018; CUEVAS et al., 2019a; HOWARD et al., 2019; KRAUSE et al., 2019.
As well as various tree species HOLLIDAY, WANG, and AITKEN, 2012; RESENDE et al., 2012; ZAPATA-VALENZUELA et al., 2013; JARAMILLO-CORREA et al., 2014; KUMAR et al., 2015; EL-DIEN et al., 2016; RATCLIFFE et al., 2017; RINCENT et al., 2018; KAINER et al., 2018; ALMEIDA FILHO et al., 2019.
Even though GS finds broad application in plant breeding it has been originally developed for the use in animal breeding HAYES and GODDARD, 2010; GODDARD, HAYES, and MEUWISSEN, 2011. The gold standard is a method known as genomic BLUP (GBLUP) VANRADEN, 2008, which utilizes a relationship matrix based on the co-occurrence of genetic markers. This method is derived from the pre-genomic era in animal breeding, where the relationship matrix was constructed after pedigrees according to the best linear unbiased predictors (BLUP) based on the linear mixed model equations developed by HENDERSON, 1975.
GBLUP accounts only for additive-genetic effects VANRADEN, 2008. There are other methods that are able to account for more complex genomic effects that are non-additive. Popular among those are for example Reproducing Kernel Hilbert Spaces (RKHS) GIANOLA and KAAM, 2008. Alternatively to Henderson's linear mixed models, a large variety of different Bayesian methods for genomic prediction became popular HAYES and GODDARD, 2001; GIANOLA et al., 2009; HABIER et

al., 2011; GIANOLA, 2013; CROSSA et al., 2017.

4.3.2 Genomic prediction in recurrent selection and the breeders equation

While the quantitative genetic methods breeders utilize are complex their goals can be defined in one sentence: genetically improve plant germplasms for agriculture. The breeding process started at the same time as farming around 10,000 BC in the region between the Euphrat and Tigris rivers known as the fertile crescent KINGSBURY, 2009. This changed the phenotypic appearance of the early crops dramatically to the point where they share little external traits with their wild ancestors. Those changes have been deeply carved into the genomes, which underwent serious alterations, including hybridization, duplications etc. This lead to most crop plants not having any wild ancestors with whom they could naturally mate. For example wheat (*T. aestivum*), one of the three most important sources of food on a global scale, underwent multiple hybridization steps OZKAN, LEVY, and FELDMAN, 2001. Wheat is a hybrid from either the diploid emmer (*T. diccoides*) or durum wheat (*T. durum*) and *Aegilops tauschii*, while emmer and durum are hybrids derived from wild emmer, which is a hybrid of a wild grass of the genus of *Aegilops* and *T. urata* FRIEBE et al., 2000; FELDMAN and LEVY, 2012.

While being ignorant of modern genetics early "plant breeders" must have had an intuitive, yet naive, understanding of the general concept of heritability in a way that they must have comprehended that offsprings share properties with their parents, which motivated them to regrow individuals with desired traits generation after generation. This induced many changes including that artificial selected plants are commonly largely inbred. That process could be considered an early form of recurrent truncation selection. Truncation selection on a normal distributed phenotype is shown in figure 4.5.

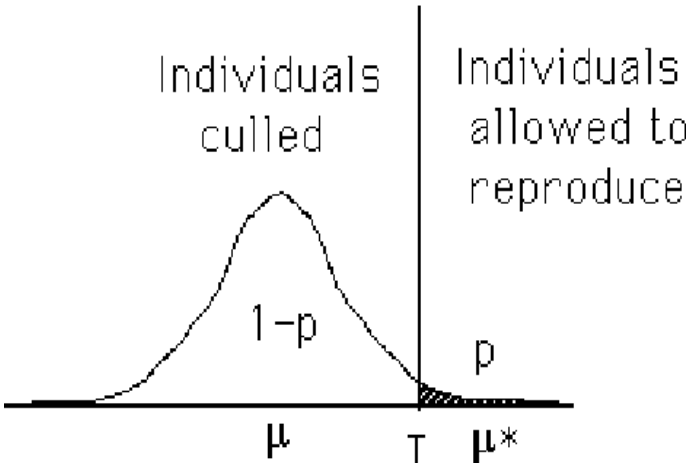


FIGURE 4.5: Truncation selection from a normal distributed phenotype with a selection threshold value of T, μ as the mean of the total population and μ^* as the mean of the selected phenotypes. Graphic from WALSH and LYNCH, 2018a

Like the early breeders, modern breeders have to determine a selection threshold T to divide the total population with the mean μ into two groups: the individuals culled and the ones allowed to reproduce with the mean μ^*. The difference between those two is the selection differential S:

$$S = \mu^* - \mu \tag{4.25}$$

In the case of normal distributed data as depicted in figure 4.5 S can be expressed as:

$$S = \varphi(\frac{T-\mu}{\sigma})\frac{\sigma}{p} \tag{4.26}$$

From which we can obtain the selection intensity i, which makes i solely a function of p.

$$i = \frac{S}{\sigma} = \frac{\varphi(z_{|1-p|})}{p} \tag{4.27}$$

With recurrent truncation selection over many generations the population

mean of the trait μ will change (hopefully in the desired direction) if the heritability (in this case the narrow sense heritability) $h^2 > 0$. It is impossible to breed for traits that do not contain any genetic components in their architecture WALSH and LYNCH, 2018b.

Next to i the selection intensity and the heritability h^2, the accuracy of the selection process r_{uA} is important for the success of a breeding program. Those three terms can be applied to estimate the gain of selection R over one generation (equation 4.28). Due to its importance in the evaluation of breeding schemes it is known as the breeder's equation MOUSSEAU and ROFF, 1987; FALCONER and MACKAY, 1996; KINGSOLVER et al., 2001.

$$R = ir_{uA}\sigma_A \tag{4.28}$$

The accuracy r_{uA} of equation 4.28 in cases when only phenotypic selection is conducted is the narrow-sense heritability and in cases where the selection process is aided by genomic prediction it is the prediction accuracy [1]. According to the breeder's equation there are three parameters, which can be influenced through genomic prediction.

(i) The prediction accuracy, which is usually smaller than the heritability, varies for different prediction equations and an increase in the accuracy will lead to an proportional increase in R. For this reason since 2001, in quantitative genetics one very active field of research was and still is to find new, better algorithms for GS as presented in the next chapter (4.3.3). As later evaluated on more than 150 phenotypes in chapter 4.8 h^2 is almost always larger than r_{uA}. Which if it was the only variable factor in equation 4.28, would make genomic selection inferior to phenotypic selection, which from

[1] The prediction accuracy in the literature is sometimes used synonymously with predictive ability, sometimes the predictive ability is defined as the prediction accuracy divided by the narrow or broad sense heritability. In the present study they will be used synonymously or termed $\rho_{(y,\hat{y})}$ as the correlation after Pearson between the real y and the predicted values $\hat{y}$

a certain point of view it is. Phenotypic trials are better approximations for phenotypic appearance as genomic estimated breeding values (GEBVs). However, as the cost of genotyping has decreased dramatically in the last 20 years, phenotyping with field trials remains tedious, laborious and vastly expensive. Taking into account that field trials have to be repeated in several years and locations to produce robust results, it becomes clear that genotyping 10s of thousands of accessions is much cheaper than conducting field trials with 1000 of them.

(ii) The selection intensity can be much stricter if the total population that is selected from is larger. In genomic prediction settings they are because breeders can select from two pools. First the pool of plants with known phenotypes and known genotype information and from those where just genomic data is available. When selecting from a pool of 1000 with $p = 0.05$ with the goal to keep 50 plants in the next breeding cycle, the same goal can be reached when genomically selecting from a pool of 10000 with and intensity of $p = 0.005$.

(iii) The decrease in time per generation is probably the largest advantage of genomic selection, when applied to breeding. While in field trials it is only possible to have one generation per year, genomic selection does not require the plants to be grown in the field. For GS it is only necessary to grow the plants large enough so that DNA can be extracted from the tissues and evaluated. After selection only the ones above the threshold are grown until they bear seats (or other reproductive organs) and be used for the next selection cycle, allowing up to ten generations per year. This development has lead to the rise to a new branch of breeding: speed breeding GHOSH et al., 2018; WATSON et al., 2018. In practical, company-level breeding, genomic prediction has largely contributed to an increase by a factor of 2 to

the gain in selection in recent years (personal communication with breeding company employees).

The last term in equation 4.28, the additive genetic variance σ_A, is not directly, yet heavily influenced by the described breeding scheme. Artificial selection has similar effects on the genetic variance as bottlenecks do in natural selection: it decreases, thus making it harder to increase R in later selection cycles WALSH and LYNCH, 2018b.

4.3.3 Genomic BLUP and Bayesian methods

All methods share a common statistical obstacle, which is commonly referred to as the $n >> p$ problem, which arises because the number n of markers is usually a multitude larger than the number of observations p. In practical applications it is not uncommon to have more than 100k markers while the number of phenotypes is no larger than 100. This does not allow to obtain genomic estimated breeding values (GEBV) by single marker regression as done by GWAS, which estimates highly inflated SNP-effects KORTE and FARLOW, 2013. One possibility is to include effect sizes as random effects and make prior assumptions about their distribution. The difference in prior distribution is the main distinction between the many methods of the Bayesian alphabet introduced in the following chapter GIANOLA, 2013.

Genomic BLUP

In the early years of research on genomic prediction, algorithms were not solely benchmarked against each other, but had to compete with the previously popular pedigree-based methods. Quickly in the course of the first decade of this millennium the superiority of the genomic methods were elucidated in livestock and plant breeding HABIER, FERNANDO, and DEKKERS, 2007; VANRADEN, 2008; VANRADEN et al., 2008; HARRIS, JOHNSON, and SPELMAN, 2009. While the genomic

methods are superior to non-genomic methods, there is no clear evidence that either of the genomic methods are superior to each other and there is lack of empirical evidence that the Bayesian methods generally outperform GBLUP MOSER et al., 2009; BERNARDO, 2010; AZODI et al., 2019.

Like pedigree BLUP for genomic BLUP the co-variance between related individuals is used for the predictions. In the latter case it is calculated from marker information. [2]

The general genomic prediction model (equation 4.29) is derived from mixed models HENDERSON, 1975; VANRADEN, 2008 and implemented as:

$$Y = X\beta + Zu + \varepsilon \tag{4.29}$$

where Y is a $n \ x \ 1$ vector of phenotypic observations, X the matrix of the fixed effects and β the vector of the fixed effects. Z is the incidence matrix for the combined marker effects and u is a $n \ x \ 1$ vector of the additive genetic effect the vector of the residuals ε.

To construct a GBLUP model lets assume a matrix of size $(n \ x \ m)$ with n individuals and m loci called M, containing marker information for three individuals on four loci, thus being of size $3x4$. The four markers of matrix 4.30 can take values of -1, 0 and 1, translating into minor allele, heterozygous locus and major allele. [3]

$$M = \begin{pmatrix} -1 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \tag{4.30}$$

[2] In the GWAS terminology the relationship matrix is referred to as K for kinship, while in GS circumstances it is called GRM (genomic relationship matrix) or abbreviated as G. This study will remain consistent with the circumstantial literature and therefore purposely inconsistent within itself. In the chapters addressing GWAS it will be called K for kinship matrix and in the following chapter elucidating GS it will be referred to as G.

[3] This example calculation has been adapted from ISIK, 2013.

The M matrix contains all the information that are necessary for the computation of the K matrix and other viable genetic parameters. The MM' matrix of size $n\ x\ n$ (4.31) bears additional parameters.

$$MM' = \begin{pmatrix} 3 & 1 & 2 \\ -1 & 1 & 0 \\ 2 & 0 & 3 \end{pmatrix} \tag{4.31}$$

The diagonal shows the number of homozygous loci per individual, while the other elements of the matrix indicate the number of markers shared by related individuals. This is an indicator for the distance of the relationship between individuals, as defined by identity-by-descent VANRADEN, 2008; MISZTAL et al., 2013. While matrix 4.31 calculates the metrics per individual, the $M'M$ matrix (4.32) accounts for metrics per marker. Likewise the diagonal contains the number of homozygous individuals per marker.

$$M'M = \begin{pmatrix} 3 & -1 & 0 & 0 \\ -1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix} \tag{4.32}$$

The next step is to obtain a matrix of the allele frequencies at each locus also of size $n\ x\ m$ like matrix M. For the design of matrix P (4.33) let the minor allele frequencies of the global population $p_1 \ldots p_4$ be $\{0.3, 0.2, 0.1, 0.15\}$. The allele frequency of the i^{th} column of P is expressed according to the n^{th} marker of matrix M as $P_i = 2(p_i - 0.5)$ resulting in:

$$P = \begin{pmatrix} -0.4 & -0.6 & -0.8 & -0.7 \\ -0.4 & -0.6 & -0.8 & -0.7 \\ -0.4 & -0.6 & -0.8 & -0.7 \end{pmatrix} \tag{4.33}$$

The allele frequencies, as in this simulated example, should be drawn from the

entire population and not only the subsample used for the calculation VANRADEN, 2008. The final step to obtain the Z matrix for the use in equation 4.29 is to subtract the P matrix from the M matrix $Z = M - P$ resulting in:

$$Z = \begin{pmatrix} 1.4 & 0.6 & 1.8 & -0.3 \\ -0.6 & 0.6 & 0.8 & 0.7 \\ 0.4 & 1.6 & 1.8 & -0.3 \end{pmatrix} \tag{4.34}$$

In Z the mean values of the allele effects are set to 0 and the subtraction of P emphasizes the effect of rare variants VANRADEN, 2008. There is a large variety of methods to generate the genomic relationship matrices and here lies the major difference between different genomic BLUP methods, but K is always of size $n\ x\ n$.

(i) The naive approach is to iterate over each individual and count the common markers with every other individual. This approach is suited for inbred or doubled-haploid populations, less so for outcrossed populations with high degrees of heterozygosity because as in the sample implementation it does only account for homozygous loci. This method becomes computationally intense when the data sets grow larger as common today (personal observation).

(ii) Probably the most popular method in GS is to obtain K as proposed by VANRADEN, 2008 designed after Wright's equations WRIGHT, 1922 for the covariance in structured populations, as described by equation 4.35 with Z as in 4.34.

$$G = \frac{ZZ'}{2\Sigma p_i(1 - p_i)} \tag{4.35}$$

In the above example this would result in the following kinship matrix:

$$G = \begin{pmatrix} 4.8 & 0.6 & 4.1 \\ 0.6 & 1.6 & 1.7 \\ 4.1 & 1.7 & 5.2 \end{pmatrix} \tag{4.36}$$

(iii) The unified additive relationship G_{UAR} according to YANG et al., 2010 and equation 4.37

$$G_{UAR} = A_{jk} = \frac{1}{N}\Sigma_i A_{ijk} = \begin{cases} \frac{1}{N}\Sigma_i \frac{(x_{ij} - 2p_i)(x_{ik} - 2p_i)}{2p_i(1-p_i)}, j \neq k \\ 1 + \frac{1}{N}\Sigma_i \frac{x_{ij}^2(1+2p_i)x_{ij} + 2p_i^2}{2p_i(1-p_i)}, j = k \end{cases} \tag{4.37}$$

where p_i is the allele frequency at locus i and x_{ij} the genotype for the j^{th} individual at the i^th locus. Another method also proposed by YANG et al., 2010 is to adjust G_{UAR} with β as in equation 4.38

$$G_{UARadj} = \begin{cases} \beta A_{jk}, & j \neq k \\ 1 + \beta(A_{jk} - 1), & j = k \end{cases} \tag{4.38}$$

with β as $\beta = 1 - \frac{c+1/N}{var(A_{jk})}$ to adjust for the bias in the estimation of the variance components, where c is the constant of a threshold for minor allele frequency.

(iv) Another approach is to weigh markers by the reciprocals of their expected variance according to the model 4.39. This was originally designed to investigate population structures in human genomics LEUTENEGGER et al., 2003; AMIN, VAN DUIJN, and AULCHENKO, 2007.

$$\begin{aligned} G &= ZDZ', with \\ D_{ii} &= \frac{1}{m|2p_i(1-p_i)|} \end{aligned} \tag{4.39}$$

(v) Other methods like the gaussian kernel compute kinship between individuals by the euclidean distance between the respective genotypes MOROTA and GIANOLA, 2014 as in equation 4.40.

$$\begin{aligned} K(x_i, x_j) &= exp(-\theta d_{ij}^2) \\ &= \prod_{k=1}^{m} exp(-\theta(x_{ik} - x_{jk})^2) \end{aligned} \tag{4.40}$$

with $d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{ik} - x_{jk})^2 + \cdots + (x_{im} - x_{j,a})^2}$ and $x_{ik} (i, j = 1, \ldots, n, k = 1, \ldots, m)$ and x_{ik} as the i^{th} individual at SNP k.

The linear model of equation 4.29 $Y = X\beta + Zu + \varepsilon$, with β as the vector of fixed effects and u as the vector of additive genetic effects, can be solved to obtain genomic estimated breeding values as:

$$\begin{pmatrix} X'X & X'Z & 0 \\ Z'X & Z'Z + G^{11} & G^{12} \\ 0 & G^{21} & G^{22} \end{pmatrix} \begin{pmatrix} \hat{b} \\ \hat{y}_1 \\ \hat{y}_2 \end{pmatrix} = \begin{pmatrix} X'y \\ Z'y \\ 0 \end{pmatrix} \tag{4.41}$$

with G^{12} as the part of G^{-1} containing individuals with phenotypic data and with G^{22} as the part of G^{-1} containing individuals without phenotypic data and just marker information available.

This can be algebraically solved to compute the GEBV of the unknown phenotypes $\hat{y}_2$ as:

$$\hat{y}_2 = -\left(G^{22}\right)^{-1} G^{21} \hat{y}_1 \tag{4.42}$$

GBLUP is fairly easy compared to more complex Bayesian methods and can be quickly implemented in any programming language capable of solving linear equations like R or Python R CORE TEAM, 2018; VAN ROSSUM and DRAKE JR, 1995. Computationally, as the number of phenotypes in the study increases in numbers,

the time demand grows exponentially because the kinship matrix quadruples in size and it becomes more complicated to compute the inverse of G (personal observations).

Bayesian methods

Next to the universal GBLUP a set of related algorithms became popular for solving the mixed models involved in genomic selection, known as the Bayesian alphabet GIANOLA et al., 2009; GIANOLA, 2013. They are all based on Bayes' fundamental theorem (equation 4.43).

$$P(\theta|y) = \frac{P(\theta)P(y|\theta)}{P(y)} \tag{4.43}$$

with $P(\theta)$ as the prior distribution, $P(y|\theta)$ as the likelihood and $P(y)$ as the marginal density of y. The prior distribution in GS assume that y was drawn from a certain distribution. Infinitesimal models assume that the genetic effects follow a normal distribution LEGARRA, LOURENCO, and VITEZICA, 2018, while the Bayesian frameworks, however, will assume non-normal distributed marker effects. This can be explained by a two-step hierarchical model. Stage one assumes that every marker has *a priori* a different variance LEGARRA, LOURENCO, and VITEZICA, 2018.

$$p(a_i|\sigma^2_{ai}) = N(0, \sigma^1_{ai}) \tag{4.44}$$

The second stage assumes prior distributions for the variances.

$$p(a_i|variable) = P(\dots) \tag{4.45}$$

with *variable* standing for the large variety of prior distributions. In total there are more than 20 different Bayesian models known to the author and probably

some more unknown ones. Their main difference "simply" lies in the *a priori* assumptions of prior distributions. This change can make some methods mathematically much more complicated then others. As shown in later chapters none of the methods is completely superior over others in terms of prediction accuracy.

Approximation to the solution of the linear equations is usually performed by Gibb's sampling using Markov Chain Monte Carlo (MCMC) simulations DE LOS CAMPOS et al., 2009; CAMPOS and RODRIGUEZ, 2016. Table 4.1 summarizes commonly applied Bayesian methods for genomic prediction indicating their key differences.

TABLE 4.1: Overview of properties of a variety of commonly applied Bayesian methods for genomic prediction. Table altered after KÄRKKÄINEN and SILLANPÄÄ, 2012

Name	Reference	Prior	Indicator	Hierarchy	Hyperprior	Estimation
BayesA	HAYES and GODDARD, 2001	Student	No	Yes	No	MCMC
BayesB	HAYES and GODDARD, 2001	Student	Yes	Yes	No	MCMC
BayesC	VERBYLA et al., 2009	Student	Yes	Yes	No	MCMC
BL	XU, 2010	Laplace	No	Yes	No	EM
BayesDπ	HABIER et al., 2011	Student	Yes	Yes	Yes	MCMC

The name is given by the author. The prior column tells which shrinkage prior is used.

4.3.4 Genomic selection using artificial neural networks

As mentioned in 4.3.1 genomic selection (GS) has been successfully applied in animal HAYES and GODDARD, 2010; GIANOLA and ROSA, 2015 and plant breeding CROSSA et al., 2010; HEFFNER et al., 2010; DESTA and ORTIZ, 2014; CROSSA et al., 2017 as well as in medical applications since it was first reported by HAYES and GODDARD, 2001. Since then the repertoire of methods for predicting phenotypic values has increased rapidly e.g. DE LOS CAMPOS et al., 2009; HABIER et al., 2011; GIANOLA, 2013; CROSSA et al., 2017. Genomic prediction has repeatedly been

proven to outperform pedigree-based methods CROSSA et al., 2010; ALBRECHT et al., 2011 and is nowadays used in many plant and animal breeding schemes. It has also been shown that using whole-genome information is superior to using only feature-selected markers with known QTLs for a given trait BERNARDO and YU, 2007; HEFFNER, JANNINK, and SORRELLS, 2011 in most cases. A more recent study AZODI et al., 2019 compared 11 different genomic prediction algorithms with a variety of data sets and found contradicting results, indicating that feature selection can be useful for some cases when whole genome regression is performed by neural nets.

While every new method is a valuable addition to the toolbox of genomic selection, some fundamental problems remain unsolved and are the same for every algorithm, of which the $n >> p$ problematic stands out. Usually in genomic selection settings the size of the training population (TRN) with p phenotypes is substantially smaller than the number of markers n FAN, HAN, and LIU, 2014, making the number of trainable features immensely large. Furthermore, every marker is treated as an independent observation neglecting collinearity and linkage disequilibrium (LD) between them. More difficulties arise through non-additive, epistatic and dominance marker effects. The main issue with epistasis in quantitative genetics is the almost infinite amount of different marker combinations, which cannot be represented within the size of TRN in the thousands. The same problems arises in GWA studies KORTE and FARLOW, 2013. With already large n the number of possible additive SNP-SNP interactions potentiates to $n^{(n-1)}$. Methods that attempt to overcome those issues are EG-BLUP, which use an enhanced epistatic kinship matrix and reproducing kernel Hilbert space regression (RKHS) JIANG and REIF, 2015; MARTINI et al., 2017.

In the past 10 years, due to increasing availability of high performance computational hardware with decreasing costs and parallel development of free easy-to-use software, most prominent being googles library TensorFlow ABADI et al.,

2016 and Keras CHOLLET, 2015, machine learning (ML) has experienced a renaissance. ML is a set of methods and algorithms used widely for regression and classification problems. Popular among those are e.g. support vector machines, multi-layer perceptrons (MLP) and convolutional neural networks. ML has been widely applied in many biological fields MAMOSHINA et al., 2016; RAMPASEK and GOLDENBERG, 2016; ANGERMUELLER et al., 2016; MIN, LEE, and YOON, 2017; LAN et al., 2018; WEBB, 2018.

A variety of studies assessed the usability of ML in genomic prediction OGUTU, PIEPHO, and SCHULZ-STREECK, 2011; GONZÁLEZ-CAMACHO et al., 2012; GONZÁLEZ-CAMACHO et al., 2016; QIU et al., 2016; MA et al., 2017; GONZÁLEZ-CAMACHO et al., 2018; GRINBERG, ORHOBOR, and KING, 2018; LI et al., 2018 MONTESINOS-LÓPEZ et al., 2019a; CUEVAS et al., 2019b; MONTESINOS-LÓPEZ et al., 2019b. Through all those studies the common denominator is that there is no such thing as a gold standard for genomic prediction. No single algorithm was able to outperform all the others tested in a single of those studies, let alone in all. While the general aptitude of ML for genomic selection has been repeatedly proven, there is no evidence that neural networks can generally outperform mixed-model approaches as GBLUP HAYES and GODDARD, 2001.

In other fields like image classification neural networks have up to 100s of hidden layers HE et al., 2016. The commonly used fully-connected networks in genomic prediction tend to have one to three hidden layers. With one layer networks often being the most successful among those. Contradicting to the idea behind machine learning in genomic selection one hidden layer networks will be inapt to capture interactions between loci and thus only account for additive effects. As shown in AZODI et al., 2019 convolutional networks perform worse than fully-connected networks in genomic selection, which again is contradicting to other fields where convolutional layers are applied successfully, e.g. natural language processing DOS SANTOS and GATTI, 2014 or medical image analysis LITJENS et al., 2017. Instead of using convolutional layers and fully-connected layers only, as

shown in Pook et al 2019, we also propose to use locally-connected layer in combination with fully-connected layers. While CL and LCL are closely related they have a significant difference. In CL weights are shared between neurons and in LCLs each neuron has its own weight. This leads to a reduced number of parameters to be trained in the following FCLs and should therefore theoretically lead to a decrease in overfitting. To evaluate the usefulness of machine learning in GS the data sets generated in the scope of the 1001 genome project of *A. thaliana* ALONSO-BLANCO et al., 2016 and the MAZE project were used.

4.4 Proof of concept for ANN-based genomic selection

Having established the quantitative architecture of traits in section 4.2.1 and the basics of machine learning and neural nets in section 4.1, that knowledge can be used to provide a proof of concept that neural networks are suitable for GP. Table 4.2 provides all the possible genotypes $G_1 \dots G_4$ that can be derived by two bi-allelic markers M_1, M_2 on a fictional haploid organism. In this simulation the effect sizes for each marker β_1 and β_2 are constant with a value of 1.

TABLE 4.2: Simple simulated phenotypes and genotypes for genomic prediction with genotypes $G_1 \dots G_4$, Markers M_1 and M_2 and phenotypes based on additive effects or *and*, *or*, *xor* logic gates.

	M_1	M_2	Y_{ADD}	Y_{AND}	Y_{OR}	Y_{XOR}
G_1	0	0	0	0	0	0
G_2	0	1	1	0	1	1
G_3	1	0	1	0	1	1
G_4	1	1	2	1	1	0

The four phenotypes Y_{ADD}, Y_{AND}, Y_{OR} and Y_{XOR} , which were derived from their respective marker effects, were used for GP. Y_{ADD} is a phenotype with

purely additive effects. So in the nomenclature introduced in chapter 4.2.1 $\sigma_A = \sigma_G$ and $\sigma_I = 0$. Since the hypothetical organism is haploid there are no dominance effects to be accounted for $\sigma_D = 0$ and since all the genetic effects are caused by additive effects and there are also no environmental effects σ_E. The narrow sense heritability h^2 - equation 4.24 - and the broad sense heritability H^2 - equation 4.23 - are equally 1. The other three phenotypes are based on epistatic effects σ_I, generated by passing the markers M_1 and M_2 through their respective logic gates. This theoretically results in $h^2 = 0$ and $H^2 = 1$ because there should be no additive effects. For y_{AND} ,however, $h \approx 0.5$ because there is a correlation between Y_{ADD} and Y_{AND}. In practical applications this allows methods like GBLUP, designed to account for additive genetic effects, to capture some of the epistatic effects of σ_I VIEIRA et al., 2017.

According to chapter 4.1 a single perceptron fails to solve *xor* gates. While a network with multiple nodes and layers should be able to overcome that deficit. A relatively simple neural network with two fully-connected hidden layers with 10 and 5 nodes was trained for the prediction of the phenotypes. To keep the simulation as simple possible, no regularization parameters like dropout etc. were included. The activation function was ReLU (4.6) with an Adam optimizer. The results of the prediction are shown in table 4.3.

TABLE 4.3: Results of genomic prediction from phenotypes and genotypes in table 4.2

	M_1	M_2	$\hat{Y}_{ADD}$	$\hat{Y}_{AND}$	$\hat{Y}_{OR}$	$\hat{Y}_{XOR}$
G_1	0	0	0.01	0.00	0.00	0.01
G_2	0	1	0.99	0.01	0.99	0.98
G_3	1	0	0.99	0.00	0.99	1.01
G_4	1	1	1.99	0.98	1.01	0.02

Not surprisingly, the simple network is able to solve all four problems and predict the phenotypes accurately. The task was rather easy because the training

data set and the testing data set were the same, but it served the purpose of showing that neural networks are generally apt to solve different marker interactions. *In natura* those interactions and the overall genetic architecture are much more complex. Effect sizes are not constant and epistasis may be caused by interactions with more than just two markers. With an increasing number of markers the number of possible two way interactions increases even more to 2^{n-1}. Smaller interaction effects could be obscured under larger additive effects. Gene-environment interactions might have a significant influence, resulting in a model that does not converge.

4.5 Data

Two different data sets were used for the genomic prediction trials. A set of doubled-haploid (DH) populations derived from maize landraces and an *A. thaliana* data sets with genomic data procured along the 1001 genomic project ALONSO-BLANCO et al., 2016 and various phenotypic trials SEREN et al., 2016.

4.5.1 DH populations derived from maize landraces

The DH populations were produced, propagated and phenotyped in the scope of the MAZE project phase I, funded by the Federal Ministry of Education and Research (BMBF) (Funding ID: 031B0195, project "MAZE") as well as the KWS SAAT SE, by various project partners at the Technical University of Munich, University of Hohenheim and the KWS. A thorough description of the germplasm selection and phenotyping was recently published by HÖLKER et al., 2019.
Modern maize cultivars are almost exclusively high-performing hybrids from two inbreed lines originating from different heterotic pools. Commonly hybrids are derived from a cross of European Flint and American Dent maize SANTOS DIAS et al., 2004; BRAUNER et al., 2019. Before hybrid breeding became the predominant method in maize breeding in the 1960s, landraces were propagated by farmers. Landraces are dynamic, open-pollinated, locally highly-adapted populations. They did not derive from modern breeding, but from locally confined selection and adaption by farmers to often very specific needs ARTEAGA et al., 2016. The hybrids grown today are derived from just a few landraces as founder lines, while the majority of landraces has been nearly forgotten. This and high intensity selection over many generation has led to a loss of genetic diversity σ_G in modern maize cultivars.
The landrace germplasm presents an important and essential stock of genetic variability for continuous success in maize breeding. The utilization of those germplasms would be impossible without the invaluable work of institutions like

the IPK Gartersleben, whose goal as genebanks is to maintain and store genetic material for long time periods. From the whole set of European landraces, three, representing large phenotypic and genetic heterogentiy, were chosen to be assessed in the scope of the MAZE project:

(i) Kemater Landmais Gelb (KE, Austria)

(ii) Petkuser Ferdinand Rot (PE, Germany)

(iii) Lalin (LL, Spain).

They represent 95% of the molecular variance in a set of 35 landraces analyzed in a preceding project by MAYER et al., 2017.
In total 1015 DH lines (516 KE, 432 PE, 67 LL) were produced with *in vivo* haploid induction with an inducer line as described in ROEBER, GORDILLO, and GEIGER, 2005.

Genomic maize data

The genomic maize data was provided by the TUM as described by HÖLKER et al., 2019.
Genotyping was performed with the 600k Affymetrix® Axiom® Maize array UNTERSEER et al., 2014. The markers were quality filtered and missing values were imputed individually for each landrace population using `Beagle 5.0` BROWNING and BROWNING, 2007; BROWNING, ZHOU, and BROWNING, 2018. After LD pruning and further quality control 29833 markers remained for 471 Kemater and 403 PE DHs. LL was excluded from further analyses due to insufficient amounts of genotypes.

Phenotypic maze data

The phenotype data was provided by the TUM as described by HÖLKER et al., 2019.

The traits were evaluated with lattice design in six different locations across Europe. Those traits were:

(i) early vigor (EV) at three different stages (V3, V4, V6)

(ii) plant height (PH) at two developing stages (V4,V6)

(iii) the final plant height (PH_final)

(iv) male flowering time: days till tasseling (DtTAS)

(v) female flowering time: days till silking (DtSILK))

To account for GxE best linear unbiased estimators (BLUE) were calculated according to Henderson's model HENDERSON, 1975 and used for further prediction. The BLUEs were calculated across all environments and for the DHs in the six environments individually, as explained by HÖLKER et al., 2019.

4.5.2 A. thaliana

Genomic data

The genomic data was generated during the course of the 1001 genome project of *A. thaliana* ALONSO-BLANCO et al., 2016 producing completely sequenced and assembled genomes for 1135 ecotypes. Combining them with a 250k marker data set for 1307 accessions HORTON et al., 2012, which partially overlaps with the fully-sequenced accessions resulting in a total of 2029 genotyped accessions, totaling in more than 10 mio. SNPs and Indels on the five chromosomes of *A. thaliana*. Imputation of missing data and upsampling of the 250k subsets was performed with `Beagle3` BROWNING and BROWNING, 2007. Those data sets were published alongside TOGNINALLI et al., 2019.

Phenotypic data

A complete list of the 164 phenotypes that are available on AraPheno can be found in Appendix B SEREN et al., 2016 of those 145 were included in this study. The phenotypic trials ranged from 100 to more than 1000 accessions per data set ATWELL et al., 2010; LI et al., 2010; MEIJÓ et al., 2014; STRAUCH et al., 2015.
For every one of the 145 phenotypes used for prediction, subsets of the marker matrix were sampled, LD pruned and MAF filtered. LD pruning was executed with the R-package SNPRelate ZHENG, 2013 with a relatively strict LD threshold of 0.65 and a $MAF > 10$. This resulted in data sets with approximately 150.000 markers for each phenotype.

4.6 Methods

The theoretical backgrounds of the methods used for genomic prediction were described in section 4.1 for the ANNs and section 4.3.3 for the Bayesian methods and GBLUP. The next sections are devoted to explaining how those methods were adapted and implemented for the prediction of the maize and *Arabidopsis* traits.

4.6.1 Validation scheme

The validation approach in this study was a little different than the commonly used five fold cross validation. All predictions were run 50 times with different splits of TST and TRN. For the full data sets randomly 20% were assigned to TST and 80% to TRN. This process was repeated 50 times reducing the chance of biases due to any TST-TRN combination being randomly more predictable for one or the other method. The validation scheme was generated *a priori* and stored in cross-validation files to allow reusing the validation sets.

4.6.2 ANN

The scripts for ANN based GP were written in Python using the lower level API TensorFlow ABADI et al., 2016 and the higher level API Keras CHOLLET, 2015 (appendix A. Both are very versatile, well-documented and are capable of performing a large variety of machine learning applications. For those reasons they are among the most used ML libraries. Another advantage is that they work well on GPUs, which allows ML algorithms to run in a reasonable amount of time compared to CPU-based calculations.

The markers of TRN served as the input layer for the network, while the phenotypes were the values trained upon in the output node. Preliminary trials showed that Adam is the superior optimizer for GS and hence was the only one further used. Likewise ReLU was the activation of choice being superior to sigmoid or other non-rectifiers. All the weights and the biases of the kernel were initialized with truncated normal distributed values. The loss function used was always MSE.

Having a few hyperparameters fixed the remaining ones were optimized via a grid search. For each training set multiple networks were trained to fine tune the input parameters. Those were the number of layers, the nodes per layer, the magnitude of the dropout, the type of dropout used, whether the first layer was locally-connected or fully-connected and the duration of training via the training epochs. This amounted to a total of almost 260000 trained networks for the 145 *A. thaliana* data sets alone.

After another set of preliminary runs, LCL as the first layer appeared to result in higher prediction accuracies than FLC and were henceforth exclusively used and applied with a stride length of 7. The stride length determines how many nodes of the input layer, in this case markers, are combined in the first hidden layer. The type of drop out used (alpha dropout, Gaussian noise or normal dropout) did not

show any effect therefore the normal dropout function was used further. The network's training was iterated over the different number of epochs, architectures, drop out values and the cross validation cycles, thus explaining the tremendous amount of total networks trained. Epochs from 5 to 60 in steps of 5 and several 1, 2 or 3 Layer architectures, following the locally-connected layer, were tested.

Single environment prediction

Next to the across environment BLUEs the single environment BLUEs were used for prediction to be able to gain insights into the structure of σ_{GxE} of the maize traits. This resulted in 2246 genotype x environment combinations for Kemater and 1975 for Petkuser with at least one data point. This number is lower than the maximum number of DHs per population times the six environments because, naturally, not all genotypes yielded reliable data in all the environments. Each DH x environment combination was treated as an individual for the across environment prediction. The marker matrix was enhanced with the environmental origin as cofactors as show in table 4.4 with one-hot encoded markers.

TABLE 4.4: Schematic representation of the enhanced genotype matrix for across environment prediction of maize phenotypes with DHs 1-2 with markers M 1-2 in environments E1-2

	M-1	M-2	E-1	E-2
DH1-E1	0	1	1	0
DH2-E1	1	0	1	0
DH1-E2	0	1	0	1
DH2-E2	1	0	0	1

4.6.3 GBLUP and Bayesian methods

The evaluation of the genomic BLUP and the Bayesian methods was performed with the R-package BGLR CAMPOS and RODRIGUEZ, 2016. To allow pairwise comparison of the individual validation runs the same validation schemes as for the

ANNs were used with the same TST and TRN sets. BGLR implements GBLUP as Bayesian ridge regression (BRR), which mathematically has the same results as GBLUP, but uses a Bayesian approach CAMPOS and RODRIGUEZ, 2016. For further comparison of the prediction methods, not only ANN and GBLUP were compared, but also five different Bayesian methods were applied to the maize data sets:

(i) BayesA

(ii) BayesB

(iii) BayesC

(iv) Bayesian Lasso (BL)

(v) BRR / GBLUP

Besides the actual prediction algorithms the number of markers and the number of accessions will influence the final accuracy. To assess the prediction accuracy as related to the number of markers, the full Petkuser genotype matrix was subsampled five times into 1k, 2k, 5k, 10k and 20k marker subsets. To analyze the prediction accuracy as a function of the number of accessions the Kemater data set was sampled into 50, 100, 200, 300 and 400 accession subsets randomly 10 times. Both trials were run with 50 fold validation with 80% TRN and 20% TRN.

4.7 Results

4.7.1 Results of *A. thaliana* prediction

Table 4.5 shows the results for genomic prediction for 145 *A. thaliana* phenotypes with ANNs and GBLUP and the architecture, determined via grid search, yielding the highest prediction accuracies.

TABLE 4.5: Prediction accuracies of *A. thaliana* phenotypes for GBLUP and ANN

Phenotype	GBLUP	ANN	Architecture	Epochs
FT16	0.8237	0.8215	100	10
2W	0.8156	0.8205	50, 30	35
FT10	0.8249	0.8191	48	50
LD	0.8128	0.8159	150	30
DTF sweden 2009 (1st experiment)	0.8063	0.8141	48	30
DTF sweden 2009 (2nd experiment)	0.8035	0.8091	50, 30	20
DTF sweden 2008 (2nd experiment)	0.7986	0.8057	150	25
4W	0.795	0.8052	50, 35, 15	30
FT22	0.8009	0.8043	150	15
DTF spain 2008 (2nd experiment)	0.7975	0.8032	150	40
LN16	0.7996	0.7999	50, 30	20
DTF spain 2009 (2nd experiment)	0.7917	0.7988	150	55
LDV	0.8158	0.7975	150	15
0W GH FT	0.7873	0.7942	50, 30	15
DTFmainEffect2009	0.7794	0.7855	50, 35, 15	35
SD	0.7905	0.7848	48	30
DTFplantingSummer2008	0.75	0.7746	50, 30	20
FT GH	0.7693	0.7702	50, 30	15
DTFlocSweden2009	0.7595	0.7626	50, 30	60
DTFplantingSummer2009	0.7521	0.7584	50, 30	50
0W	0.7488	0.7473	48	40
DTF spain 2009 (1st experiment)	0.7691	0.7425	48	40
DTF sweden 2008 (1st experiment)	0.727	0.728	50, 30	20
DTFlocSweden2008	0.7161	0.7271	50, 30	55
Seed Dormancy	0.7014	0.7241	50, 30	35
DTFmainEffect2008	0.7102	0.7142	50, 30	20
8W	0.7259	0.7083	150	50
LN22	0.7004	0.7069	50, 30	20
Size sweden 2009 (1st experiment)	0.6905	0.6994	48	50
LN10	0.6934	0.698	50, 30	20
DTF spain 2008 (1st experiment)	0.6944	0.677	150	25
SDV	0.6775	0.6728	150	15
8W GH FT	0.7001	0.6546	48	40
0W GH LN	0.6568	0.654	50, 30	20
Storage 7 days	0.6496	0.65	50, 30	25
Storage 28 days	0.6627	0.6483	50, 30	55
8W GH LN	0.671	0.6434	48	70
Size sweden 2009 (2nd experiment)	0.6114	0.6268	48	50
SizeLocSweden2009	0.6144	0.619	150	35
FLC	0.6118	0.6161	50, 30	30
LFS GH	0.6178	0.6136	150	35
FT Field	0.7324	0.6112	150	60
LY	0.6072	0.6088	150	60

Storage 56 days	0.6085	0.5788	150	15
LES	0.56	0.5764	150	50
M216T665	0.5155	0.5674	50, 30	50
LC Duration GH	0.5799	0.5664	150	55
M172T666	0.5165	0.5487	150	60
Trichome avg JA	0.588	0.5343	150	55
Secondary Dormancy	0.5184	0.5264	150	30
SizeMainEffect2009	0.52	0.5171	48	50
DSDS50	0.4754	0.5006	50, 30	60
avrPphB	0.5054	0.4942	150	60
Hypocotyl length	0.4934	0.4807	150	50
Size spain 2009 (1st experiment)	0.5121	0.4751	150	50
Yield spain 2009 (1st experiment)	0.5205	0.4719	50, 30	50
Leaf serr 10	0.4636	0.4683	150	55
Size spain 2009 (2nd experiment)	0.471	0.4623	48	50
Trichome avg C	0.4617	0.4385	48	40
Germ in dark	0.4447	0.4382	150	15
YieldMainEffect2009	0.505	0.4345	150	30
FT Diameter Field	0.5004	0.4274	150	15
Bacterial titer	0.5406	0.417	150	55
FRI	0.4011	0.4119	48	30
Rosette Erect 22	0.3973	0.3934	48	30
Area sweden 2009 (1st experiment)	0.4203	0.3895	50, 35, 15	30
Width 10	0.3932	0.3784	50, 30	60
Silique 22	0.4339	0.377	50, 30	50
avrRpt2	0.3757	0.3737	50, 30	30
M130T666	0.4381	0.3733	150	60
SizePlantingSummer2009	0.3769	0.3615	150	5
Area sweden 2009 (2nd experiment)	0.359	0.3542	48	45
FW	0.3397	0.3522	50, 30	25
P31	0.3632	0.3419	50, 30	45
MT GH	0.4016	0.3397	150	50
avrB	0.3304	0.3384	50, 30	30
avrRpm1	0.361	0.3368	50, 30	20
Seed bank 133-91	0.3446	0.3334	150	5
Mg25	0.5321	0.3288	50, 30	60
Leaf roll 10	0.3558	0.3272	48	40
Yield spain 2009 (2nd experiment)	0.4184	0.3197	20, 10	40
Noco2	0.3051	0.3174	48	30
Emwa1	0.3226	0.3124	50, 30	30
FT Duration GH	0.2659	0.3123	48	5
Leaf serr 22	0.3021	0.3108	150	60
Anthocyanin 10	0.3198	0.3107	50, 35, 15	60
Cd114	0.3345	0.3069	50, 30	50
Leaf serr 16	0.2895	0.3011	48	40
Fe56	0.2802	0.3006	150	35
YieldLocSweden2009	0.3431	0.2993	150	60

Width 16	0.3463	0.2983	150	50
Co59	0.2738	0.2953	50, 35, 15	25
K39	0.3036	0.2952	50, 30	60
Leaf roll 16	0.3072	0.2886	150	15
DTFplantingLoc2008	0.2971	0.275	50, 30	5
SizePlantingSummerLocSweden2009	0.2803	0.2704	50, 30	60
Mn55	0.2775	0.2662	50, 30	20
Anthocyanin 22	0.2731	0.2635	150	15
As75	0.254	0.2619	50, 30	35
Na23	0.2564	0.2598	50, 30	15
Ni60	0.2894	0.2539	150	25
Mo98	0.2765	0.2537	50, 30	35
Chlorosis 22	0.2622	0.2453	50, 35, 15	10
Hiks1	0.2441	0.2452	20, 10	20
Zn66	0.2553	0.2444	150	35
B11	0.2891	0.2392	48	40
Germ 16	0.2987	0.2356	50, 30	41
At2	0.2147	0.216	150	15
Emco5	0.166	0.2101	150, 30	20
Se82	0.2192	0.2075	150	25
Mature cell length	0.1987	0.2052	150	45
DW	0.2878	0.2048	50, 30	60
Yield sweden 2009 (1st experiment)	0.2274	0.2033	150	55
As2	0.1774	0.1962	150	15
Meristem zone length	0.1976	0.195	150	50
Germ 10	0.2073	0.1873	20, 10	40
Anthocyanin 16	0.2433	0.1867	20, 10	10
Width 22	0.2224	0.1856	50, 30	50
YieldPlantingSummerLocSweden2009	0.2146	0.18	150	55
DTFplantingSummerLocSweden2009	0.2032	0.1775	150	55
Bs	0.2161	0.1656	50, 30	60
Bs CFU2	0.1672	0.1584	50, 35, 15	15
Germ 22	0.1267	0.1533	50, 30	35
Leaf roll 22	0.1135	0.1511	48	45
RP GH	0.1755	0.1458	150	15
Cu65	0.1543	0.1315	150	5
Li7	0.1611	0.1297	150	60
As	0.1089	0.1227	100	20
At1	0.1473	0.1197	48	40
S34	0.1045	0.11	50, 30	60
YieldPlantingSummer2009	0.1265	0.0984	150	50
Silique 16	0.2366	0.0884	50, 30	60
Chlorosis 10	0.0243	0.088	50, 35, 15	55
Ca43	0.3333	0.0732	50, 35, 15	55
Seedling Growth	0.0813	0.0636	48	30
Vern Growth	-0.0096	0.0422	150	15
At2 CFU2	0.0694	0.0378	150	25

Yield sweden 2009 (2nd experiment)	0.0536	0.0355	150	25
As CFU2	0.0312	0.035	150	5
At1 CFU2	0.0818	0.0319	50, 30	50
Aphid number	-0.0246	0.029	50, 35, 15	10
After Vern Growth	-0.1433	0.0057	50, 35, 15	5
Chlorosis 16	-0.0313	-0.0121	150	5
As2 CFU2	0.0504	-0.0325	50, 30	60

Table 4.5 contains in total 145 phenotypes where both ANN and GBLUP yielded successful predictions. For 60 of 145 phenotypes ANNs were able to outperform GBLUP. However, when the overall prediction accuracies are generally high $\rho(y, \hat{y}) \geq 0.75$, 16 out 20 predictions yielded higher predictive abilities for the ANNs than GBLUP. At an intermediate level they perform both similar and at low levels $\rho(y, \hat{y}) < 0.30$ GBLUP appears to be better than the tested ANNs.
Figure 4.6 compares the average prediction accuracies after 50 validation folds for ANN and GBLUP for all the 145 phenotypes.

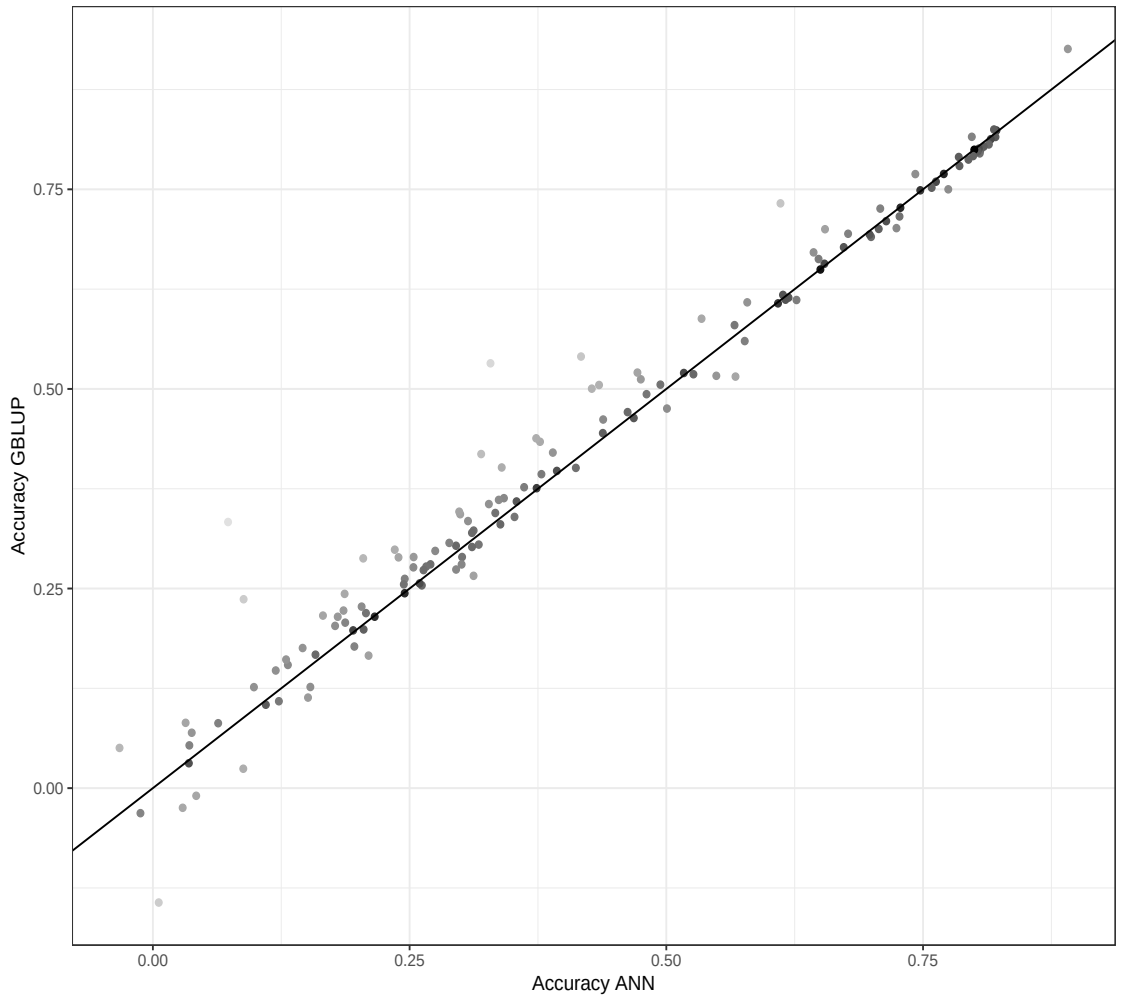


FIGURE 4.6: Scatterplot comparing prediction accuaracies of ANN and GBLUP in *A. thaliana*. Greyscale indicates the magnitude of the difference between the methods

Usually the prediction accuracies across methods are closely correlated with each other. Only for a few phenotypes there is a large difference in the prediction accuracies observable. For more than 100 of the prediction sets the difference in accuracies is smaller than 0.03. The ones with a difference in accuracies larger than 0.05 are among those with low general prediction accuracies, with extreme values of 0.2 and 0.15 going in either direction, however, with the majority of those with GBLUP being dominant over ANNs. Furthermore, also visible in figure 4.6, when the predictive abilities are high and ANNs perform better, the differences between the methods become insignificantly small.

4.7.2 Results of maize prediction

Across environments

For the prediction with the BLUEs across the six environments different results for the two sub-populations are observable. For the Kemater DH-population the ANNs can compete with and outperform GBLUP for all but the flowering time traits (DtTAS and DtSILK) as shown in the violinplots of figure 4.7.

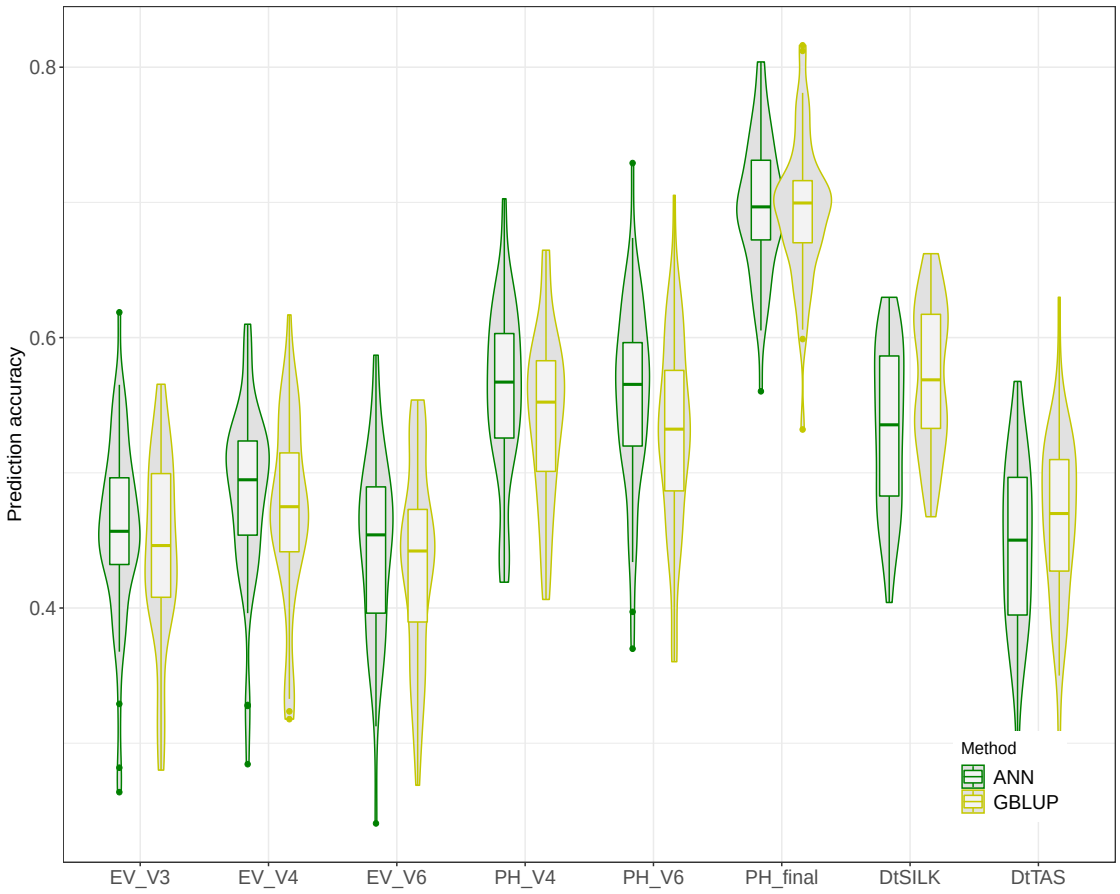


FIGURE 4.7: Violinplot comparing the results of genomic prediction in the doubled-haploid population Kemater for ANN and GBLUP for the early vigor (EV_V3, V4, V6) and plant height (PH_V4, V6, final) traits and days till silking (DtSILK) and days till tasseling (DtTAS).

While prediction for the Kemater subset shows that ANNs can perform reasonably well compared to GBLUP, the same observation cannot be reached for the Petkuser subset as shown in figure 4.8. Here GBLUP outperforms, even though

by a small margin, the tested ANNs for every trait. However, the overall prediction accuracies are smaller than for the Kemater subpopulation, which will be discussed in section 4.8. Additionally, the results for DtTAS were removed from the analyses with Petkuser, due to lack of sufficient data.

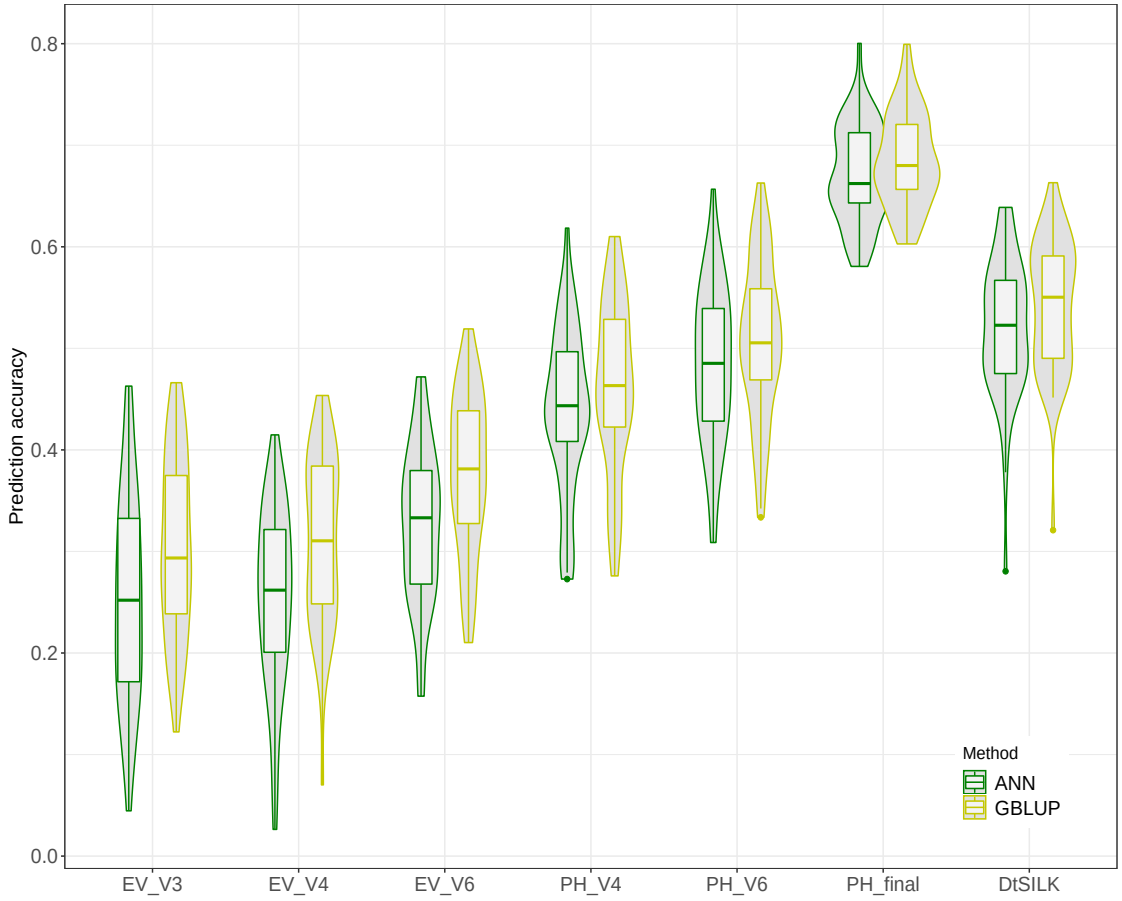


FIGURE 4.8: Violinplot comparing the results for genomic prediction in the doubled-haploid population Petkuser for ANN and GBLUP for the early vigor (EV_V3, V4, V6) and plant height (PH_V4, V6, final) traits and days till silking (DtSILK).

Table 4.6 compares the results of the predictions for Kemater and Petkuser. Similar to the *A. thaliana* traits there is a strong correlation between the prediction algorithms. If the overall prediction accuracy is below 0.40, like it is for the early vigor traits in Petkuser, the ANN begin to struggle to find optimal solutions for the networks. With an increasing overall accuracy the ANNs are comparable to GBLUP or better.

TABLE 4.6: Prediction accuracies of maize phenotypes for the doubled-haploid populations Kemater and Petkuser and the early vigor (EV_V3, V4, V6) and plant height (PH_V4, V6, final) traits and days till silking (DtSILK) and tasseling (DtTAS).

	Kemater		**Petkuser**	
Phenotype	GBLUP	ANN	GBLUP	ANN
EV_V3	0.44	0.46	0.31	0.25
EV_V4	0.47	0.49	0.31	0.25
EV_V6	0.43	0.44	0.38	0.33
DtTAS	0.47	0.44		
PH_V4	0.54	0.56	0.46	0.44
PH_V6	0.53	0.56	0.51	0.48
PH_final	0.69	0.70	0.68	0.67
DtSILK	0.57	0.53	0.54	0.52

Single environment prediction

The prediction of the single environment BLUEs with the environmentally enhanced marker matrix yielded substantially higher prediction accuracies than the prediction with the across environment BLUEs (previous section). The gain is higher if the prediction accuracies previously have been lower, which is an indicator for large GxE interactions. Figure 4.9 **A** compares the results for within and across location prediction for the Kemater DHs and **B** for the Petkuser population. The overall gain of adding the environmental information to the marker matrix is higher for Petkuser, where the prediction accuracies with the across environment BLUEs have been smaller.

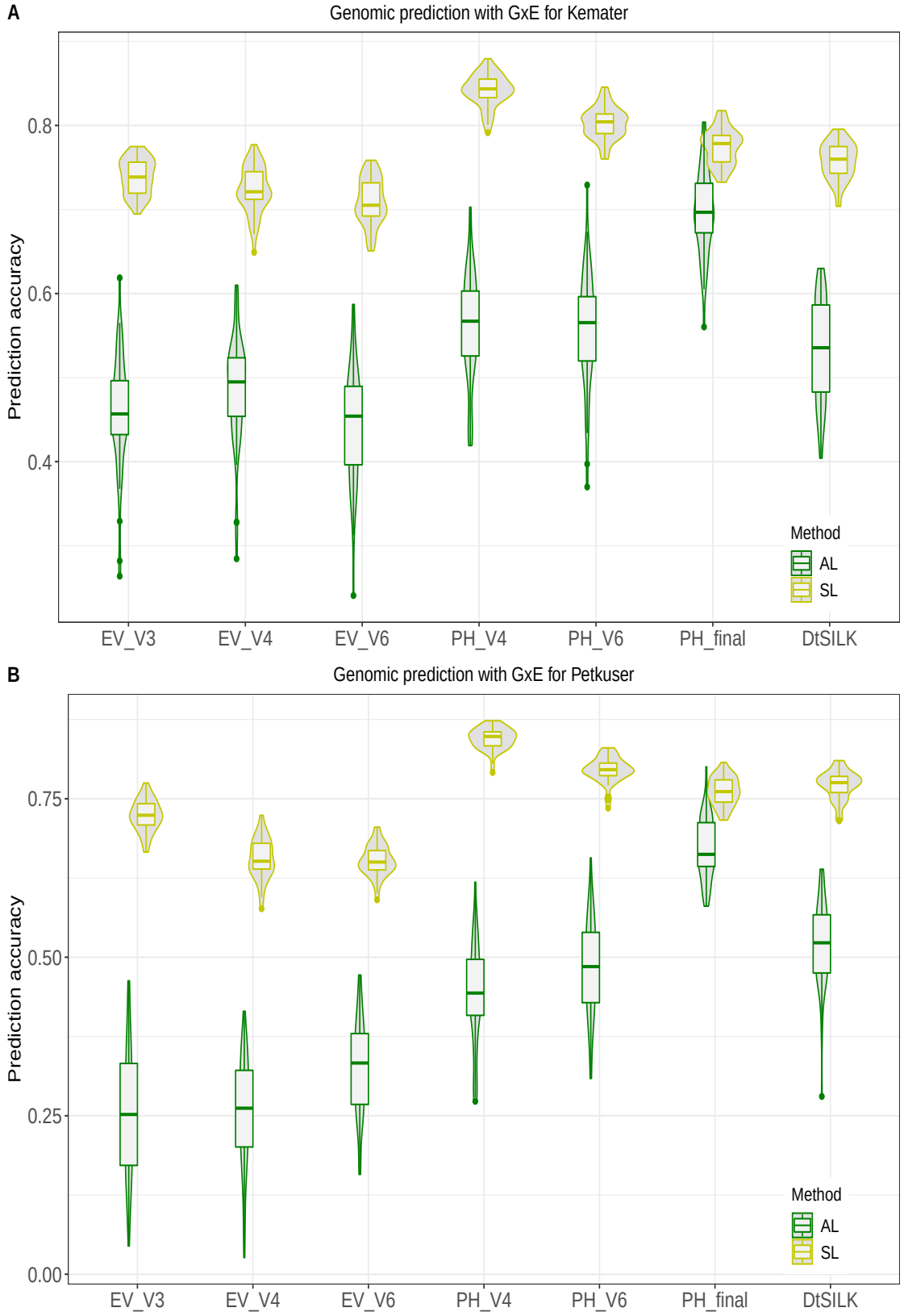


FIGURE 4.9: Results of genomic prediction for single environments for **A** Kemater and **B** Petkuser DH populations. With the prediction accuracies of the across location BLUEs (AL) and the single location BLUEs (SL)

Comparison of Bayesian methods in maize phenotype prediction

Figure 4.10 compares the results of phenotype prediction for five different Bayesian methods in terms of the respective prediction accuracy. The trials have been run for both DH populations independently. The results back those from the literature, mentioned in chapter 4.3.3.

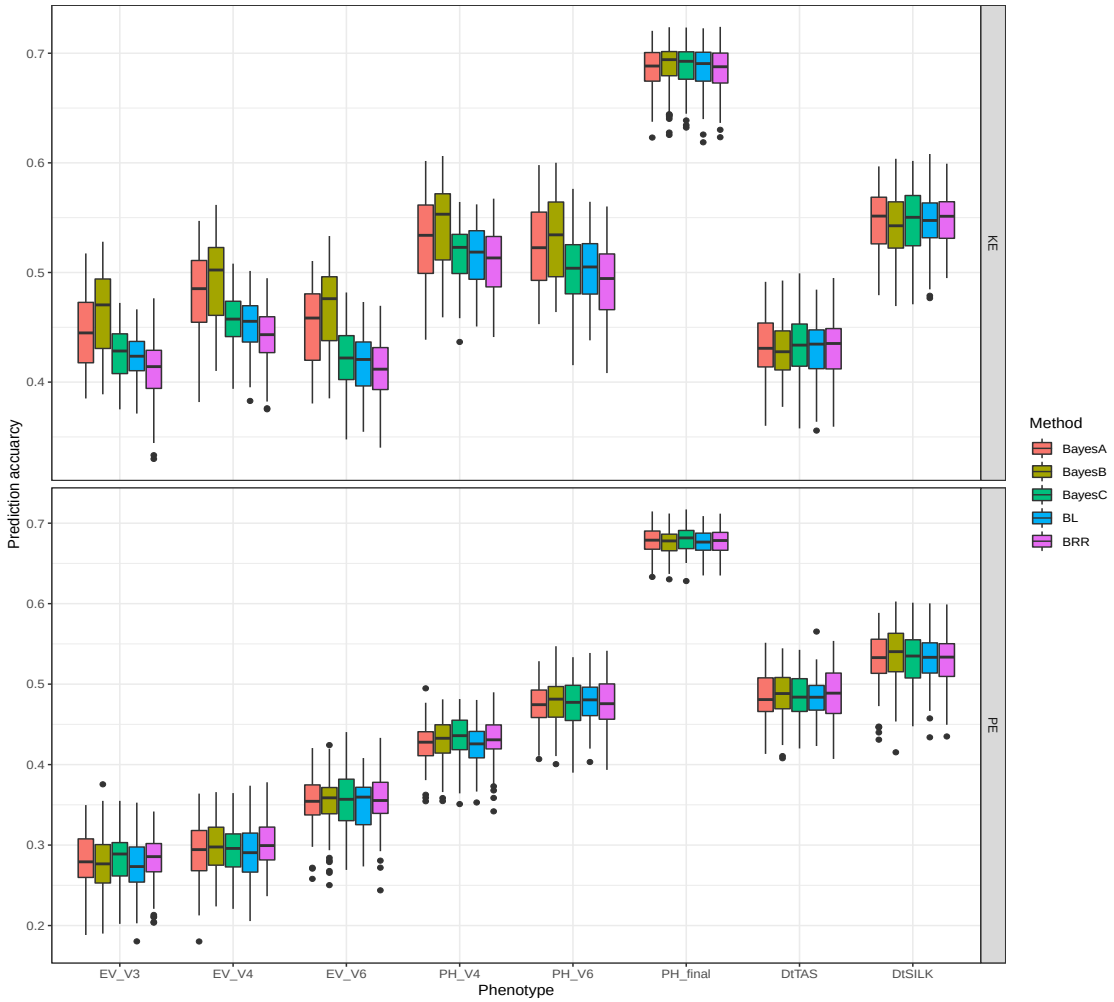


FIGURE 4.10: Results of genomic prediction of maize traits with five different Bayesian methods for eight difference traits for the DH populations Kemater (KE) and Petkuser (PE)

No single method is superior over all the others. This is more pronounced in the Petkuser subpopulation, where there is almost no difference between those methods at all and less articulated for Kemater. At first view the plot for Kemater

might suggest that BayesA and especially BayesB perform on higher levels for most trades. However, all the early vigor and plant height traits are closely correlated since they are basically the same trait measured at different time points, it is not surprising that the same algorithm that works well on one of those works well on the others as well.

Number of marker and prediction accuracy

Marker chips like the ones used to analyze the genomic maize data for this study contain hundreds of thousands of SNPs and other polymorphisms UNTERSEER et al., 2014. Due to LD, many of those markers do not segregate independently and are highly co-linear. In elite breeding materials LD is typical very large and cultivated maize is no exception. The markers used were already LD pruned to the remaining 28933 markers. Figure 4.11 shows the mean of prediction accuracies for the Bayesian methods as a function of the markers used. For that the complete marker set has been subsampled multiple times in 1k, 2k, 5k, 10k or 20k subsets and used in the prediction pipeline.

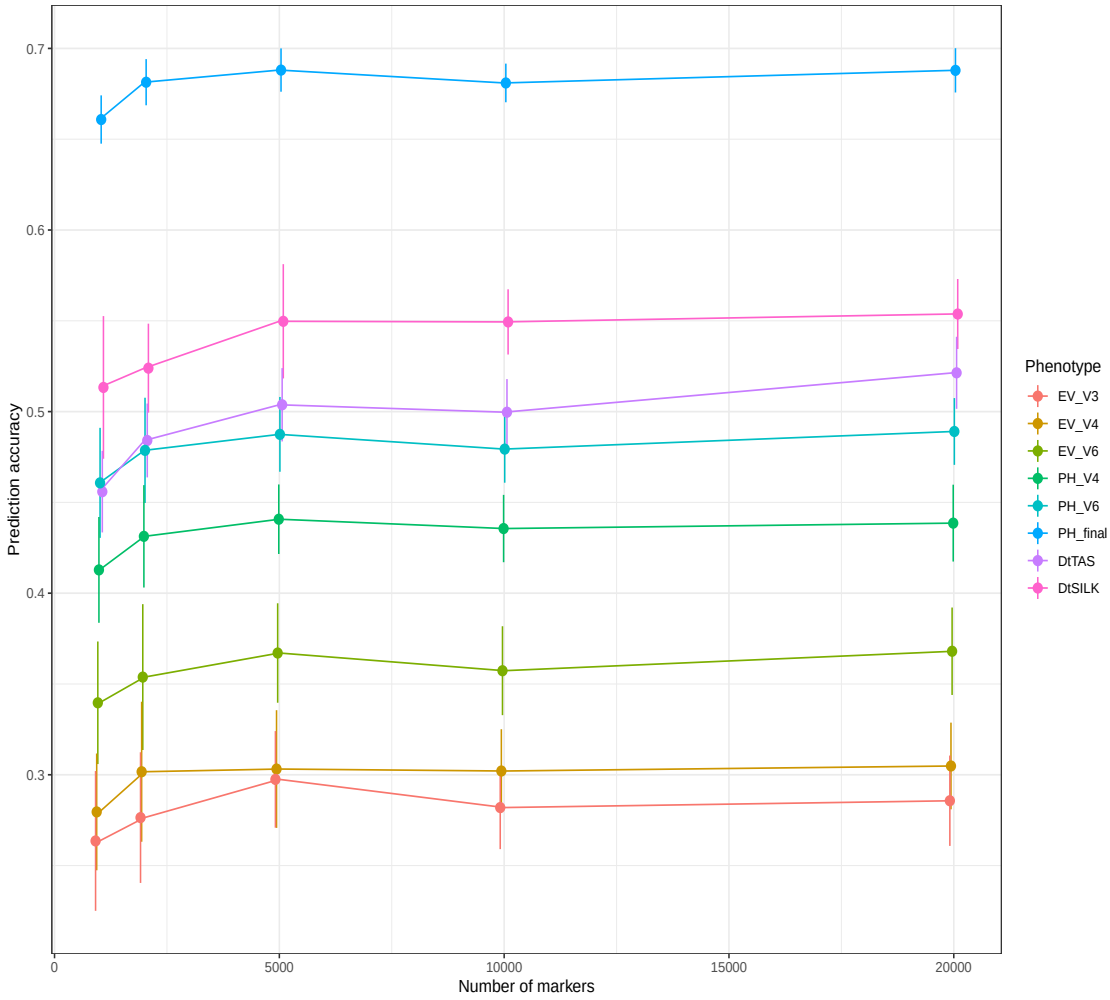


FIGURE 4.11: Predictive ability as a function of the number of markers for the Petkuser population of maize and varying amounts of markers and the Bayesian methods

As visible in figure 4.11 there is no increase in prediction accuracies after the marker sets exceed 5000 markers for all the maize traits in the Petkuser population. Furthermore, the majority of the predictive ability is already met with just 1000 genomic markers in the prediction set, so that an increase from 1k to 5k markers usually results in an increase smaller than 0.05. This holds true for all the traits and does not increase or decrease as the overall predictive ability grows larger.

Number of DHs and prediction accuracy

Figure 4.12 shows the prediction accuracy as a function of the number of Kemater DHs in the prediction set. As explained in section 4.6.3, the full 471 Kemater DH library was subsampled in 50, 100, 200, 300 and 400 DH subsets 10 times each and again each subset has been split into 80% TRN and 20% TRN 50 times.

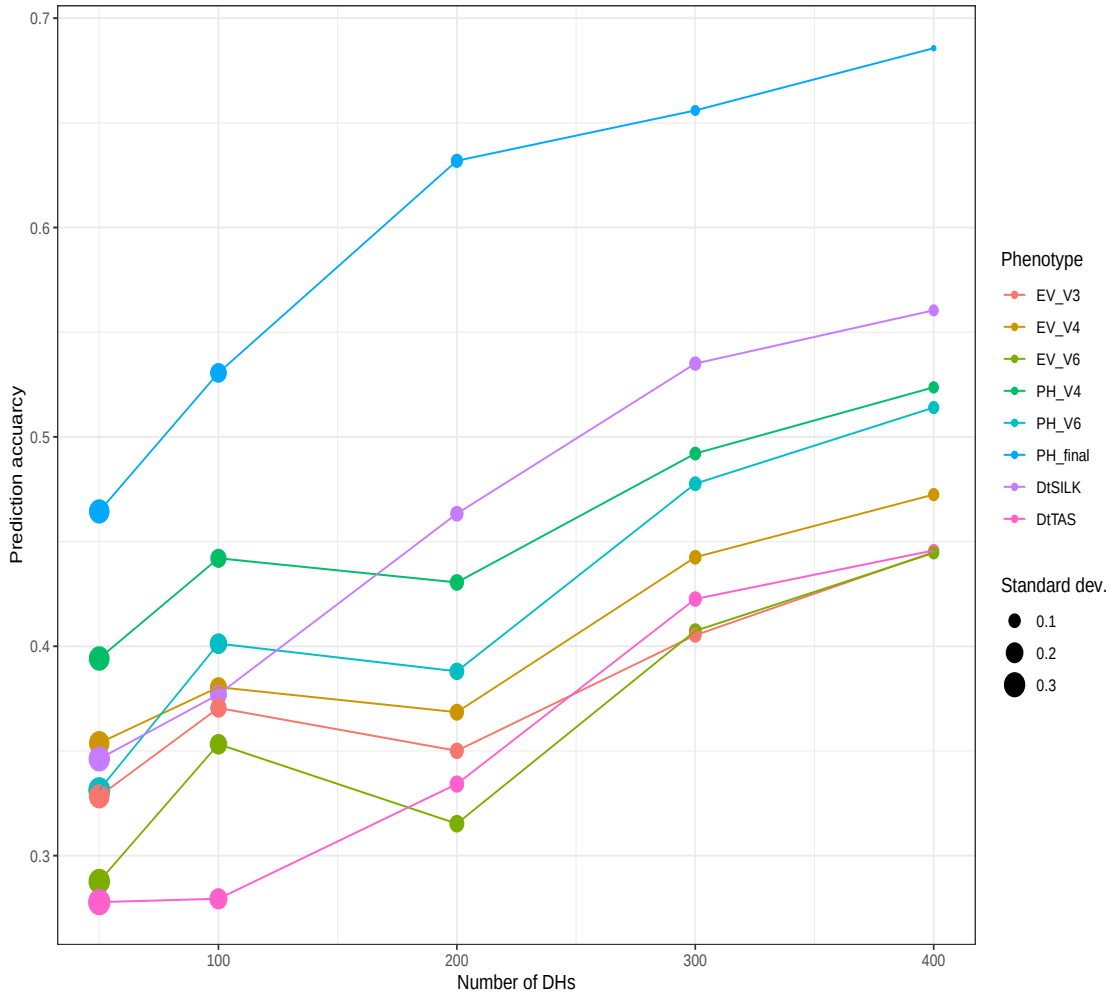


FIGURE 4.12: Predictive ability as a function of the number of DHs included in the prediction from the Kemater population. The dots represent the mean prediction accuracies for 10 randomly subsets with 50 fold validation each. The pointsize indicates the standard deviation

Figure 4.12 shows similar behaviors for all the eight different phenotypes assessed. With increasing number of DHs the prediction accuracy will gradually

increase, while the standard deviation of a total of 500 predictions for each subset and phenotype decreases. Figure 4.11, addressing $\rho_{(y,\hat{y})}$ as a function as the number of markers shows a plateau just after a couple of thousand markers. For the number of DHs a similar effect for $\rho_{(y,\hat{y})}$ is not observable. Even though the largest increases are realized between 50 and 200 DHs, between 200 and 400 DHs there appears to be a linear increase of the predictive ability, which does not cease yet.

4.8 Discussion

4.8.1 Correlation between heritability and prediction accuracy

The results in section 4.7.1 and table 4.5 show that for a large variety of different *A. thaliana* traits prediction accuracies vary from 0 to almost 0.9, depending on the trait being assessed. Next to the number of markers and phenotypes, the architectures of a trait as explained in section 4.2.1 will have an definite influence on the ability of prediction algorithms being able to produce meaningful results. Plot 4.13 compares the heritability with the results of GBLUP prediction for the 145 *A. thaliana* traits used for prediction. The heritability here is the pseudo-heritability as estimated during GWAS using REML estimations of the variance components of a trait with given genotypes.

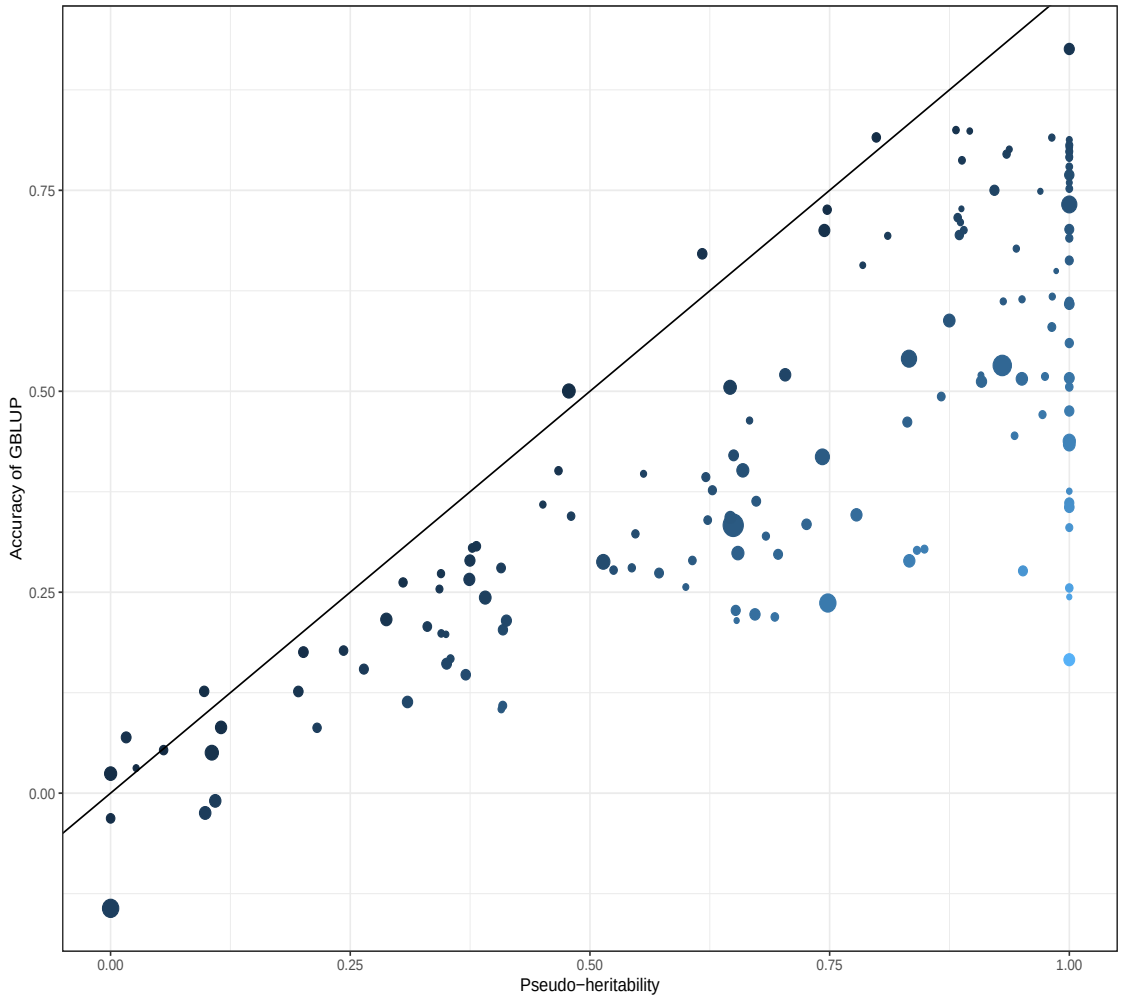


FIGURE 4.13: Prediction accuracies of GBLUP compared to the pseudo-heritability estimate of *A. thaliana* traits. The color scale indicates the difference between the accuracy and the pseudo-heritability. The size of the dots indicates the absolute difference between ANN and GBLUP prediction. Dots on the diagonal line have the same accuracy and heritability

The predictive ability and the pseudo-heritability estimate are highly correlated with each other and the latter can be used as a valid approximation for the accuracy of prediction. In chapter 4.2.1 it was stated that it is theoretically impossible for $\rho_{(y,\hat{y})}$ to exceed H^2. As figure 4.13 shows this assumption holds almost completely true. Only a few points are slightly above the diagonal and are either close to 0 for both $\rho_{(y,\hat{y})}$ and the pseudo-heritability or might be due to

over- or under inflation of the REML model. Many published studies found similar results concerning the comparison of the two metrics e.g KWONG et al., 2017; MORGANTE et al., 2018; YAP et al., 2018; PIASKOWSKI et al., 2018; ZHANG et al., 2019

4.8.2 Two or three layer networks outperform deeper ANNs

While different network architectures were tested, the ones with LCL were better performing than with just FCL architectures. Table 4.7 summarizes the success rates of different network architectures used for the prediction of the *A. thaliana* traits, whose results have been presented in section 4.7. The most successful architectures had only one or two hidden layer following the FLC. The most successful of those, being the best network for 56 of the traits, only has one layer with 150 nodes in the single, fully-connected hidden layer. Shallower networks being more successful in genomic prediction has been reported several times AZODI et al., 2019.

Since it is difficult to directly connect the genetic architecture of a trait to a certain type of network performing worth or better because the set of equations being solved is highly unbiased, the genetic architecture will have an influence on the success rates GIANOLA, 2013. If the number of layers increases in a network the number of trainable features increases much faster, than if the number of nodes per layer is increased with a constant number of layers. Therefore those networks, hypothetically, should be better suited to connect complex gene interaction networks, while shallower networks should capture all the additive signals.

TABLE 4.7: ANN architectures resulting in highest prediction accuracies, with the number of hidden layer (HL) and the total count (n)

LCL	Architecture	HL	n
True	150	2	56
True	50, 30	3	47
True	48	2	23
True	50, 35, 15	4	11
True	20, 10	3	5
True	100	2	2
True	150, 30	3	1

The constant problem remains, which is the dimensionality of the data. Even if epistatic interactions are present, their signal is mostly likely too weak within small populations and becomes obscured by the immense number of trainable features in deeper networks.

4.8.3 GxE interactions have great influence on plant development traits in maize

Genotype-environment interactions are present in all field trials, which complicates breeding and selection, as compared to trials with controlled environments. One advantage of machine learning, is that environmental information can easily be incorporated into ANN models because there are no assumptions prior to the analyses for effect sizes and the shrinkage imposed on the markers, as in the Bayesian models GIANOLA, 2013; BUSTOS-KORTS et al., 2016a. The maize traits under investigation show that σ_{GxE} has a great influence. Table 4.8 summarizes the results for the GxE predictions and shows that just by adding the environmental markers the prediction accuracies increase significantly for all traits.

TABLE 4.8: Comparison of prediction results of ANN with the single location (SL) and the across location (AL) BLUEs for Kemater and Petkuser

	Kemater			**Petkuser**		
Phenotype	SL	AL	Δ	SL	AL	Δ
EV_V3	0.73	0.46	0.27	0.72	0.25	0.47
EV_V4	0.72	0.49	0.23	0.66	0.25	0.40
EV_V6	0.70	0.44	0.26	0.65	0.33	0.33
PH_V4	0.84	0.56	0.28	0.84	0.44	0.41
PH_V6	0.80	0.56	0.25	0.80	0.48	0.31
PH_final	0.78	0.70	0.08	0.76	0.67	0.09
DtSILK	0.76	0.53	0.23	0.77	0.52	0.25

The largest gains were realized for traits with previously rather low predictive abilities, especially in the Petkuser population, suggesting that GxE has a greater influence then the actual narrow-sense heritability (h^2).

The extend of GxE, however, heavily depends on the environments that the traits were assessed in. The set of locations need to allow for the traits to segregate, otherwise a potentially great influence of environmental factors cannot be detected. There are also Bayesian approaches to include GxE in the prediction equations available CUEVAS et al., 2017; CROSSA et al., 2019, however, they are less straight forward to implement.

The future goal of research in GxE predictions, should be to predict the phenotypic of values traits of unknown genotypes in unknown environments. To achieve this environments, like genotypes need to be described by features or markers. In the case of the environments those are traits like soil constitution, whether and climate data etc. This field of enviromics or envirotyping is hitherto not very advanced, but certainly will receive more attention in the coming years due to its potential gains for breeding and medical applications alike RESENDE et al., 2019; CHANG and STOLER, 2019.

4.8.4 No algorithm outperforms the others

In the *A. thaliana* and maize traits, there is no single algorithm that is always able to outperform all the others. This holds true for many studies, that compared a variety of GP algorithms DE LOS CAMPOS et al., 2009; HESLOT et al., 2012; BLONDEL et al., 2015; RAMSTEIN et al., 2016; ROORKIWAL et al., 2016; AZODI et al., 2019. More influential the GP method on the predictive ability is the size of the training set, the number of markers and the overall heritability. As shown in previous chapters not only do algorithms do not have the tendency to outperform each other in general, even for individual traits there rarely is a significant difference between the performance of the methods. At first, this might seem surprising because the almost 200 trait-population combinations tested in the present studies, are unlikely to have the same genetic architecture, e.g. the distribution of marker effects, and they should vary in the magnitude of the additive and epistatic variance components of the total genetic variance. While the Bayesian methods and GBLUP, as previously thoroughly discussed, capture linear effects the ANNs should technically be able to asses non-linear effects in the prediction. While this has been shown in the proof of concept in section 4.4, it is not reflected in the real world examples, which points to the major issue in GP that is the size of the training set. Small training sets are unsuited to capture epistatic effects with low allele frequencies because the smaller the set is, the less likely it becomes that epistatic effects are distributed evenly in both the training and the testing population.
Secondly, even if a given trait biological is epistatic there might be a single marker, which is similar to an interaction pseudo-marker, whose effect size can be captured by the linear methods HILL, GODDARD, and VISSCHER, 2008; MONIR and ZHU, 2018. All this comes down to the problem of dimensionality due to the $n >> p$ problematic mentioned earlier in chapter 4.3.4. Other studies suggest that non-linear methods are superior when the number of markers n is smaller compared to the number of phenotypes p AZODI et al., 2019, which would allow

for epistatic effects to be more likely to appear in both TRN and TST and make it less likely that they are obscured behind co-linear additive markers.
A study that included more than 8000 wheat lines conducted by NORMAN et al., 2018 found similar results to those shown in figure 4.11. For four different wheat traits the increase of the predictive ability did not reach a plateau, even after more than 8000 genotypes were included in the training set, while the increase in gain due to adding markers to the analysis ceases after 5000 markers.

4.9 Conclusion

Artificial neural networks replaced older methods in many fields in a short amount of time and biological research is no exception ANGERMUELLER et al., 2016. Even though neural nets present a valuable addition to the toolbox of genomic selection in plant and animal breeding they do not perform to the potential they have shown in other fields. Among the reasons could be, next to the too few phenotypes usually present in the study to capture interactions, that neural networks are applied the exact same way as GBLUP or as the Bayesian methods are implemented, which does not allow to use the networks strength. Instead of using imputed data neural nets could be fed with raw input data and utilize multiple data sets at once including multiple traits and environments to increase prediction accuracies. Additionally to using the raw genotypic data, it could be promising to use the raw phenotypic data for genomic prediction. With the calculation of the BLUEs or means an important feature of the traits is lost, which is the variance of the observation between and among environments and repetitions. The networks for example could be trained an a multi-dimensional tensor containing all the measured values instead of the BLUEs.
Concluding, there is still a need to further assess the possibilities of machine learning and neural networks in quantitative genetics in general and especially genomic prediction.

5 General discussion and further observations

5.1 Genomic data preparation is error-prone

Researching and applying quantitative genetics from genome assemblies to genomic selection is tedious with many error-prone steps involved. To obtain optimal results every step in the entire process has to be optimized individually, without loosing the larger frame out of sight.

To perform analyses for quantitative genetics in general there are two types of data required: (i) genotypic and (ii) phenotypic data. Both are equally important and take many steps to procure.

Figure 5.1, reintroduced from chapter 1, schematically displays the key steps involved in obtaining genomic marker matrices for downstream analyses as GWAS and GS, from selection of candidate genotypes to the final numeric marker matrix. Genotyping can either be achieved by whole genome sequencing or by SNP analysis with a SNP array. The first step after sequencing, which provides raw reads, is to assemble the genome. As discussed in chapter 2, genome assembly is a complicated process. This holds true for both the assembly of core and plastid genomes. There is a large variety of tools available for core genome assemblies and like the ones for plastid genomes they vary in their algorithmic approaches and likewise their accuracy ZHANG et al., 2011, which makes it hard to determine whether polymorphisms between individual genomes are due to artifacts in the genome assembly pipeline or actually are mirrored in the biological

genome. Furthermore, genome assemblies result in one dimensional representations of formerly three dimensional genomes, losing most of the spatial and epigenetic information.

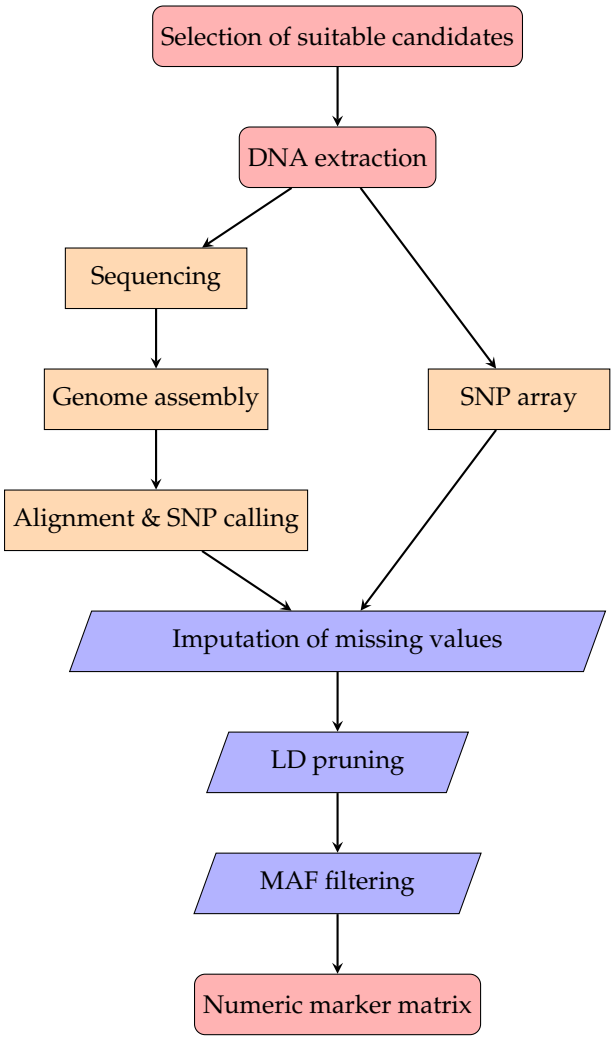


FIGURE 5.1: Schematic process of genotyping for quantitative genetics analyses with its crucial steps

After sequencing and assembling multiple genomes of a species the next step is to align them to detect genetic polymorphisms such as SNPs, InDels, etc. followed by the imputation of missing values. Imputation tools assume that all the missing data are actually missing due to the assembly and not actually missing in the genome as deletions. However, this step is necessary because GWAS and genomic selection requires complete data without missing values. Again, there is a

variety of tools for the imputation of missing markers. In plant genomics the most commonly used software is `Beagle` BROWNING and BROWNING, 2007; BROWNING, ZHOU, and BROWNING, 2018, which is based on hidden Markov models. As thoroughly reviewed by POOK et al., 2019 the accuracy of the algorithm varies vastly depending on the population, LD structure, chromosome region, effective population size and the allele frequency, all possibly leading to errors adding up the ones already introduced in the upper branches of the entire pipeline.

5.1.1 Imputation can lead to false positive GWAS results

Faulty imputation and SNP calling can result in false positive GWAS results as shown in the following example. Data from phenotypic trials with 330 fully sequenced *A. thaliana* accessions for carbon isotope discrimination were used to perform GWAS with a marker matrix containing 10 million SNPs imputed with `Beagle 3.0` DITTBERNER et al., 2018. This resulted in one marker with a significant p value on the fourth chromosome. Upon further investigation of the chromosomal region in question using the unimputed data, a complex haplotype structure was revealed as shown in figure 5.2.

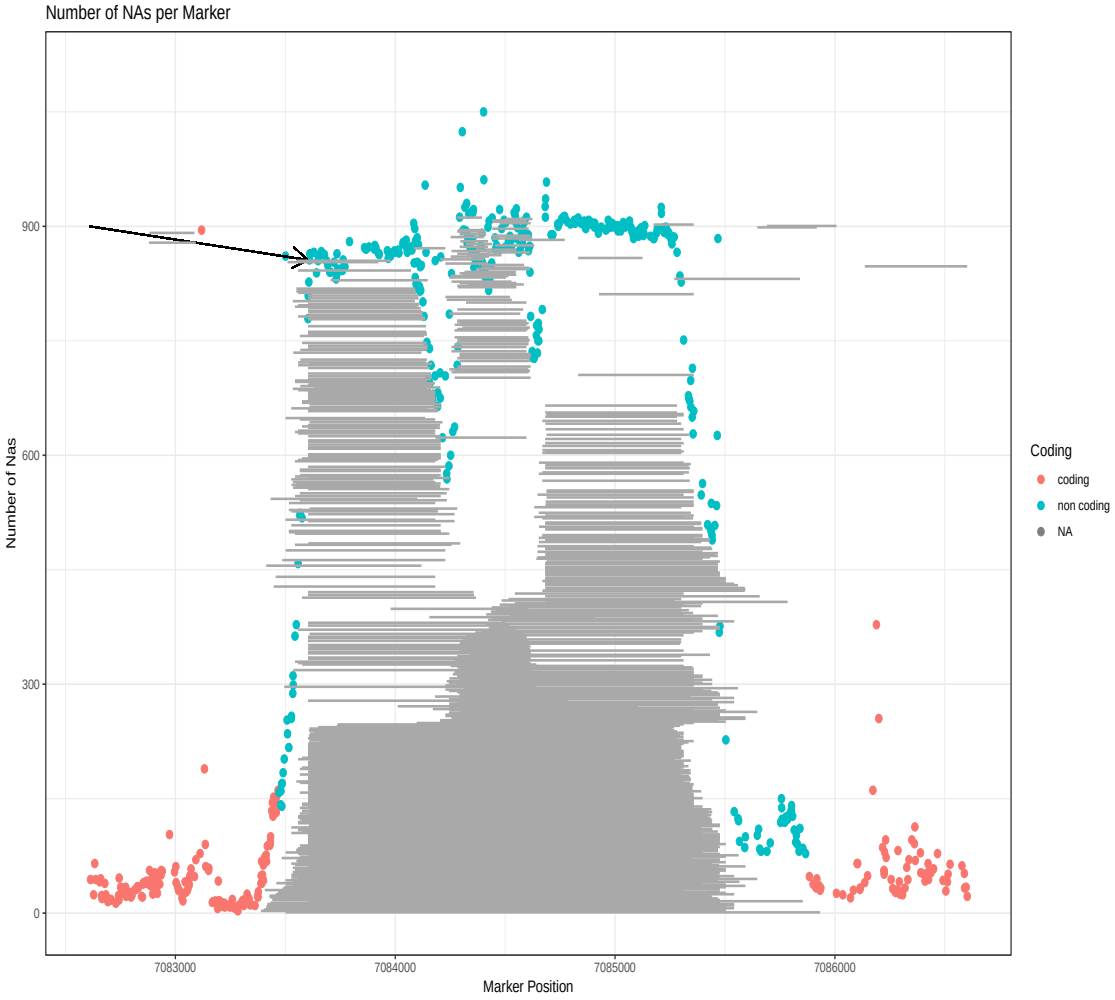


FIGURE 5.2: Haplotype structure on a 4 kbp window of chromosome five of *A. thaliana*. On the vertical axis the number of NAs in the population of 1135 accessions for a given marker is displayed. The horizontal axis gives the physical position on the chromosome. Red markers are located in coding and blue markers in non-coding regions according to the TAIR10 annotation RHEE et al., 2003. The gray bars indicate more than five coherent missing values for one accession. The arrow points to the location of the significant GWAS hit.

The significant SNP is located in a region where up to 80% of the data were originally missing values and were filled with Beagle 3.0. Additionally a complex structure of longer or shorter deletions is present, completely cutting out the non-coding region between the two coding ones in some accession. Taking a look at figure 5.2 it immediately becomes obvious that imputation in this region has to be

wrong because the complex haplotype structure is a clear indication for the missing values not being due to sequencing errors, but that they are actually mirrored in the biological genomes. The possibility of imputation leading to false positives has been discussed by LIN et al., 2010. The present case provides an practical example of the phenomenon.

Further in the scope of the study it was assessed weather the phasing algorithm used in Beagle 3.0 detected some signal from the haplotype structure that lead to the faulty imputation. The different haplotypes and deletions were coded as pseudo-markers for further association studies, all resulting in non-significant p-values. The plots in figure 5.2 provide a good example to show how the information loss about complex genomic structures can lead to false statistical assumptions.

5.1.2 Numeric marker matrices cannot represent the complexity of genomes

Figure 5.3 shows the complex haplotype structure of chromosome one of *A. thaliana*. The plots for chromosome two to five are included in appendix C.2. They basically all follow a similar pattern. The region directly flanking the centromere is more polymorphic than the telomeres at the p and q arms of the chromosomes, independent if the chromosome is metacentric like chromosome one and five, telocentric as chromosome two and four acrocentric as chromosome three. The centromere itself is highly conserved and generally coding regions have less haplotypes than non-coding regions. E.g. on chromosome one over a 1 kbp window in 1135 accessions there are ca. 78 different haplotypes in general and 98 in non-coding and 62 in coding regions on average. The most polymorphic regions, however, are often coding regions, like a region on the q arm of chromosome one located at around 22 Mbp, which has more than 700 segregating haplotypes in the 1 kbp window. The region harbors a locus containing disease resistance

genes CHENG et al., 2017, over the evolutionary advantages or disadvantages for those regions being highly polymorphic can only be speculated.

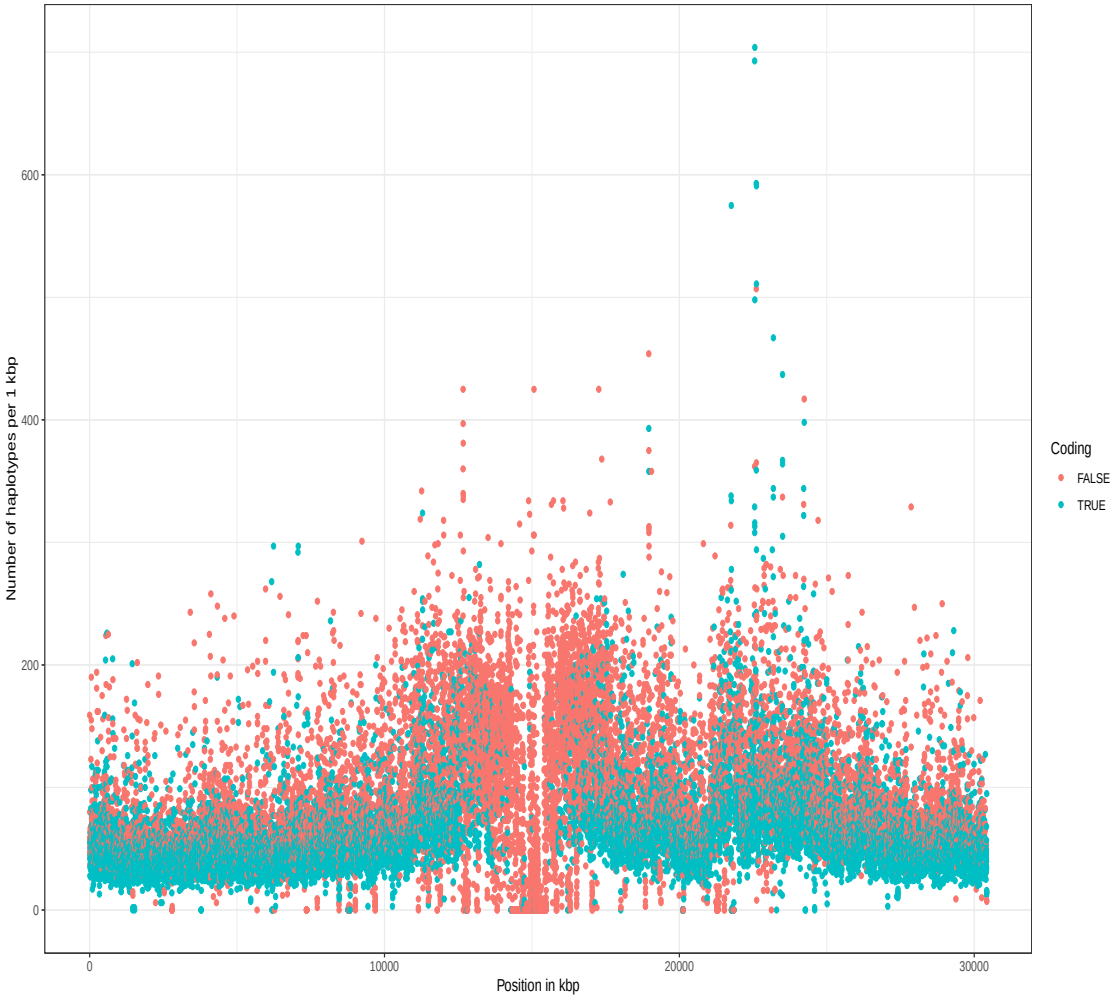


FIGURE 5.3: The number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kbp on chromosome one of *A. thaliana*.

Next to highly polymorphic regions there are regions, which are completely conserved and do not have a single polymorphism in a 1 kbp window. Around the centromere there are regions longer than 10 kbp with no SNPs. Intuitively one would assume that this would indicate important household genes that do not allow for any alterations in the amino acid sequence, however, the majority, around 75% of those regions are considered to be non-coding. Conserved

non-coding sequences (CNS) have been widely studied and shown great evolutionary importance and were witnessed across species with millions of years of evolutionary distance BURGESS and FREELING, 2014.

The haplotype analysis allows to visualize another interesting evolutionary artifact. Chromosome one of *A. thaliana* was derived from a fusion of two chromosomes of its next relative *A. lyrata*. Next to the active centromere located in the middle of the chromosome, at around 20 Mbp there is a region that shares some properties of a centromere, where one of the *A. lyrata* centromeres was located KOCH and MATSCHINGER, 2007.

The haplotype as well as the LD structure overall or for special regions e.g flowering time associated loci LI et al., 2014, is two complex to be represented sufficiently in a binary marker matrix. If his hold true, as shown, for *A. thaliana* it matters even more so in plants whose genomes are much larger, underwent multiple whole genome duplications and consist of many more chromosome. Like the diploid *Z. mays* with 10 chromosomes, the allotetraploid *B. napus* a product of hybridization with 19 chromosomes from the two ancestral species *B. rapa* and *B. oleracea* LIU, SNOWDON, and CHALHOUB, 2018, or the even more complex allohexaploid genome of *T. aestivum* CONSORTIUM, 2018.

Haplotypes are also an interesting prospect in genomic selection and GWAS. If the algorithms are calculated with haplotypes instead of marker matrices it could potentially be possible to reduce the number of features while keeping all information at the same time CALUS, DE ROOS, and VEERKAMP, 2008; CUYABANO, SU, and LUND, 2014; BEKELE et al., 2018. This could aid in reducing the dimensionality and the demand in computational resources alike.

5.1.3 Input data for GWAS and GS

Phenotypic trials are only able to represent a small subsample of whole populations. Even larger trials in the 1001 genome project only feature a bit more than

1000 accessions ATWELL et al., 2010; ALONSO-BLANCO et al., 2016. In practical trials it is common to randomly pick accessions, cultivars or genotypes in the hope that they will segregate for a certain trait. Sometimes this can be backed by a PCA or an analysis of molecular variance to choose suitable candidates HÖLKER et al., 2019 but this is not common practice. This results in allele frequencies in the subpopulations not following those of the global populations and phenotypic values not following normal distributions. For the 402 tested *A. thaliana* traits analyzed only 72 follow a normal distribution (own observation), according the Shapiro-Wilk test SHAPIRO and WILK, 1965. Taking into account that many statistical tests assume normal distributed data, this is an another source of errors in the genomic analysis pipelines, leading to over or under inflation of p-values. This effect can become very large for imbalanced and/or binary phenotypes like YEL (appendix B). One method to overcome this problematic is to use permutation-based thresholds for significance, which can account better for phenotypic distributions than Bonferroni thresholds (chapter 4). In the given example the permutation threshold is around 10^{-16} and the Bonferroni threshold is approximately 10^{-8}, potentially leading to a larger number of number of false positive markers.
Due the many sources of statistical inaccuracies that can be possibly introduced in the whole genome analysis pipeline all results have to be carefully evaluated, which often times is not done sufficiently. For each significant marker that has been detected the raw genomic information needs to assessed to validate the results.

5.2 Prospects in genomic selection and plant breeding and conclusion

Plant breeding, like in the last decades and centuries, will utilize the technology of its time. Many new tools including genome editing were recently added to the

tool box of breeding and despite regulatory issues of quickly found its way to revolutionize modern plant biotechnology ARAKI and ISHII, 2015. While in the future GWAS and its relatives or progeny will be used to further elucidate the nature of quantitative traits, genomic selection and genome editing will allow further improvement of the worlds major crop plants RODRÍGUEZ-LEAL et al., 2017, aiming to provide germplasms with the yield potential required by the demand of the growing world's population.

Following the current trends in bioinformatics for plant breeding this will lead to a further increase in the dimensionality of data, as genotyping costs will further decline and modern automated phenotyping techniques allow for larger trials, swallowing less resources.

Thus there is an still increasing demand for computational tools and novel algorithms that can handle vast amounts of data and extract information in reasonable time frames. Therefore, quantitative genetics and genomic selection for plant breeding will remain under active research and as this thesis succeeded many studies concerning similar topics, it precedes many that will follow.

6 Abstract

Quantitative genetics - from genome assemblies to neural network aided omics-based prediction of complex traits

Quantitative genetics is the study of continuously distributed traits and their genetic components. Recent developments in DNA sequencing technologies and computational systems allow researchers to conduct large scale *in silico* studies. However, going from raw DNA reads to genomic prediction of quantitative traits with the help of neural networks is a long and error-prone process. In the course of this thesis, many steps involved in this process will be assessed in depth. Chapter 2 will feature a study that compares the landscape of chloroplast genome assembly tools. Chapter 3 will present a software to perform genome-wide association studies using modern tools, which allow GWAS-Flow to outperform current state of the art software packages. Chapter 4 will give an in depth introduction to machine learning and the nature of quantitative traits and will combine those to genomic prediction with artificial neural networks and compares the results to those of algorithms based on linear mixed models. Finally, in Chapter 5 the results from the previous chapters are summarized and used to elucidate the complex nature of studies concerning quantitative genetics.

7 Zusammenfassung

Quantitative Genetik - von Genomassemblierungen bis zur Vorhersage von Phänotypischen Merkmalen mit Hilfe von Omics unterstüzten Neuronalen Netzwerken

Quantitative Genetik beschäftigt sich mit kontinuierlich verteilten Merkmalen und deren genetischer Komponenten. In den letzten Jahren gab es vielfältige Entwicklungen in der Computertechnik und der Genomik, insbesondere der DNA Sequenzierung, was Forschern erlaubt großflächig angelegte *in silico* Studien durchzuführen. Jedoch ist es ein komplexer Prozess von rohen Sequenzdaten bis zur genomischen Vorhersage mit Hilfe von neuronalen Netzwerken zu kommen. Im Rahmen der vorliegenden Studien werden viele Schritte, die an diesem Prozess beteiligt sind beleuchtet. Kapitel 2 wird einen Vergleich zwischen einer Vielzahl an Werkzeugen zur Assemblierung von Chloroplasten Genomen ziehen. Kapitel 3 stellt eine neu entwickelte Software zur genom-weiten Assoziationskartierung vor, die bisherigen Programmen überlegen ist. Kapitel 4 stellt maschinelles Lernen und die genetischen Komponenten von quantitativen Merkmalen vor und bringt diese im Kontext der genomischen Vorhersagen zusammen. Zum Schluss in Kapitel 5 werden die vorherigen Ergebnisse im Gesamtkontext der quantitativen Genetik beleuchtet.

A Source code

A.1 GWAS-Flow

A.1.1 gwas.py

```
import os
import sys
import time
import numpy as np
import pandas as pd
import main
import h5py

# set defaults
mac_min = 1
batch_size =  500000
out_file = "results.csv"
m = 'phenotype_value'
perm = 1
mac_min= 6

X_file = 'gwas_sample_data/AT_geno.hdf5'
Y_file = 'gwas_sample_data/phenotype.csv'
K_file = 'gwas_sample_data/kinship_ibs_binary_mac5.h5py'



for i in range (1,len(sys.argv),2):
    if sys.argv[i] == "-x" or sys.argv[i] == "--genotype":
        X_file = sys.argv[i+1]
    elif sys.argv[i] == "-y" or sys.argv[i] == "--phenotype":
```

```
27          Y_file = sys.argv[i+1]
28      elif sys.argv[i] == "-k" or sys.argv[i] == "--kinship":
29          K_file = sys.argv[i+1]
30      elif sys.argv[i] == "-m":
31          m = sys.argv[i+1]
32      elif sys.argv[i] == "-a" or sys.argv[i] == "--mac_min":
33          mac_min = int(sys.argv[i+1])
34      elif sys.argv[i] == "-bs" or sys.argv[i] == "--batch-size":
35          batch_size = int(sys.argv[i+1])
36      elif sys.argv[i] == "-p" or sys.argv[i] == "--perm":
37          perm  = int(sys.argv[i+1])
38      elif sys.argv[i] == "-o" or sys.argv[i] == "--out":
39          out_file = sys.argv[i+1]
40      elif sys.argv[i] == "-h" or sys.argv[i] == "--help":
41          print("-x , --genotype :file containing marker information
    in csv or hdf5 format of size")
42          print("-y , --phenotype: file container phenotype
    information in csv format"  )
43          print("-k , --kinship : file containing kinship matrix of
    size k X k in csv or hdf5 format")
44          print("-m : name of columnn containing the phenotype :
    default m = phenotype_value")
45          print("-a , --mac_min : integer specifying the minimum
    minor allele count necessary for a marker to be included.
    Default a = 1" )
46          print("-bs, --batch-size : integer specifying the number of
     markers processed at once. Default -bs 500000" )
47          print("-p , --perm : single integer specifying the number
    of permutations. Default 1 == no perm ")
48          print("-o , --out : name of output file. Default -o results
    .csv  ")
49          print("-h , --help : prints help and command line options")
50          quit()
51      else:
52          print('unknown option ' + str(sys.argv[i]))
53          quit()
54
55
```

```
56
57 print("parsed commandline args")
58
59 start = time.time()
60
61 X,K,Y_,markers = main.load_and_prepare_data(X_file,Y_file,K_file,m)
62
63
64 ## MAF filterin
65 markers_used , X , macs = main.mac_filter(mac_min,X,markers)
66
67 ## prepare
68 print("Begin performing GWAS on ", Y_file)
69
70 if perm == 1:
71     output = main.gwas(X,K,Y_,batch_size)
72     if( X_file.split(".")[-1] == 'csv'):
73         chr_pos = np.array(list(map(lambda x : x.split("- "),
    markers_used)))
74     else:
75         chr_reg = h5py.File(X_file,'r')['positions'].attrs['
    chr_regions']
76         mk_index= np.array(range(len(markers)),dtype=int)[macs >=
    mac_min]
77         chr_pos = np.array([list(map(lambda x: sum(x > chr_reg
    [:,1]) + 1, mk_index)), markers_used]).T
78         my_time = np.repeat((time.time()-start),len(chr_pos))
79     pd.DataFrame({
80         'chr' : chr_pos[:,0] ,
81         'pos' : chr_pos[:,1] ,
82         'pval': output[:,0] ,
83         'mac' : np.array(macs[macs >= mac_min],dtype=np.int) ,
84         'eff_size': output[:,1] ,
85         'SE' : output[:,2]}).to_csv(out_file,index=False)
86 elif perm > 1:
87     min_pval = []
88     perm_seeds = []
89     my_time = []
```

```
90      for i in range(perm):
91          start_perm = time.time()
92          print("Running permutation ", i+1, " of ",perm)
93          my_seed  = np.asscalar(np.random.randint(9999,size=1))
94          perm_seeds.append(my_seed)
95          np.random.seed(my_seed)
96          Y_perm = np.random.permutation(Y_)
97          output = main.gwas(X,K,Y_perm,batch_size)
98          min_pval.append(np.min(output[:,0]))
99          print("Elapsed time for permuatation",i+1 ," with p_min",
    min_pval[i]," is",": ", round(time.time() - start_perm,2))
100         my_time.append(time.time()-start_perm)
101     pd.DataFrame({
102         'time': my_time ,
103         'seed': perm_seeds ,
104         'min_p': min_pval }).to_csv(out_file,index=False)
105
106 print("done")
107
108 end = time.time()
109 eltime = np.round(end -start,2)
110
111 if eltime <= 59:
112     print("Total time elapsed",  eltime, "seconds")
113 elif eltime > 59 and eltime <= 3600:
114     print("Total time elapsed",  np.round(eltime / 60,2) , "minutes
    ")
115 elif eltime > 3600 :
116     print("Total time elapsed",  np.round(eltime / 60 / 60,2), "
    hours")
117
118
```

A.1.2 main.py

```
1     import pandas as pd
2     import numpy as np
3     from scipy.stats import f
4     import tensorflow as tf
```

```
    import limix
    import herit
    import h5py
    import limix
    import multiprocessing as mlt

    def load_and_prepare_data(X_file,Y_file,K_file,m):
    type_K = K_file.split(".")[-1]
    type_X = X_file.split(".")[-1]

    ## load and preprocess genotype matrix
    Y = pd.read_csv(Y_file,engine='python').sort_values(['
accession_id']).groupby('accession_id').mean()
    Y = pd.DataFrame({'accession_id' :  Y.index, 'phenotype_value'
: Y[m]})
    if type_X == 'hdf5' or type_X == 'h5py'  :
        SNP = h5py.File(X_file,'r')
        markers= np.asarray(SNP['positions'])
        acc_X =  np.asarray(SNP['accessions'][:],dtype=np.int)
    elif type_X == 'csv' :
        X = pd.read_csv(X_file,index_col=0)
        markers = X.columns.values
        acc_X = X.index
        X = np.asarray(X,dtype=np.float32)/2
    else :
        sys.exit("Only hdf5, h5py and csv files are supported")

    if type_K == 'hdf5' or type_K == 'h5py':
        k = h5py.File(K_file,'r')
        acc_K = np.asarray(k['accessions'][:],dtype=np.int)
    elif type_K == 'csv':
        k = pd.read_csv(K_file,index_col=0)
        acc_K = k.index
        k = np.array(k, dtype=np.float32)

    acc_Y =  np.asarray(Y[['accession_id']]).flatten()
    acc_isec = [isec for isec in acc_X if isec in acc_Y]

```

```
41      idx_acc = list(map(lambda x: x in acc_isec, acc_X))
42      idy_acc = list(map(lambda x: x in acc_isec, acc_Y))
43      idk_acc = list(map(lambda x: x in acc_isec, acc_K))
44
45      Y_ = np.asarray(Y.drop('accession_id',1),dtype=np.float32)[
    idy_acc,:]
46
47      if type_X == 'hdf5' or type_X == 'h5py' :
48          X = np.asarray(SNP['snps'][0:(len(SNP['snps'])+1),],dtype=
    np.float32)[:,idx_acc].T
49          X = X[np.argsort(acc_X[idx_acc]),:]
50          k1 = np.asarray(k['kinship'][:])[idk_acc,:]
51          K  = k1[:,idk_acc]
52          K = K[np.argsort(acc_X[idx_acc]),:]
53          K = K[:,np.argsort(acc_X[idx_acc])]
54      else:
55          X  = X[idx_acc,:]
56          k1 = k[idk_acc,:]
57          K  = k1[:,idk_acc]
58
59
60      print("data has been imported")
61      return X,K,Y_,markers
62
63
64  def mac_filter(mac_min, X, markers):
65      ac1 = np.sum(X,axis=0)
66      ac0 = X.shape[0] - ac1
67      macs = np.minimum(ac1,ac0)
68      markers_used  = markers[macs >= mac_min]
69      X = X[:,macs >= mac_min]
70      return markers_used, X, macs
71
72  def gwas(X,K,Y,batch_size):
73      n_marker = X.shape[1]
74      n = len(Y)
75      ## REML
```

```
76      K_stand = (n-1)/np.sum((np.identity(n) - np.ones((n,n))/n) * K)
        * K
77      vg, delta, ve  = herit.estimate(Y,"normal",K_stand,verbose =
        False)
78      print(" Pseudo-heritability is " , vg / (ve + vg + delta))
79      print(" Performing GWAS on ", n , " phenotypes and ", n_marker
        ,"markers")
80      ## Transform kinship-matrix, phenotypes and estimate intercpt
81      Xo = np.ones(K.shape[0]).flatten()
82      M = np.transpose(np.linalg.inv(np.linalg.cholesky(vg * K_stand
        + ve  * np.identity(n)))).astype(np.float32)
83      Y_t = np.sum(np.multiply(np.transpose(M),Y),axis=1).astype(np.
        float32)
84      int_t = np.sum(np.multiply(np.transpose(M),np.ones(n)),axis=1).
        astype(np.float32)
85      ## EMMAX Scan
86      RSS_env = (np.linalg.lstsq(np.reshape(int_t,(n,-1)) , np.
        reshape(Y_t,(n,-1)))[1]).astype(np.float32)
87      ## calculate betas and se of betas
88      def stderr(a,M,Y_t2d,int_t):
89            x = tf.stack((int_t,tf.squeeze(tf.matmul(M.T,tf.reshape(a
        ,(n,-1))))),axis=1)
90            coeff = tf.matmul(tf.matmul(tf.linalg.inv(tf.matmul(tf.
        transpose(x),x)),tf.transpose(x)),Y_t2d)
91            SSE = tf.reduce_sum(tf.math.square(tf.math.subtract(Y_t,tf
        .math.add(tf.math.multiply(x[:,1],coeff[0,0]),tf.math.multiply(x
        [:,1],coeff[1,0])))))
92            SE = tf.math.sqrt(SSE/(471-(1+2)))
93            StdERR = tf.sqrt(tf.linalg.diag_part(tf.math.multiply(SE ,
         tf.linalg.inv(tf.matmul(tf.transpose(x),x)))))[1]
94            return tf.stack((coeff[1,0],StdERR))
95      ## calculate residual sum squares
96      def rss(a,M,y,int_t):
97            x_t = tf.reduce_sum(tf.math.multiply(M.T,a),axis=1)
98            lm_res = tf.linalg.lstsq(tf.transpose(tf.stack((int_t,x_t)
        ,axis=0)),Y_t2d)
99            lm_x = tf.concat((tf.squeeze(lm_res),x_t),axis=0)
```

```
100         return tf.reduce_sum(tf.math.square(tf.math.subtract(tf.
    squeeze(Y_t2d),tf.math.add(tf.math.multiply(lm_x[1],lm_x[2:]),
    tf.multiply(lm_x[0],int_t)))))
101     ## loop over the batches
102     for i in range(int(np.ceil(n_marker/batch_size))):
103         tf.reset_default_graph()
104         if n_marker < batch_size:
105             X_sub = X
106         else:
107             lower_limit = batch_size * i
108             upper_limit = batch_size * i + batch_size
109             if upper_limit <= n_marker :
110                 X_sub = X[:,lower_limit:upper_limit]
111                 print("Working on markers ", lower_limit , " to ",
    upper_limit, " of ", n_marker )
112             else:
113                 X_sub = X[:,lower_limit:]
114                 print("Working on markers ", lower_limit , " to ",
    n_marker, " of ", n_marker )
115         config = tf.ConfigProto()
116         n_cores = mlt.cpu_count()
117         config.intra_op_parallelism_threads = n_cores
118         config.inter_op_parallelism_threads = n_cores
119         sess = tf.Session(config=config)
120         Y_t2d = tf.cast(tf.reshape(Y_t,(n,-1)),dtype=tf.float32)
121         y_tensor =  tf.convert_to_tensor(Y_t,dtype = tf.float32)
122         StdERR = tf.map_fn(lambda a : stderr(a,M,Y_t2d,int_t),
    X_sub.T)
123         R1_full = tf.map_fn(lambda a: rss(a,M,Y_t2d,int_t), X_sub.T
    )
124         F_1 = tf.divide(tf.subtract(RSS_env, R1_full),tf.divide(
    R1_full,(n-3)))
125         if i == 0 :
126             output = sess.run(tf.concat([tf.reshape(F_1,(X_sub.
    shape[1],-1)),StdERR],axis=1))
127         else :
128             tmp = sess.run(tf.concat([tf.reshape(F_1,(X_sub.shape
    [1],-1)),StdERR],axis=1))
```

```
129             output = np.append(output,tmp,axis=0)
130         sess.close()
131         F_dist = output[:,0]
132     pval  = 1 - f.cdf(F_dist,1,n-3)
133     output[:,0] = pval
134     return output
135
136
137
```

A.1.3 herit.py

```
1
2 def estimate(y, lik, K, M=None, verbose=True):
3     from numpy_sugar.linalg import economic_qs
4     from numpy import pi, var, diag
5     from glimix_core.glmm import GLMMExpFam
6     from glimix_core.lmm import LMM
7     from limix._data._assert import assert_likelihood
8     from limix._data import normalize_likelihood, conform_dataset
9     from limix.qtl._assert import assert_finite
10    from limix._display import session_block, session_line
11    lik = normalize_likelihood(lik)
12    lik_name = lik[0]
13    with session_block("Heritability analysis", disable=not verbose
    ):
14        with session_line("Normalising input...", disable=not
    verbose):
15            data = conform_dataset(y, M=M, K=K)
16        y = data["y"]
17        M = data["M"]
18        K = data["K"]
19        assert_finite(y, M, K)
20        if K is not None:
21           # K = K / diag(K).mean()
22            QS = economic_qs(K)
23        else:
24            QS = None
25        if lik_name == "normal":
```

```
            method = LMM(y.values, M.values, QS, restricted=True)
            method.fit(verbose=verbose)
        else:
            method = GLMMExpFam(y, lik, M.values, QS, n_int=500)
            method.fit(verbose=verbose, factr=1e6, pgtol=1e-3)
        g = method.scale * (1 - method.delta)
        e = method.scale * method.delta
        if lik_name == "bernoulli":
            e += pi * pi / 3
        v = var(method.mean())
        return g , v , e


```

A.2 Genomic prediction

A.2.1 GP ANN

```

import os,sys,gc
import pandas as pd
import numpy as np
import timeit
from datetime import datetime
import keras
import tensorflow as tf
from keras import backend as K
from keras import layers
from keras.models import Sequential
from keras.layers import Dense, Dropout, GaussianNoise,
    AlphaDropout, Reshape
from keras.layers import Flatten, LocallyConnected1D,
    LocallyConnected2D
from keras.optimizers import Adam, Adagrad, Adadelta
from keras.backend.tensorflow_backend import set_session

##set default values
```

```
18
19 learning_rate = 0.01
20 JobID = 1
21 ps = 25
22 optim = "adam"
23 X_file = "KE.geno.csv"
24 Y_file = "KE_pheno.csv"
25 CV_file = "KE_cv_pw.csv"
26 label = "DtSILK"
27 start_time = timeit.default_timer()
28 act="relu"
29 drop_rate = str('0.5,0.5,0.5')
30 arc = str('63,63')
31 DG = 'D,D,D,D,D,G'
32 LC = True
33 training_epochs = 25
34 hyp = False
35
36 ### parse command line arguments
37
38 for i in range (1,len(sys.argv),2):
39     if sys.argv[i] == "-x":
40         X_file = sys.argv[i+1]
41     elif sys.argv[i] == "-y":
42         Y_file = sys.argv[i+1]
43     elif sys.argv[i] == "-cv":
44         CV_file = sys.argv[i+1]
45     elif sys.argv[i] == "-JobID":
46         JobID = int(sys.argv[i+1])
47     elif sys.argv[i] == "-label":
48         label = sys.argv[i+1]
49     elif sys.argv[i] == "-act":
50         act = str(sys.argv[i+1])
51     elif sys.argv[i] == "-epochs":
52         training_epochs = int(sys.argv[i+1])
53     elif sys.argv[i] == "-lr":
54         learning_rate = float(sys.argv[i+1])
55     elif sys.argv[i] == "-arc":
```

```
56          arc = sys.argv[i+1]
57      elif sys.argv[i] == "-ps":
58          ps = int(sys.argv[i+1])
59      elif sys.argv[i] == "-dr":
60          drop_rate=str(sys.argv[i+1])
61      elif sys.argv[i] == "-LC":
62           LC = bool(sys.argv[i+1])
63      elif sys.argv[i] == "hyp":
64          hyp = bool(sys.argv[i+1])
65      else:
66          print('unknown option ' + str(sys.argv[i]))
67          quit()
68
69
70  x = pd.read_csv(X_file, index_col = 0)
71  y = pd.read_csv(Y_file, index_col = 0)
72  cv_folds = pd.read_csv(CV_file,index_col=0)
73
74  ## select column of phenotype file via columnname
75
76  y = y[[label]]
77  ## activity_regularizer=regularizers.l1(0.01)))
78
79  def build_network(arc,drop_rate,LC,DG):
80      def add_drops(model,drop_out,k):
81          if DG[k].upper() == 'D':
82              model.add(Dropout(drop_out[0]))
83          elif DG[k].upper() == 'G':
84              model.add(GaussianNoise(drop_out[k]))
85          elif DG[k].upper() == "A":
86              model.add(AlphaDropout(drop_out[k]))
87          else:
88              pass
89          return model
90      DG = DG.strip().split(",")
91      arc = arc.strip().split(",")
92      archit = []
93      for layer in  arc:
```

```
94          archit.append(int(layer))
95      layer_number = len(archit)
96      drop_rate = drop_rate.strip().split(",")
97      drop_out = []
98      for drops in drop_rate:
99          drop_out.append(float(drops))
100     model = Sequential()
101     if LC == True:
102         model.add(Reshape(input_shape=(x_train.shape[1],),
    target_shape=(x_train.shape[1],1)))
103         model.add(LocallyConnected1D(1,10,strides=7,input_shape=(
    x_train.shape[1],1)))
104         model.add(Flatten())
105         start = 0
106         model = add_drops(model,drop_out,start)
107     elif LC == False:
108         model.add(Dense(archit[0], kernel_initializer='
    truncated_normal', activation=act, input_shape=(x_train.shape
    [1],)))
109         model = add_drops(model,drop_out,start)
110     start = 1
111     for k in range(start,len(archit)):
112         model.add(Dense(archit[k], kernel_initializer='
    truncated_normal', activation=act))
113         model = add_drops(model,drop_out,k)
114     model.add(Dense(1, kernel_initializer='truncated_normal'))
115     return(model)
116
117 config = tf.ConfigProto()
118 #config.gpu_options.per_process_gpu_memory_fraction = 0.1
119 config.gpu_options.allow_growth = True
120 set_session(tf.Session(config=config))
121
122 if not os.path.isfile("RESULTScv50.txt"):
123     out2 = open("RESULTScv50.txt",'w')
124     out2.write('DateTime\tCompTime\tDF\tGenos\tPhenos\tCV_fold\
    tArchit\tConv\tActFun\tEpochs\tdrop_rate\tAccuracy\n' )
125
```

```
126 for k in range(1,51):
127     print("Training on cv fold "+ str(k))
128     cv = cv_folds['cv_' + str(k)]
129     num_cvs = np.ptp(cv) + 1
130 
131     i = 1
132     x_train = x[cv != i]
133     x_test = x[cv == i]
134     y_train = y[cv != i]
135     y_test = y[cv == i]
136 
137     yhat = np.zeros(shape = y_test.shape)
138 
139     model = build_network(arc,drop_rate,LC,DG)
140     model.compile(loss='mse', optimizer=Adam(lr=0.01,decay = 0.001)
    ,metrics=['accuracy'])
141     model.fit(x_train,y_train, epochs=training_epochs , verbose=0)
142 #     score = model.evaluate(x_test, y_test, verbose=0)
143     bla = model.predict(x_test)
144     y_sub= y[np.asarray(cv == i)]
145 
146     print(model.summary())
147     print('\n')
148     print(label)
149 
150     comp_time = int(round(timeit.default_timer() - start_time,0))
151 
152     DateTime = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
153     acc = np.corrcoef(bla[:,0],np.asarray(y_sub)[:,0])[0,1]
154 
155     out2 = open("RESULTScv50.txt", 'a')
156     out2.write('%s\t%i\t%s\t%s\t%s\t%i\t%s\t%s\t%s\t%i\t%s\t%0.5f\n
    ' % (
157         DateTime, comp_time, label, X_file, Y_file, int(k), arc, LC
    ,  act,int(training_epochs), drop_rate, round(acc,4)))
158 
159     del model,bla, x_train, x_test, y_train, y_test
160     K.clear_session()
```

```
    gc.collect()

    config = tf.ConfigProto()
    #config.gpu_options.per_process_gpu_memory_fraction = 0.1
    config.gpu_options.allow_growth = True
    set_session(tf.Session(config=config))
```

A.2.2 GBLUP

```
    geno_pred <- function(phenocsv,genocsv,cvfcsv,cvf=1,mod = "BRR",label,phe)
{
    my_phe <- phe
    depends<- c("BGLR","doBy","doParallel",'R.utils',"BBmisc","dplyr")
    foo <- sapply(depends,
                  function(X){if(!suppressPackageStartupMessages(require(X,character.only = T))){install.packages(X)}})
    foo <- sapply(depends,function(X){suppressPackageStartupMessages(library(X,character.only=TRUE))})
    rm(foo)

    maze <- read.csv(genocsv, row.names = 1)
    phe <- read.csv(phenocsv, row.names = 1)
    cvffolds <- read.csv(cvfcsv,row.names=1)

    X <- scale(maze)
    y <- phe[[label]]
    if(any(is.na(y))){
        rms <- which(is.na(y))
        y <- y[-rms]
        X <- X[-rms,]
    }
    for(i in 1:50){
        cvf = i
        n=length(y)
        seed <- sample(1:100,1)
                                        #set.seed(seed)
```

```
26                                          #folds=sample(1:cvf,size=n,
    replace=T)
27          folds = cvffolds[,cvf]
28          yHatCV=rep(NA,n)
29
30          for(i in 1:max(folds)){
31              cat("Predicting cv-fold ",i," of ", max(folds))
32              tst=which(folds==i)
33              yNA=y
34              yNA[tst]=NA
35              fm=BGLR(y=yNA,ETA=list(list(X=X,model=mod)),verbose =F
    ,nIter=7000,burnIn=1000)
36              yHatCV[tst]=fm$yHat[tst]
37              cat("   done\n")
38          }
39
40          my_cor <- cor(yHatCV,y,use = "complete.obs")
41          print(c("Corrleation of GP", mod, my_cor))
42          filename = paste0(my_phe,"_gp_results.csv")
43          print(filename)
44          if(!any(dir() == filename)){
45              res <- matrix(ncol=8, nrow = 1) %>%
46                  setColNames(c("geno","pheno","cv_folds","seed","
    label", "cor","method","nmark"))
47              res[1,] <- c(as.character(genocsv),as.character(
    phenocsv),as.character(cvf),as.character(seed),
48                           as.character(label),as.character(my_cor),
    as.character(mod),dim(X)[2])
49              print("#################")
50         print(res)
51         print("############")
52              write.csv(res,filename)
53          }else{
54              res <- read.csv(filename,row.names = 1)
55              for(i in 1:7){
56                  res[,i] <- as.character(res[,i])
57              }
```

```
58              res[dim(res)[1]+1,] <- c(as.character(genocsv),phenocsv
    ,cvf,seed,as.character(label),my_cor,mod,dim(X)[2])
59              write.csv(res,filename)
60          }
61      }
62
63    }
64 ## execute this script with: Rscript ex.gblup.r -x genofile -y
    phenofile -c cv file
65 source("~/PHD/Projects/gblup/bglr.r")
66
67
68 my.args <- commandArgs(trailingOnly = TRUE)
69 #my.args <- c("-x", "gent_geno.csv","-y" , "gent_pheno.csv")
70 ### set defaults
71 #cvf.name = NA
72
73 ## parsing the command line options
74 all.opts <- c("-x","-y","-label","-h","-cv","-phe")
75 for(i in 1:length(my.args)){
76     if( i %% 2 == 1){
77         if(!my.args[i] %in%  all.opts){
78             cat("unknown option", my.args[i], "Use only", all.opts
    , "\n")
79             cat("use -h for help \n")
80             quit()
81         }
82     }
83     if(my.args[i] == "-x"){
84         geno.name <- as.character(my.args[i+1])
85     } else if(my.args[i] == "-y") {
86         pheno.name <- as.character(my.args[i+1])
87     } else if(my.args[i] == "-label") {
88         my_ph <- as.character(my.args[i+1])
89     } else if(my.args[i] == "-cv"){
90         cv.name = as.character(my.args[i+1])
91     } else if(my.args[i] == "-phe"){
92         my_phe <- as.character(my.args[i+1])
```

```
93      } else if(my.args[i] == "-h") {
94          print(" This script takes as a minimum two intputs\n")
95          print(" -x genotypefile")
96          print(" -y phenotypefile ")
97          print(" -cv cross-valiadtion file : is optional if none is
        specified random 5 fold cv will be used")
98          print(" -JobID : specify column number to use in your cross
         validation file")
99          print(" -label : use header of phenotype file column you
        want to use")
100         quit()
101     }
102 }
103
104
105 #pheno.name <- my.args[1]
106 #geno.name <- my.args[2]
107 #cvf.name <- my.args[3]
108
109 geno_pred(phenocsv = pheno.name,genocsv=geno.name, cvfcsv = cv.name
        , label=my_ph,mod = "BRR", phe =my_phe)
```

B *A. thaliana* phenotypic data

ID	Phenotype name	doi	Reference
1	FT Diameter Field	10.21958/phenotype:1	ATWELL et al., 2010
2	At2 CFU2	10.21958/phenotype:2	ATWELL et al., 2010
3	Leaf serr 16	10.21958/phenotype:3	ATWELL et al., 2010
4	Seed bank 133-91	10.21958/phenotype:4	ATWELL et al., 2010
5	Na23	10.21958/phenotype:5	ATWELL et al., 2010
6	Leaf serr 10	10.21958/phenotype:6	ATWELL et al., 2010
7	Emco5	10.21958/phenotype:7	ATWELL et al., 2010
8	Leaf roll 16	10.21958/phenotype:8	ATWELL et al., 2010
9	Leaf roll 10	10.21958/phenotype:9	ATWELL et al., 2010
10	Bs	10.21958/phenotype:10	ATWELL et al., 2010
11	2W	10.21958/phenotype:11	ATWELL et al., 2010
12	Rosette Erect 22	10.21958/phenotype:12	ATWELL et al., 2010
13	Cd114	10.21958/phenotype:13	ATWELL et al., 2010
14	Width 16	10.21958/phenotype:14	ATWELL et al., 2010
15	Storage 28 days	10.21958/phenotype:15	ATWELL et al., 2010
16	LY	10.21958/phenotype:16	ATWELL et al., 2010
17	avrRpm1	10.21958/phenotype:17	ATWELL et al., 2010
18	Width 10	10.21958/phenotype:18	ATWELL et al., 2010
19	Chlorosis 22	10.21958/phenotype:19	ATWELL et al., 2010
20	Storage 7 days	10.21958/phenotype:20	ATWELL et al., 2010
21	As2 CFU2	10.21958/phenotype:21	ATWELL et al., 2010
22	Co59	10.21958/phenotype:22	ATWELL et al., 2010

23	FW	10.21958/phenotype:23	ATWELL et al., 2010
24	Cu65	10.21958/phenotype:24	ATWELL et al., 2010
25	Bacterial titer	10.21958/phenotype:25	ATWELL et al., 2010
26	Width 22	10.21958/phenotype:26	ATWELL et al., 2010
27	Storage 56 days	10.21958/phenotype:27	ATWELL et al., 2010
28	YEL	10.21958/phenotype:28	ATWELL et al., 2010
29	FLC	10.21958/phenotype:29	ATWELL et al., 2010
30	FT16	10.21958/phenotype:30	ATWELL et al., 2010
31	FT10	10.21958/phenotype:31	ATWELL et al., 2010
32	FT Duration GH	10.21958/phenotype:32	ATWELL et al., 2010
33	Se82	10.21958/phenotype:33	ATWELL et al., 2010
34	LDV	10.21958/phenotype:34	ATWELL et al., 2010
35	Noco2	10.21958/phenotype:35	ATWELL et al., 2010
36	8W GH LN	10.21958/phenotype:36	ATWELL et al., 2010
37	0W	10.21958/phenotype:37	ATWELL et al., 2010
38	MT GH	10.21958/phenotype:38	ATWELL et al., 2010
39	After Vern Growth	10.21958/phenotype:39	ATWELL et al., 2010
40	Aphid number	10.21958/phenotype:40	ATWELL et al., 2010
41	LN22	10.21958/phenotype:41	ATWELL et al., 2010
42	Bs CFU2	10.21958/phenotype:42	ATWELL et al., 2010
43	avrRpt2	10.21958/phenotype:43	ATWELL et al., 2010
44	Hypocotyl length	10.21958/phenotype:44	ATWELL et al., 2010
45	Germ 22	10.21958/phenotype:45	ATWELL et al., 2010
46	Leaf roll 22	10.21958/phenotype:46	ATWELL et al., 2010
47	SD	10.21958/phenotype:47	ATWELL et al., 2010
48	8W	10.21958/phenotype:48	ATWELL et al., 2010
49	FT GH	10.21958/phenotype:49	ATWELL et al., 2010
50	DSDS50	10.21958/phenotype:50	ATWELL et al., 2010

51	Ca43	10.21958/phenotype:51	ATWELL et al., 2010
52	LC Duration GH	10.21958/phenotype:52	ATWELL et al., 2010
53	0W GH FT	10.21958/phenotype:53	ATWELL et al., 2010
54	B11	10.21958/phenotype:54	ATWELL et al., 2010
55	Chlorosis 10	10.21958/phenotype:55	ATWELL et al., 2010
56	RP GH	10.21958/phenotype:56	ATWELL et al., 2010
57	Chlorosis 16	10.21958/phenotype:57	ATWELL et al., 2010
58	LFS GH	10.21958/phenotype:58	ATWELL et al., 2010
59	Germ 10	10.21958/phenotype:59	ATWELL et al., 2010
60	Germ 16	10.21958/phenotype:60	ATWELL et al., 2010
61	Anthocyanin 16	10.21958/phenotype:61	ATWELL et al., 2010
62	Anthocyanin 10	10.21958/phenotype:62	ATWELL et al., 2010
63	At1 CFU2	10.21958/phenotype:63	ATWELL et al., 2010
64	Ni60	10.21958/phenotype:64	ATWELL et al., 2010
65	P31	10.21958/phenotype:65	ATWELL et al., 2010
66	Emwa1	10.21958/phenotype:66	ATWELL et al., 2010
67	As75	10.21958/phenotype:67	ATWELL et al., 2010
68	Germ in dark	10.21958/phenotype:68	ATWELL et al., 2010
69	FRI	10.21958/phenotype:69	ATWELL et al., 2010
70	As CFU2	10.21958/phenotype:70	ATWELL et al., 2010
71	Trichome avg C	10.21958/phenotype:71	ATWELL et al., 2010
72	Vern Growth	10.21958/phenotype:72	ATWELL et al., 2010
73	Mo98	10.21958/phenotype:73	ATWELL et al., 2010
74	Hiks1	10.21958/phenotype:74	ATWELL et al., 2010
75	Anthocyanin 22	10.21958/phenotype:75	ATWELL et al., 2010
76	Zn66	10.21958/phenotype:76	ATWELL et al., 2010
77	Trichome avg JA	10.21958/phenotype:77	ATWELL et al., 2010
78	LES	10.21958/phenotype:78	ATWELL et al., 2010

79	Silique 16	10.21958/phenotype:79	ATWELL et al., 2010
80	Emoy*	10.21958/phenotype:80	ATWELL et al., 2010
81	K39	10.21958/phenotype:81	ATWELL et al., 2010
82	0W GH LN	10.21958/phenotype:82	ATWELL et al., 2010
83	At2	10.21958/phenotype:83	ATWELL et al., 2010
84	At1	10.21958/phenotype:84	ATWELL et al., 2010
85	LN10	10.21958/phenotype:85	ATWELL et al., 2010
86	FT Field	10.21958/phenotype:86	ATWELL et al., 2010
87	LN16	10.21958/phenotype:87	ATWELL et al., 2010
88	avrB	10.21958/phenotype:88	ATWELL et al., 2010
89	LD	10.21958/phenotype:89	ATWELL et al., 2010
90	Seedling Growth	10.21958/phenotype:90	ATWELL et al., 2010
91	S34	10.21958/phenotype:91	ATWELL et al., 2010
92	Leaf serr 22	10.21958/phenotype:92	ATWELL et al., 2010
93	DW	10.21958/phenotype:93	ATWELL et al., 2010
94	Seed Dormancy	10.21958/phenotype:94	ATWELL et al., 2010
95	Mn55	10.21958/phenotype:95	ATWELL et al., 2010
96	Silique 22	10.21958/phenotype:96	ATWELL et al., 2010
97	avrPphB	10.21958/phenotype:97	ATWELL et al., 2010
98	Fe56	10.21958/phenotype:98	ATWELL et al., 2010
99	8W GH FT	10.21958/phenotype:99	ATWELL et al., 2010
100	4W	10.21958/phenotype:100	ATWELL et al., 2010
101	Li7	10.21958/phenotype:101	ATWELL et al., 2010
102	FT22	10.21958/phenotype:102	ATWELL et al., 2010
103	As2	10.21958/phenotype:103	ATWELL et al., 2010
104	SDV	10.21958/phenotype:104	ATWELL et al., 2010
105	Mg25	10.21958/phenotype:105	ATWELL et al., 2010
106	Secondary Dormancy	10.21958/phenotype:106	ATWELL et al., 2010

107	As	10.21958/phenotype:107	ATWELL et al., 2010
108	Area Sweden 2009 (1st experiment)	10.21958/phenotype:108	LI et al., 2010
109	Size Planting Summer 2009	10.21958/phenotype:109	LI et al., 2010
110	Size Sweden 2009 (2nd experiment)	10.21958/phenotype:110	LI et al., 2010
111	Size Planting Summer Loc Sweden 2009	10.21958/phenotype:111	LI et al., 2010
112	Area Sweden 2009 (2nd experiment)	10.21958/phenotype:112	LI et al., 2010
113	DTF Sweden 2008 (1st experiment)	10.21958/phenotype:113	LI et al., 2010
114	Yield Sweden 2009 (2nd experiment)	10.21958/phenotype:114	LI et al., 2010
115	Size Loc Sweden 2009	10.21958/phenotype:115	LI et al., 2010
116	DTF planting Summer Loc Sweden 2009	10.21958/phenotype:116	LI et al., 2010
117	DTF loc Sweden 2008	10.21958/phenotype:117	LI et al., 2010
118	DTF loc Sweden 2009	10.21958/phenotype:118	LI et al., 2010
119	DTF Spain 2009 (1st experiment)	10.21958/phenotype:119	LI et al., 2010
120	DTF planting Loc 2008	10.21958/phenotype:120	LI et al., 2010
121	DTF Spain 2009 (2nd experiment)	10.21958/phenotype:121	LI et al., 2010
122	Yield Spain 2009 (2nd experiment)	10.21958/phenotype:122	LI et al., 2010

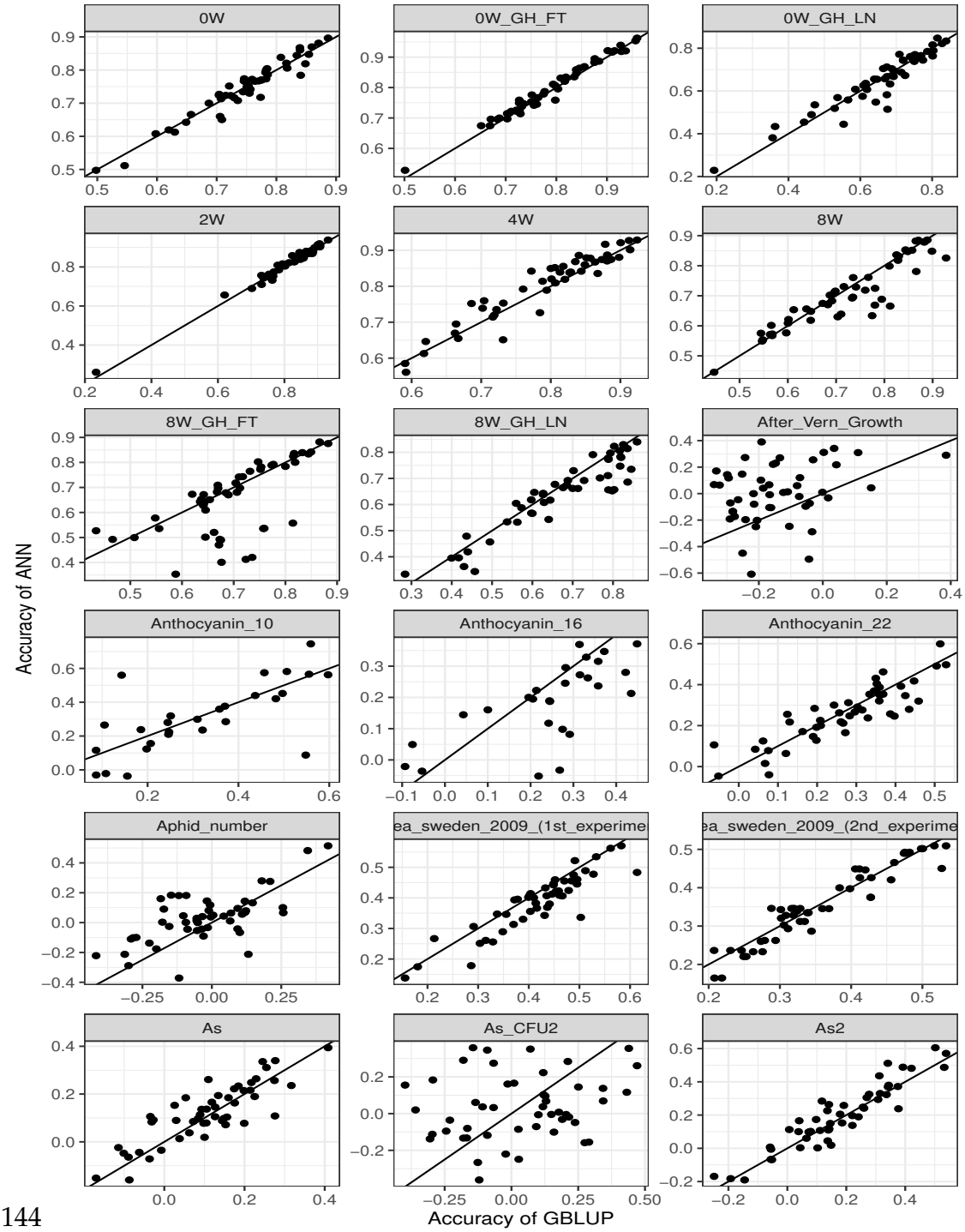
123	Size Sweden 2009 (1st experiment)	10.21958/phenotype:123	LI et al., 2010
124	Yield Spain 2009 (1st experiment)	10.21958/phenotype:124	LI et al., 2010
125	DTF main Effect 2009	10.21958/phenotype:125	LI et al., 2010
126	DTF main Effect 2008	10.21958/phenotype:126	LI et al., 2010
127	Size Spain 2009 (2nd experiment)	10.21958/phenotype:127	LI et al., 2010
128	Size Spain 2009 (1st experiment)	10.21958/phenotype:128	LI et al., 2010
129	DTF planting Summer 2009	10.21958/phenotype:129	LI et al., 2010
130	DTF planting Summer 2008	10.21958/phenotype:130	LI et al., 2010
131	Size Main Effect 2009	10.21958/phenotype:131	LI et al., 2010
132	DTF Spain 2008 (1st experiment)	10.21958/phenotype:132	LI et al., 2010
133	Yield Planting Summer 2009	10.21958/phenotype:133	LI et al., 2010
134	DTF Sweden 2009 (1st experiment)	10.21958/phenotype:134	LI et al., 2010
135	Yield Loc Sweden 2009	10.21958/phenotype:135	LI et al., 2010
136	DTF Spain 2008 (2nd experiment)	10.21958/phenotype:136	LI et al., 2010
137	Yield Main Effect 2009	10.21958/phenotype:137	LI et al., 2010
138	Yield Planting Summer Loc Sweden 009	10.21958/phenotype:138	LI et al., 2010

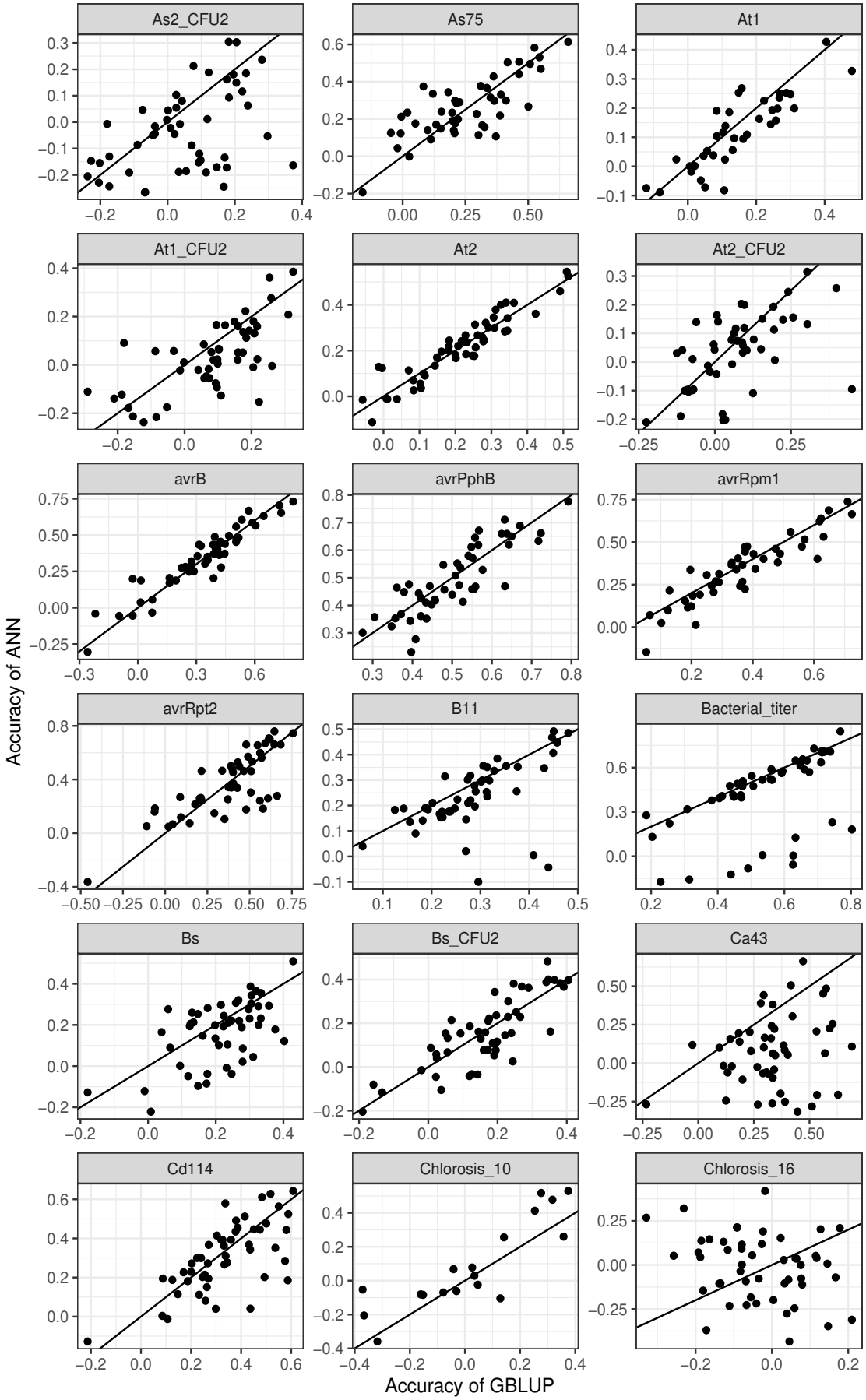
139	Yield Sweden 2009 (1st experiment)	10.21958/phenotype:139	LI et al., 2010
140	DTF Sweden 2009 (2nd experiment)	10.21958/phenotype:140	LI et al., 2010
141	DTF Sweden 2008 (2nd experiment)	10.21958/phenotype:141	LI et al., 2010
142	Mature cell length	10.21958/phenotype:142	MEIJÓ et al., 2014
143	Meristem zone length	10.21958/phenotype:143	MEIJÓ et al., 2014
144	M216T665	10.21958/phenotype:144	STRAUCH et al., 2015
145	M130T666	10.21958/phenotype:145	STRAUCH et al., 2015
146	M172T666	10.21958/phenotype:146	STRAUCH et al., 2015
261	FT10	10.21958/phenotype:261	ALONSO-BLANCO et al., 2016
262	FT16	10.21958/phenotype:262	ALONSO-BLANCO et al., 2016
269	Li7	10.21958/phenotype:269	FORSBERG et al., 2015
270	B11	10.21958/phenotype:270	FORSBERG et al., 2015
271	Na23	10.21958/phenotype:271	FORSBERG et al., 2015
272	Mg25	10.21958/phenotype:272	FORSBERG et al., 2015
273	P31	10.21958/phenotype:273	FORSBERG et al., 2015
274	S34	10.21958/phenotype:274	FORSBERG et al., 2015
275	K39	10.21958/phenotype:275	FORSBERG et al., 2015
276	Ca43	10.21958/phenotype:276	FORSBERG et al., 2015
277	Mn55	10.21958/phenotype:277	FORSBERG et al., 2015
279	Co59	10.21958/phenotype:279	FORSBERG et al., 2015
280	Ni60	10.21958/phenotype:280	FORSBERG et al., 2015
281	Cu65	10.21958/phenotype:281	FORSBERG et al., 2015
282	Zn66	10.21958/phenotype:282	FORSBERG et al., 2015

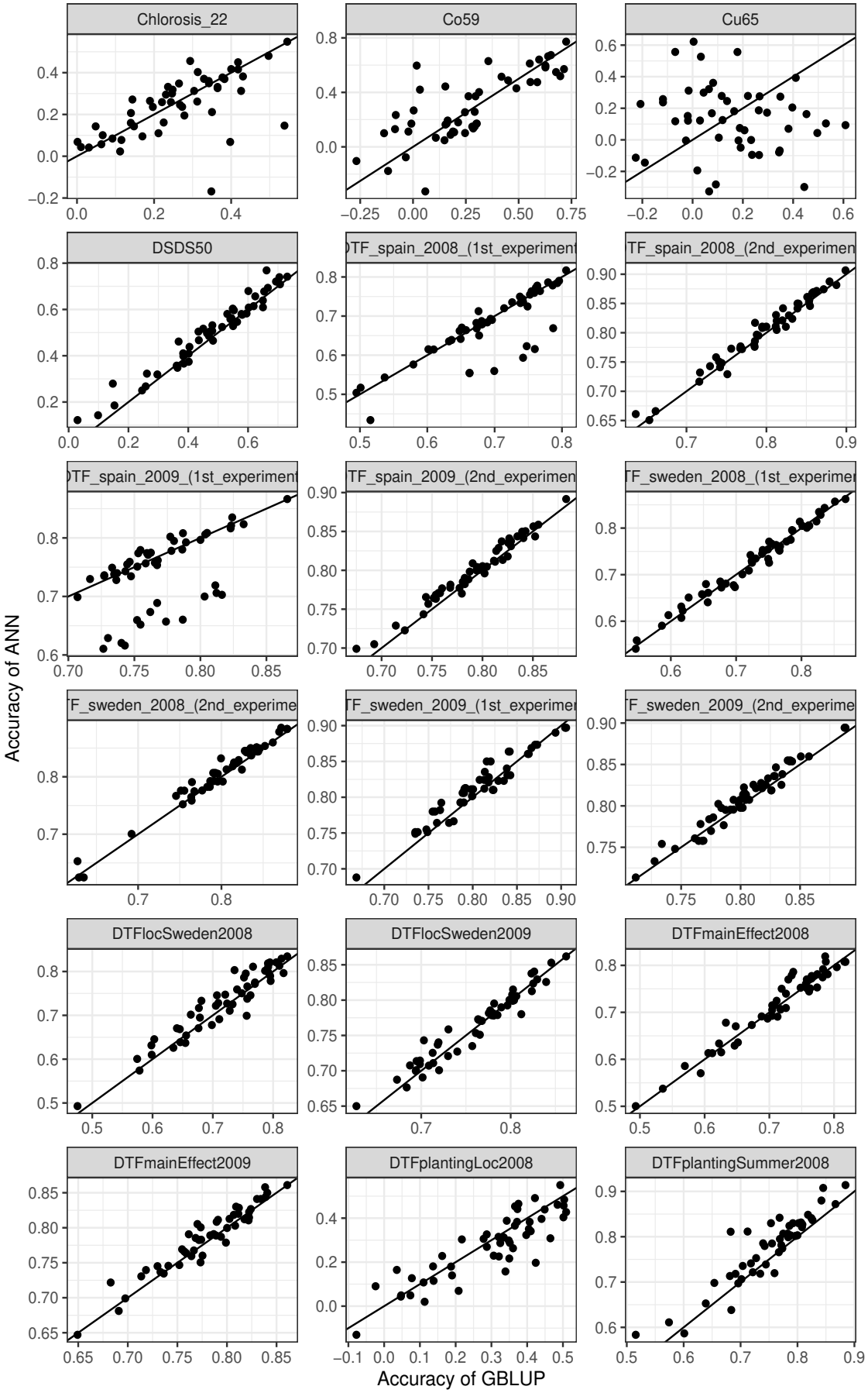
283	As75	10.21958/phenotype:283	FORSBERG et al., 2015
284	Se82	10.21958/phenotype:284	FORSBERG et al., 2015

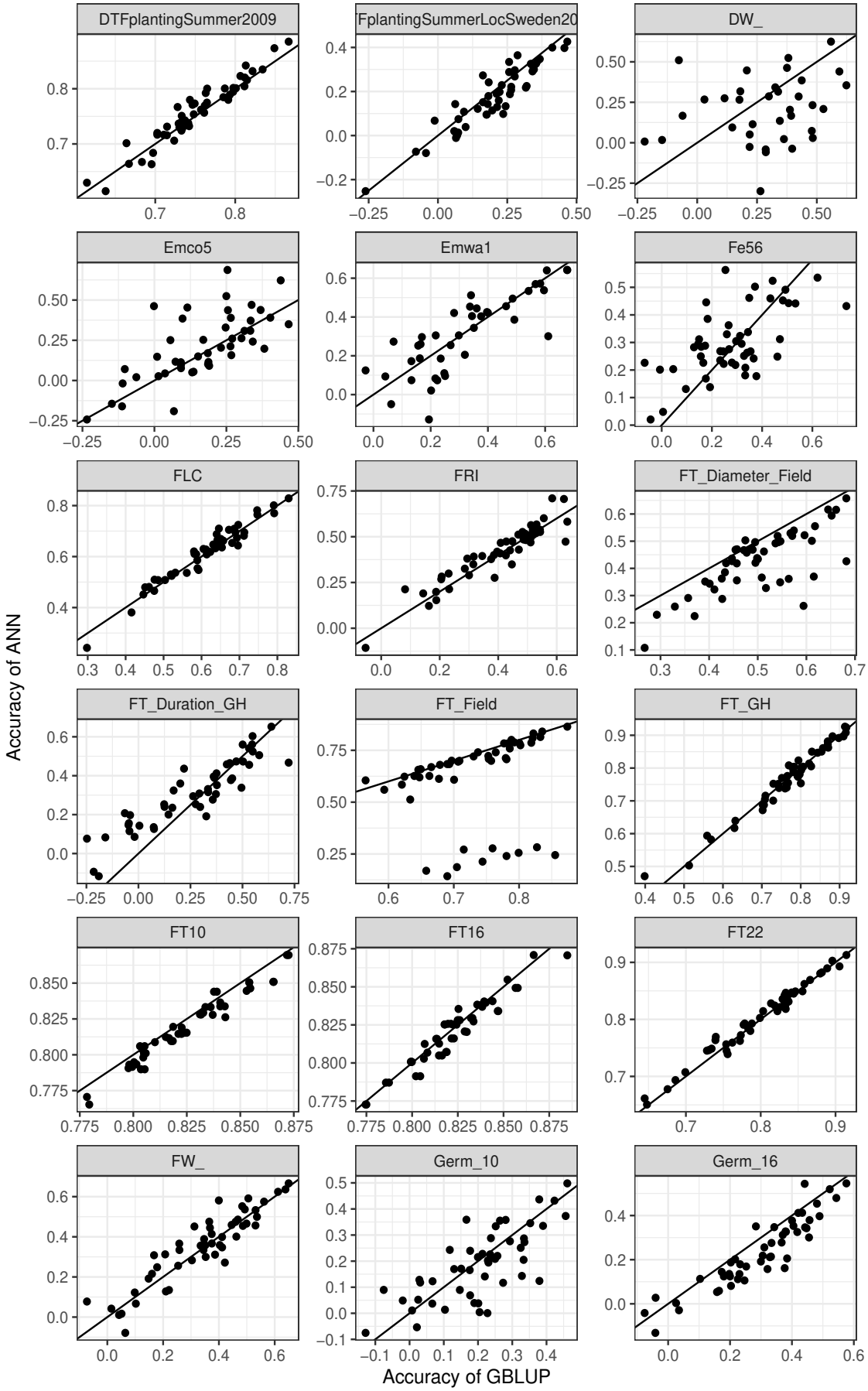
C Supplementary results

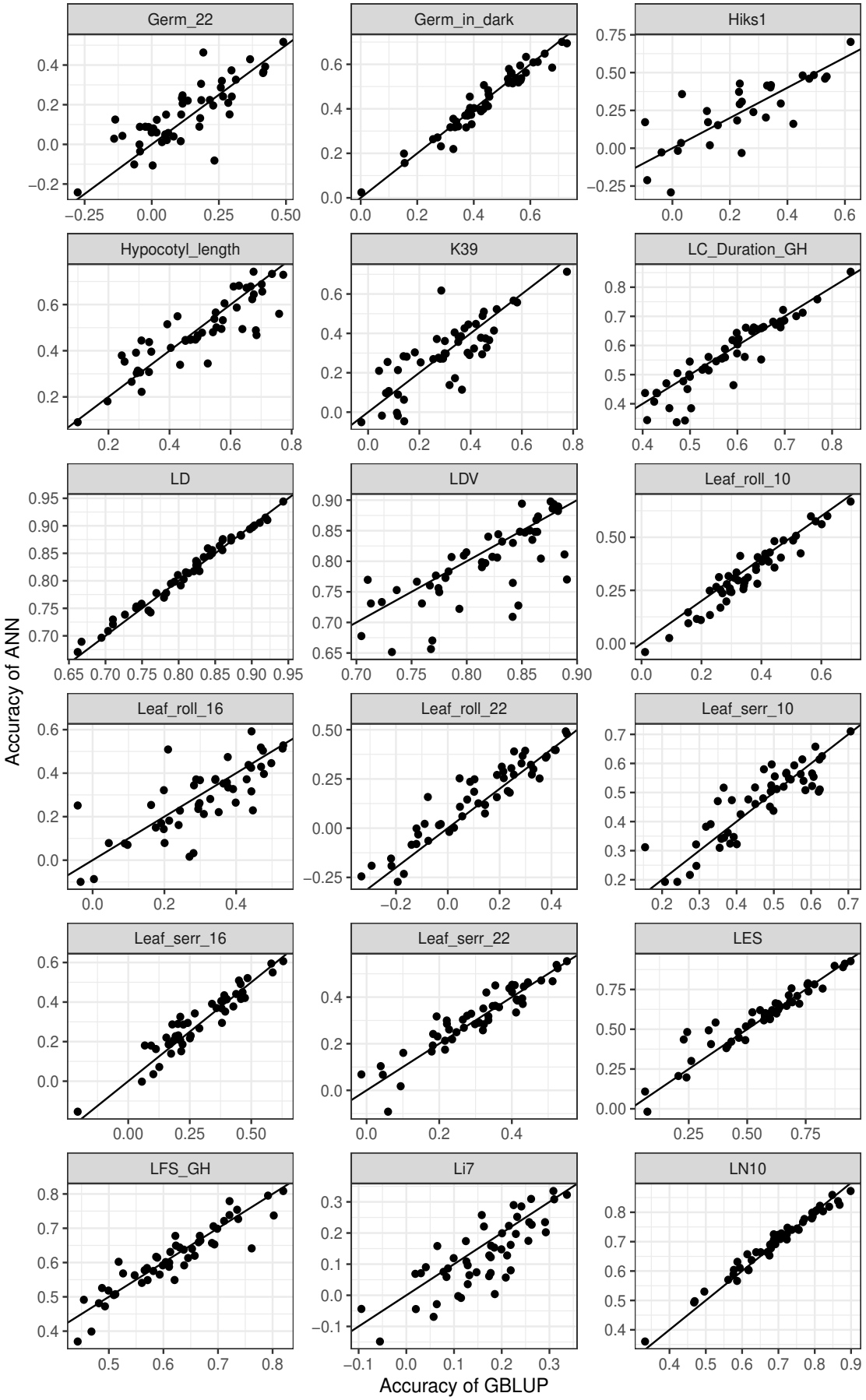
C.1 Correlation plots of *A. thaliana* GP

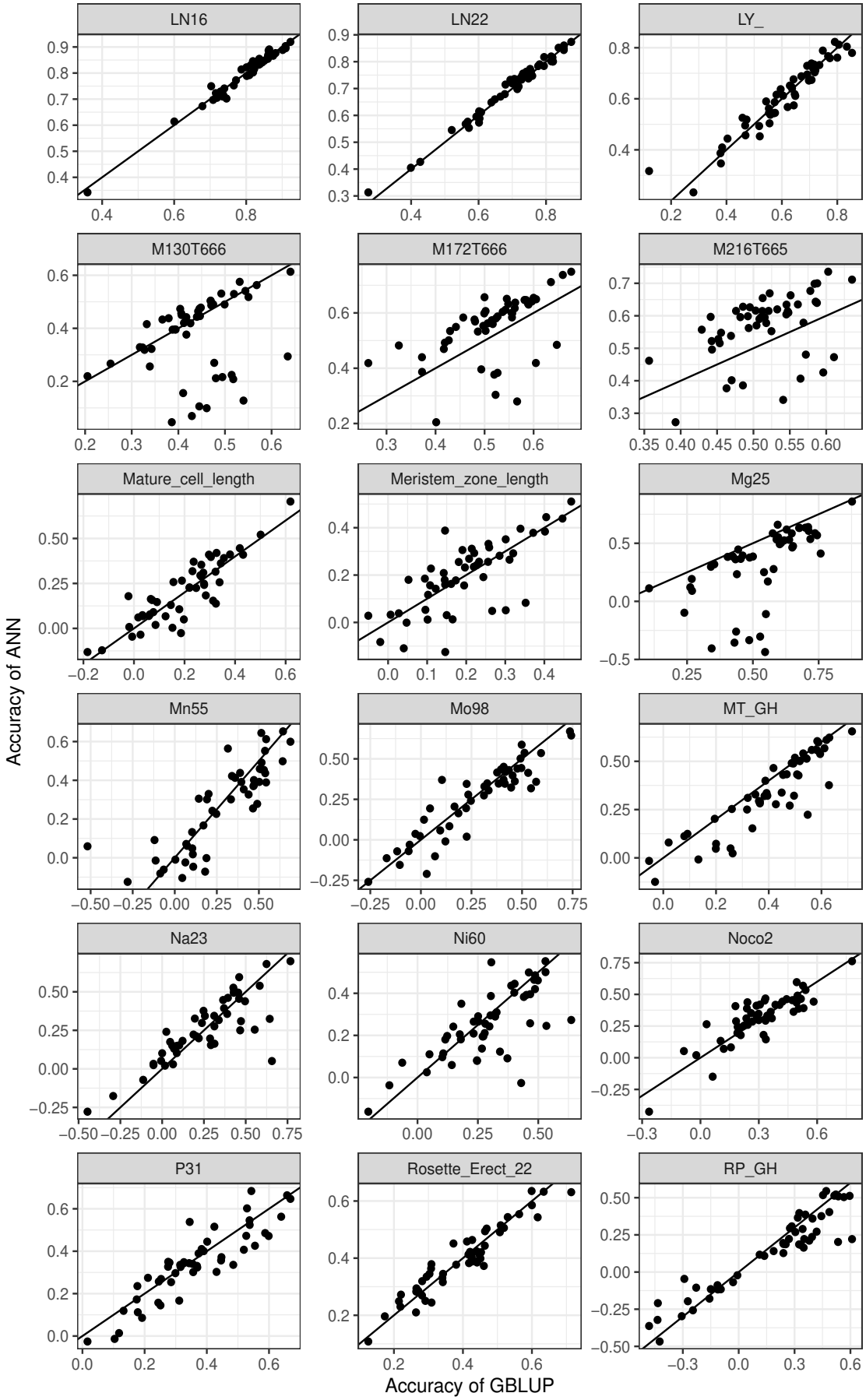


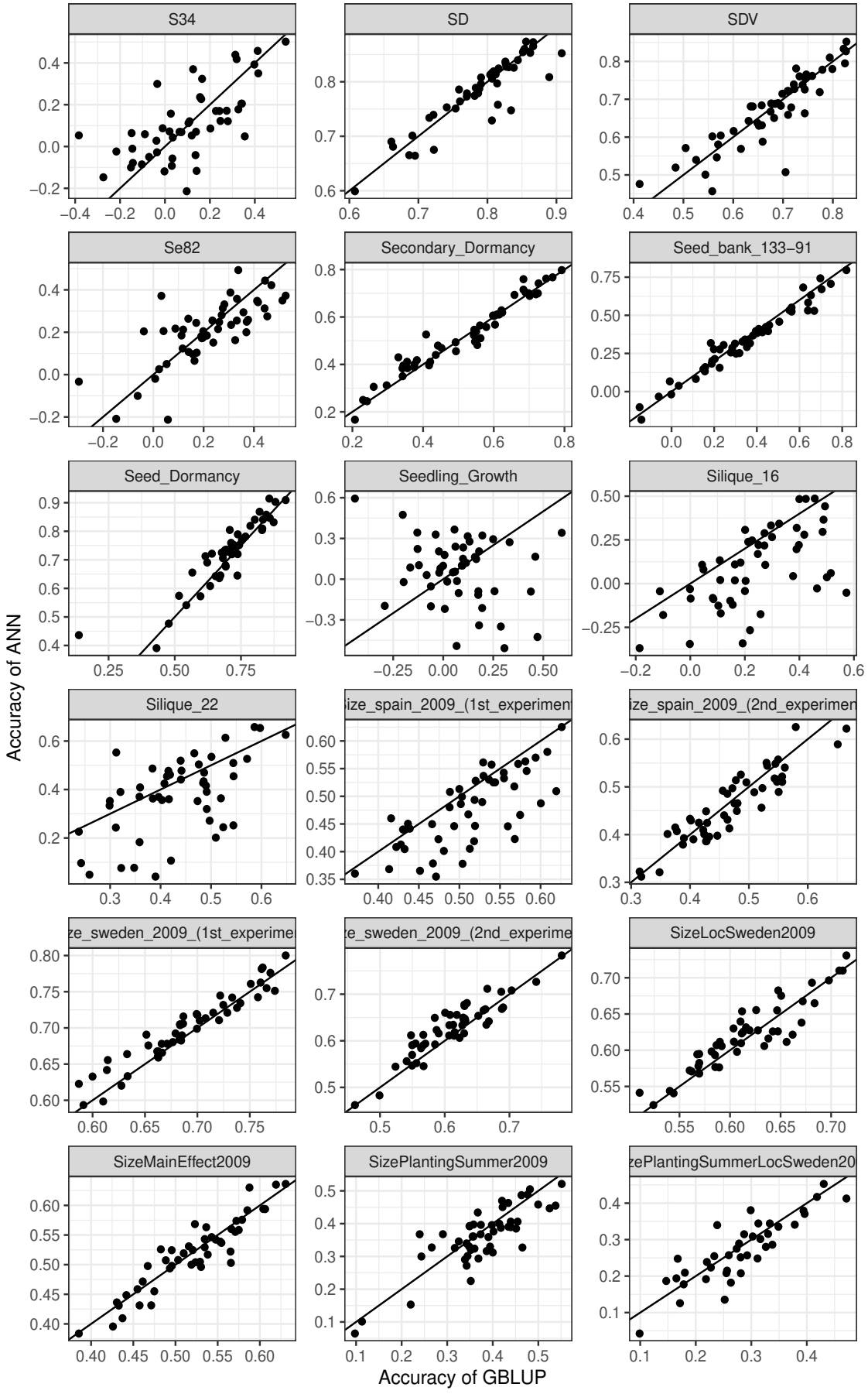

As2_CFU2
As75
At1
At1_CFU2
At2
At2_CFU2
avrB
avrPphB
avrRpm1
avrRpt2
B11
Bacterial_titer
Bs
Bs_CFU2
Ca43
Cd114
Chlorosis_10
Chlorosis_16
Accuracy of ANN
Accuracy of GBLUP

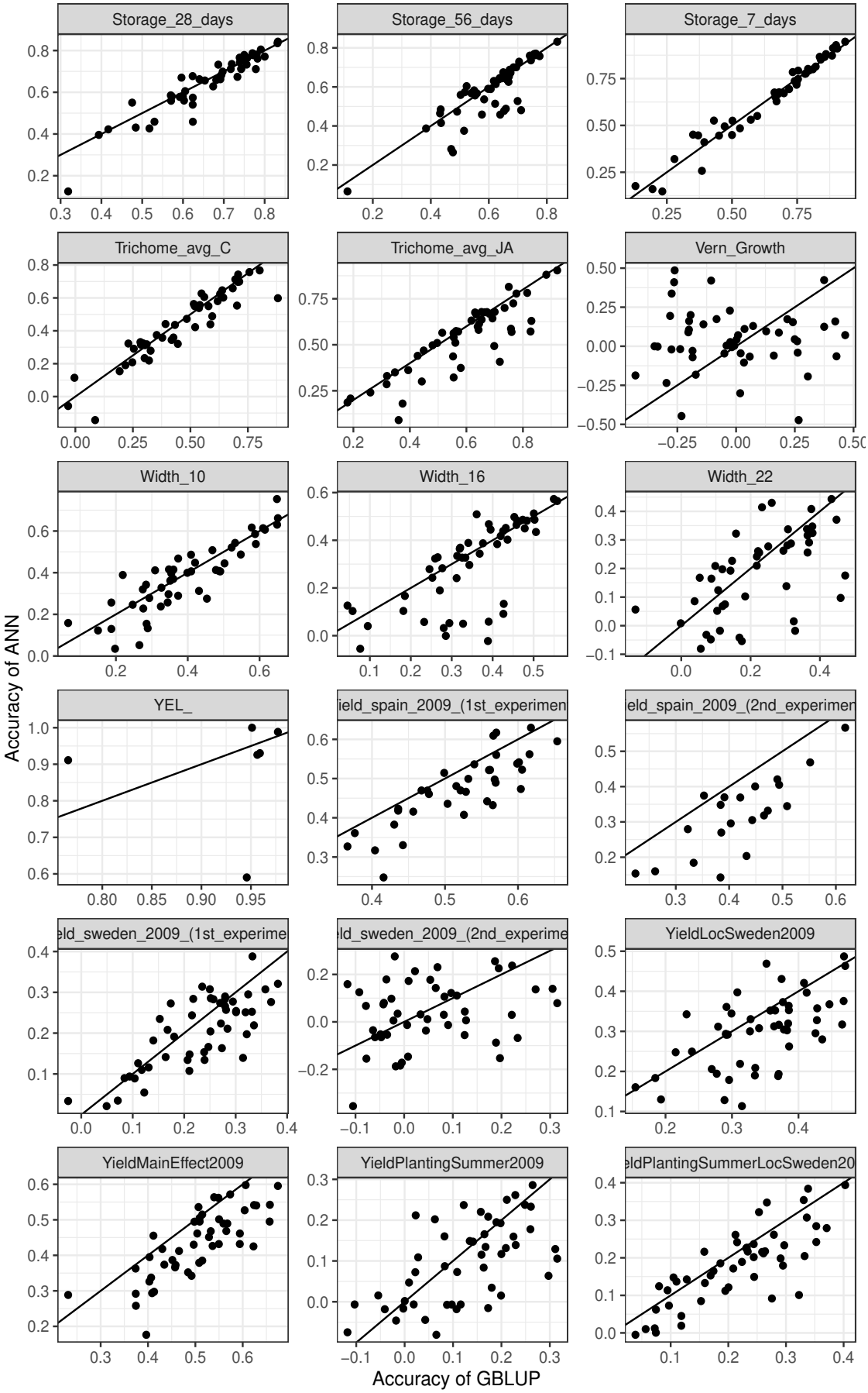

Chlorosis_22
Co59
Cu65
DSDS50
DTF_spain_2008_(1st_experiment
TF_spain_2008_(2nd_experimen
DTF_spain_2009_(1st_experiment
TF_spain_2009_(2nd_experimen
TF_sweden_2008_(1st_experimer
F_sweden_2008_(2nd_experime
TF_sweden_2009_(1st_experimer
F_sweden_2009_(2nd_experime
DTFlocSweden2008
DTFlocSweden2009
DTFmainEffect2008
DTFmainEffect2009
DTFplantingLoc2008
DTFplantingSummer2008
Accuracy of ANN
Accuracy of GBLUP

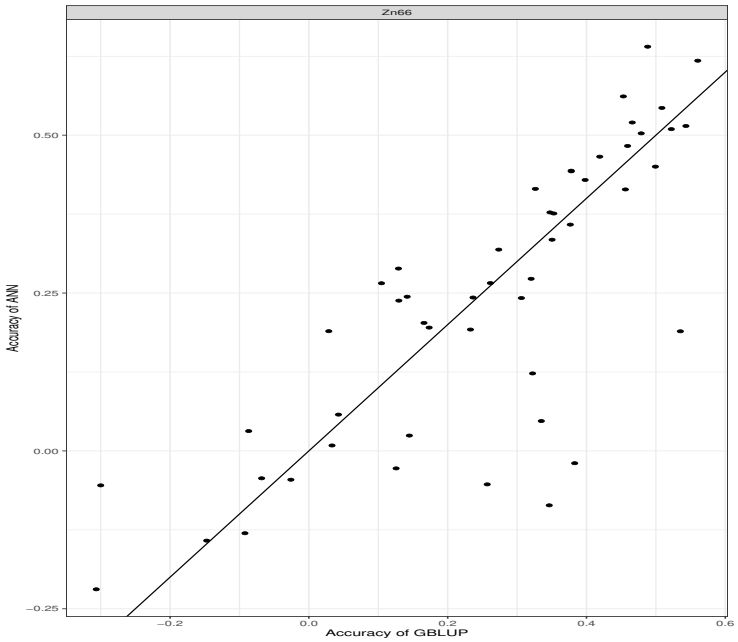

DTFplantingSummer2009
FplantingSummerLocSweden20
DW_
Emco5
Emwa1
Fe56
FLC
FRI
FT_Diameter_Field
FT_Duration_GH
FT_Field
FT_GH
FT10
FT16
FT22
FW_
Germ_10
Germ_16
Accuracy of ANN
Accuracy of GBLUP


Germ_22
Germ_in_dark
Hiks1
Hypocotyl_length
K39
LC_Duration_GH
LD
LDV
Leaf_roll_10
Leaf_roll_16
Leaf_roll_22
Leaf_serr_10
Leaf_serr_16
Leaf_serr_22
LES
LFS_GH
Li7
LN10
Accuracy of ANN
Accuracy of GBLUP


LN16
LN22
LY_
M130T666
M172T666
M216T665
Mature_cell_length
Meristem_zone_length
Mg25
Mn55
Mo98
MT_GH
Na23
Ni60
Noco2
P31
Rosette_Erect_22
RP_GH
Accuracy of ANN
Accuracy of GBLUP


S34
SD
SDV
Se82
Secondary_Dormancy
Seed_bank_133−91
Seed_Dormancy
Seedling_Growth
Silique_16
Silique_22
ize_spain_2009_(1st_experimen
ize_spain_2009_(2nd_experimen
ze_sweden_2009_(1st_experime
e_sweden_2009_(2nd_experime
SizeLocSweden2009
SizeMainEffect2009
SizePlantingSummer2009
zePlantingSummerLocSweden20
Accuracy of ANN
Accuracy of GBLUP


Storage_28_days
Storage_56_days
Storage_7_days
Trichome_avg_C
Trichome_avg_JA
Vern_Growth
Width_10
Width_16
Width_22
YEL_
ield_spain_2009_(1st_experimen
eld_spain_2009_(2nd_experimen
eld_sweden_2009_(1st_experime
eld_sweden_2009_(2nd_experime
YieldLocSweden2009
YieldMainEffect2009
YieldPlantingSummer2009
eldPlantingSummerLocSweden20
Accuracy of ANN
Accuracy of GBLUP



C.2 Haplotype structure of *A. thaliana*

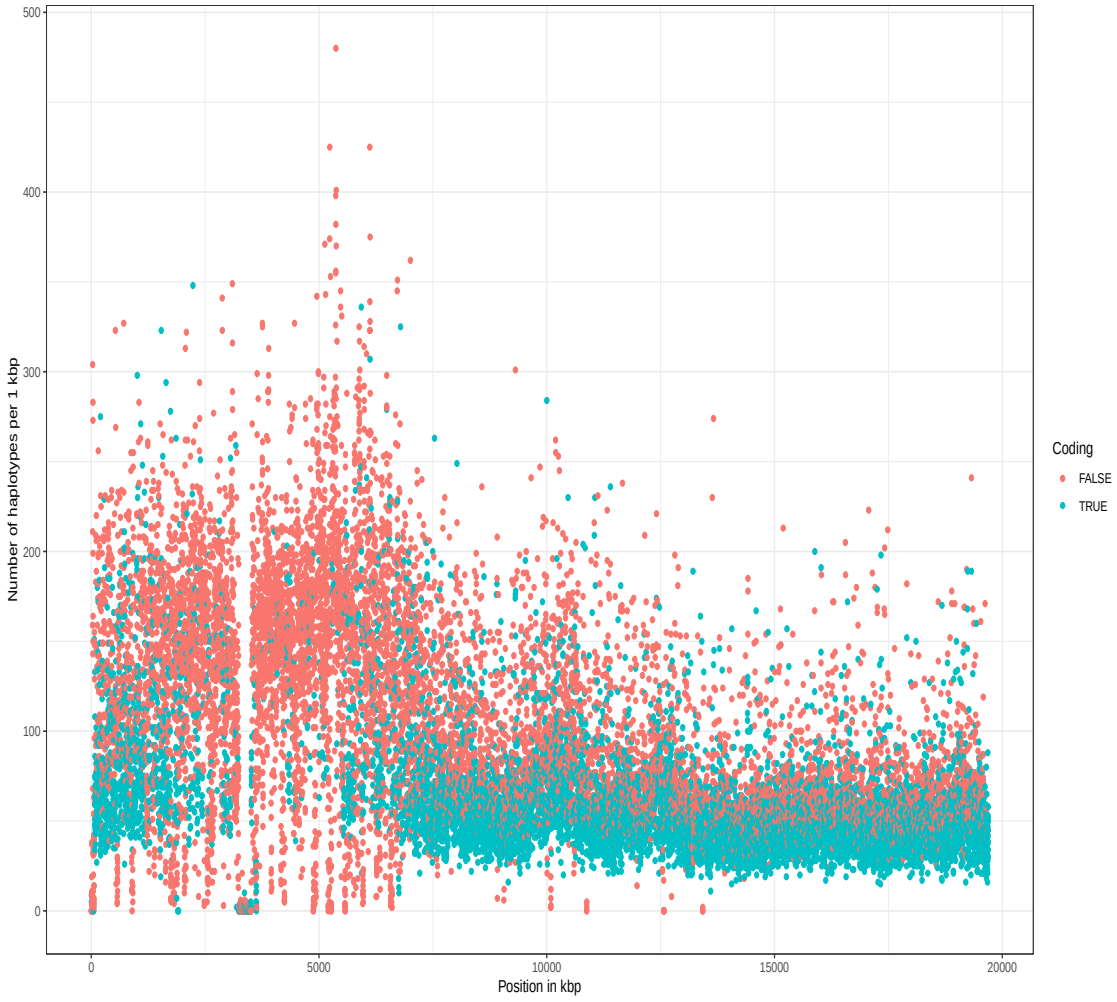


FIGURE C.1: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

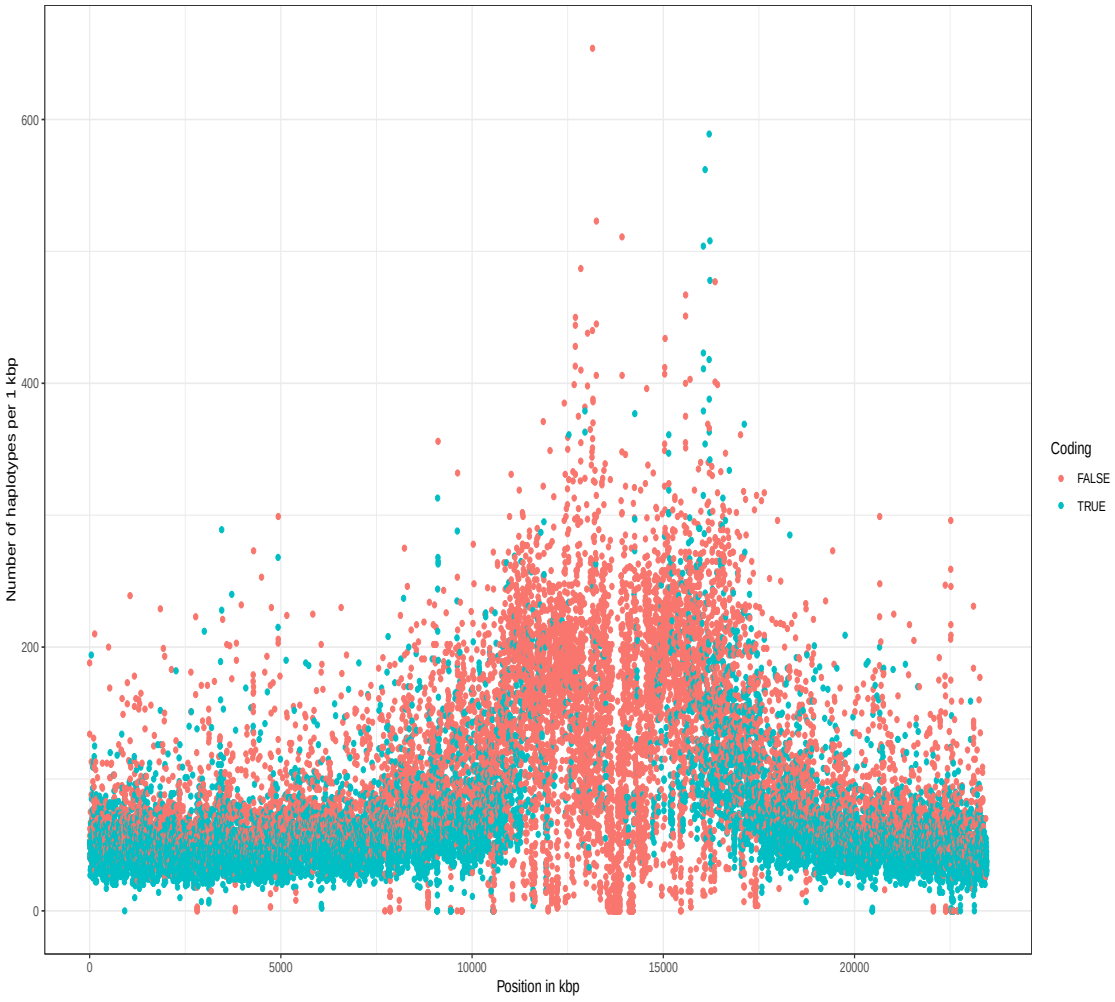


FIGURE C.2: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

FIGURE C.3: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

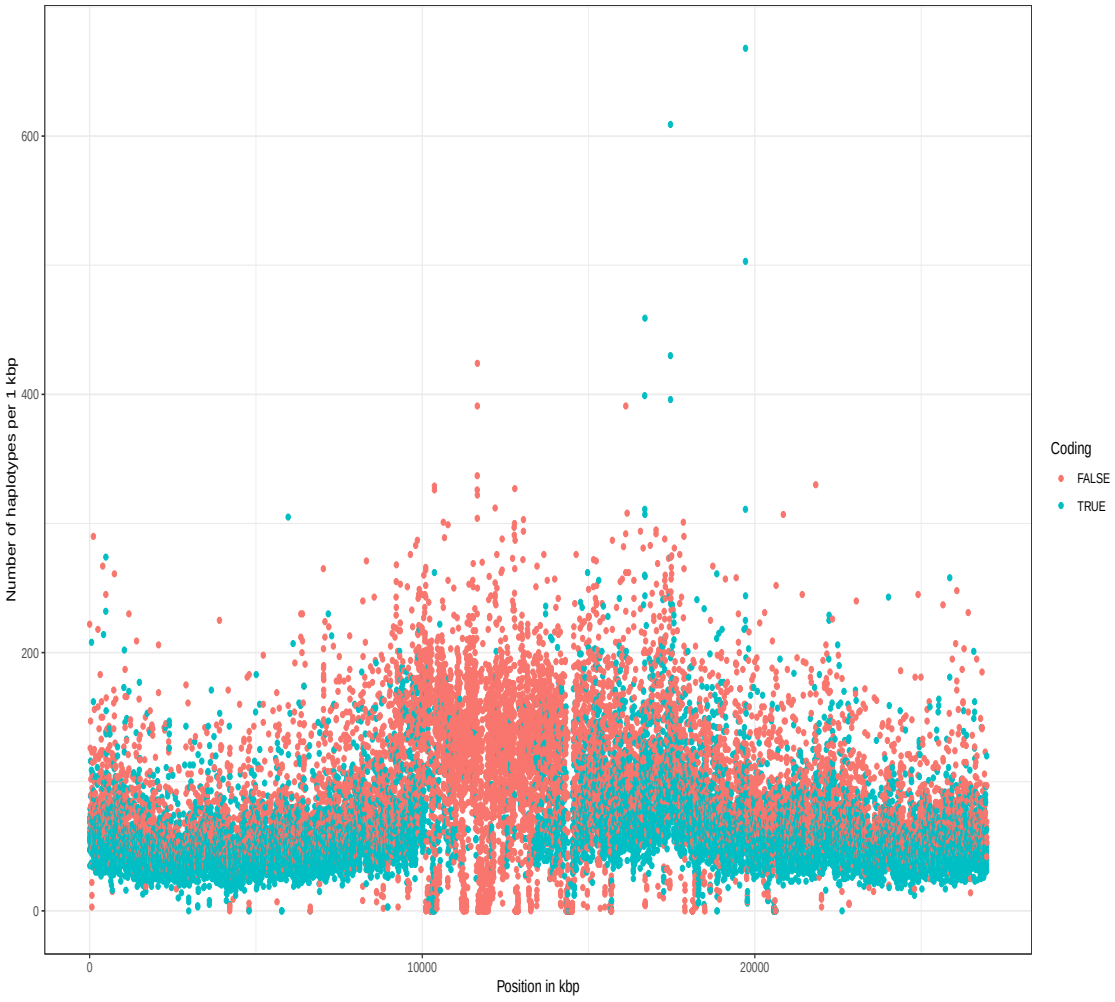


FIGURE C.4: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

Bibliography

ABADI, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

ABADI, Martín et al. (2016). "TensorFlow: A System for Large-scale Machine Learning". In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. OSDI'16. Savannah, GA, USA: USENIX Association, pp. 265–283. ISBN: 978-1-931971-33-1. URL: http://dl.acm.org/citation.cfm?id=3026877.3026899.

ALBRECHT, Theresa et al. (2011). "Genome-based prediction of testcross values in maize". In: *Theoretical and Applied Genetics* 123.2, p. 339.

ALLIER, Antoine et al. (2019). "Usefulness Criterion and post-selection Parental Contributions in Multi-parental Crosses: Application to Polygenic Trait Introgression". In: *G3: Genes, Genomes, Genetics* 9.5, pp. 1469–1479.

ALMEIDA FILHO, Janeo Eustáquio de et al. (2019). "Genomic Prediction of Additive and Non-additive Effects Using Genetic Markers and Pedigrees". In: *G3: Genes, Genomes, Genetics* 9.8, pp. 2739–2748. DOI: 10.1534/g3.119.201004. URL: https://doi.org/10.1534.

ALONSO-BLANCO, Carlos et al. (2016). "1,135 genomes reveal the global pattern of polymorphism in Arabidopsis thaliana". In: *Cell* 166.2, pp. 481–491.

AMIN, Najaf, Cornelia M VAN DUIJN, and Yurii S AULCHENKO (2007). "A genomic background based method for association analysis in related individuals". In: *PloS one* 2.12, e1274.

ANGERMUELLER, Christof et al. (2016). "Deep learning for computational biology". In: *Molecular systems biology* 12.7, p. 878.

ANKENBRAND, Markus et al. (Jan. 2018). "chloroExtractor: extraction and assembly of the chloroplast genome from whole genome shotgun data". In: *The Journal of*

Open Source Software 3.21, p. 464. ISSN: 2475-9066. DOI: 10.21105/joss.00464. URL: http://joss.theoj.org/papers/10.21105/joss.00464.

ANKENBRAND, Markus J. and Frank FÖRSTER (Apr. 2019). *Simulated Arabidopsis thaliana sequencing datasets for chloroplast assembler benchmarking*. DOI: 10.5281/zenodo.2622875. URL: https://doi.org/10.5281/zenodo.2622875.

ANKENBRAND, Markus J. et al. (June 2017). "AliTV—interactive visualization of whole genome comparisons". In: *PeerJ Computer Science* 3, e116. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.116. URL: https://doi.org/10.7717/peerj-cs.116.

ANNICCHIARICO, Paolo et al. (2015). "Accuracy of genomic selection for alfalfa biomass yield in different reference populations". In: *BMC genomics* 16.1, p. 1020.

ARAKI, Motoko and Tetsuya ISHII (2015). "Towards social acceptance of plant breeding by genome editing". In: *Trends in plant science* 20.3, pp. 145–149.

ARCHIBALD, John M (2015). "Endosymbiosis and eukaryotic cell evolution". In: *Current Biology* 25.19, R911–R921.

ARTEAGA, María Clara et al. (2016). "Genomic variation in recently collected maize landraces from Mexico". In: *Genomics Data* 7, pp. 38–45.

ATWELL, S et al. (2010). "Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines". In: *Nature* 465(7298). DOI: 10.1038/nature08800.

AUINGER, Hans-Jürgen et al. (2016). "Model training across multiple breeding cycles significantly improves genomic prediction accuracy in rye (Secale cereale L.)" In: *Theoretical and Applied Genetics* 129.11, pp. 2043–2053.

AZODI, Christina B et al. (2019). "Benchmarking algorithms for genomic prediction of complex traits". In: *bioRxiv*, p. 614479.

BAKKER, Freek T. et al. (Jan. 2016). "Herbarium genomics: plastome sequence assembly from a range of herbarium specimens using an Iterative Organelle Genome Assembly pipeline". en. In: *Biological Journal of the Linnean Society* 117.1, pp. 33–43. ISSN: 00244066. DOI: 10.1111/bij.12642. URL: https://academic.oup.com/biolinnean/article-lookup/doi/10.1111/bij.12642.

BEKELE, Wubishet A et al. (2018). "Haplotype-based genotyping-by-sequencing in oat genome research". In: *Plant biotechnology journal* 16.8, pp. 1452–1463.

BELAMKAR, Vikas et al. (2018). "Genomic Selection in Preliminary Yield Trials in a Winter Wheat Breeding Program". In: *G3: Genes, Genomes, Genetics* 8.8, pp. 2735–2747. DOI: 10.1534/g3.118.200415. URL: https://doi.org/10.1534.

BENDICH, Arnold J. (1987). "Why do chloroplasts and mitochondria contain so many copies of their genome?" In: *BioEssays* 6.6, pp. 279–282. DOI: 10.1002/bies.950060608. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bies.950060608. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/bies.950060608.

BERARDINI, Tanya Z. et al. (2015). "The *Arabidopsis* information resource: Making and mining the "gold standard"annotated reference plant genome". In: *genesis* 53.8, pp. 474–485. DOI: 10.1002/dvg.22877. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/dvg.22877. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/dvg.22877.

BERGSTRA, James S et al. (2011). "Algorithms for hyper-parameter optimization". In: *Advances in neural information processing systems*, pp. 2546–2554.

BERNAL-VASQUEZ, Angela-Maria et al. (2014). "The importance of phenotypic data analysis for genomic prediction-a case study comparing different spatial models in rye". In: *BMC genomics* 15.1, p. 646.

BERNAL-VASQUEZ, Angela-Maria et al. (2017). "Genomic prediction in early selection stages using multi-year data in a hybrid rye breeding program". In: *BMC genetics* 18.1, p. 51.

BERNARDO, R (2010). *Breeding for quantitative traits in plants*. Tech. rep. Stemma Press.

BERNARDO, Rex and Jianming YU (2007). "Prospects for genomewide selection for quantitative traits in maize". In: *Crop Science* 47.3, pp. 1082–1090.

BIAZZI, Elisa et al. (2017). "Genome-wide association mapping and genomic selection for alfalfa (Medicago sativa) forage quality traits". In: *PLoS One* 12.1, e0169234.

BISCARINI, Filippo et al. (2014). "Genome-enabled predictions for binomial traits in sugar beet populations". In: *BMC genetics* 15.1, p. 87.

BLONDEL, Mathieu et al. (2015). "A ranking approach to genomic selection". In: *PloS one* 10.6, e0128570.

BOCK, Ralph (2017). "Witnessing genome evolution: experimental reconstruction of endosymbiotic and horizontal gene transfer". In: *Annual review of genetics* 51, pp. 1–22.

BOTTOU, Léon (1991). "Stochastic gradient learning in neural networks". In: *Proceedings of Neuro-Nımes* 91.8, p. 12.

BOTTOU, Léon and Olivier BOUSQUET (2008). "The Tradeoffs of Large Scale Learning". In: *Advances in Neural Information Processing Systems 20 (NIPS 2007)*. Ed. by J.C. PLATT et al. NIPS Foundation (http://books.nips.cc), pp. 161–168. URL: http://leon.bottou.org/papers/bottou-bousquet-2008.

BOYLE, Evan A, Yang I LI, and Jonathan K PRITCHARD (2017). "An expanded view of complex traits: from polygenic to omnigenic". In: *Cell* 169.7, pp. 1177–1186.

BRAUNER, Pedro C et al. (2018). "Genomic prediction within and among doubled-haploid libraries from maize landraces". In: *Genetics* 210.4, pp. 1185–1196.

BRAUNER, Pedro C et al. (2019). "Testcross performance of doubled haploid lines from European flint maize landraces is promising for broadening the genetic base of elite germplasm". In: *Theoretical and Applied Genetics* 132.6, pp. 1897–1908.

BROOKER, Robert J (1999). *Genetics: analysis & principles*. Addison-Wesley Reading, MA.

BROWNING, Brian L, Ying ZHOU, and Sharon R BROWNING (2018). "A one-penny imputed genome from next-generaiton reference panels". In: *The American Journal of Human Genetics* 103.3, pp. 338–348.

BROWNING, Sharon R and Brian L BROWNING (2007). "Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering". In: *The American Journal of Human Genetics* 81.5, pp. 1084–1097.

BURGESS, Diane and Michael FREELING (2014). "The Most Deeply Conserved Noncoding Sequences in Plants Serve Similar Functions to Those in Vertebrates Despite Large Differences in Evolutionary Rates". In: *The Plant Cell* 26.3, pp. 946–961. ISSN: 1040-4651. DOI: 10.1105/tpc.113.121905. eprint: http://www.plantcell.org/content/26/3/946.full.pdf. URL: http://www.plantcell.org/content/26/3/946.

BUSTOS-KORTS, Daniela et al. (2016a). "Improvement of predictive ability by uniform coverage of the target genetic space". In: *G3: Genes, Genomes, Genetics* 6.11, pp. 3733–3747.

— (2016b). "Improvement of Predictive Ability by Uniform Coverage of the Target Genetic Space". In: *G3: Genes, Genomes, Genetics* 6.11, pp. 3733–3747. DOI: 10.1534/g3.116.035410. URL: https://doi.org/10.1534.

CALUS, MPL, APW DE ROOS, RF VEERKAMP, et al. (2008). "Accuracy of genomic selection using different methods to define haplotypes". In: *Genetics* 178.1, pp. 553–561.

CAMPOS, Gustavo De los and Paulino Perez RODRIGUEZ (2016). *BGLR: Bayesian Generalized Linear Regression*. R package version 1.0.5. URL: https://CRAN.R-project.org/package=BGLR.

CHAN, Cheong Xin and Mark A. RAGAN (2013). "Next-generation phylogenomics". In: *Biology direct* 8, pp. 3–3. ISSN: 1745-6150. DOI: 10.1186/1745-6150-8-3. URL: https://www.ncbi.nlm.nih.gov/pubmed/23339707.

CHANG, Ta Chen and Justin STOLER (2019). "Envirotyping: The Next Leap Forward in the Practice of Precision Medicine?" In: *American journal of ophthalmology* 202, pp. xi–xiii.

CHAT, J. et al. (July 2002). "A Case of Chloroplast Heteroplasmy in Kiwifruit (Actinidia deliciosa) That Is Not Transmitted During Sexual Reproduction". In: *Journal of Heredity* 93.4, pp. 293–300. ISSN: 0022-1503. DOI: 10.1093/jhered/93.4.293. eprint: http://oup.prod.sis.lan/jhered/article-pdf/93/4/293/6454216/293.pdf. URL: https://doi.org/10.1093/jhered/93.4.293.

CHE, Ronglin et al. (2014). "An adaptive permutation approach for genome-wide association study: evaluation and recommendations for use". In: *BioData mining* 7.1, p. 9.

CHENG, Chia-Yi et al. (2017). "Araport11: a complete reannotation of the Arabidopsis thaliana reference genome". In: *The Plant Journal* 89.4, pp. 789–804.

CHOLLET, François et al. (2015). *Keras*. https://keras.io.

COISSAC, Eric et al. (2016). "From barcodes to genomes: extending the concept of DNA barcoding". In: *Molecular Ecology* 25.7, pp. 1423–1428. ISSN: 1365-294X.

DOI: 10.1111/mec.13549. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/mec.13549 (visited on 05/16/2019).

COLLETTE, Andrew (2013). *Python and HDF5*. O'Reilly.

CONSORTIUM, International Wheat Genome Sequencing et al. (2018). "Shifting the limits in wheat research and breeding using a fully annotated reference genome". In: *Science* 361.6403, eaar7191.

CORRIVEAU, Joseph L. and Annette W. COLEMAN (1988). "Rapid Screening Method to Detect Potential Biparental Inheritance of Plastid DNA and Results for Over 200 Angiosperm Species". In: *American Journal of Botany* 75.10, pp. 1443–1458. ISSN: 00029122, 15372197. URL: http://www.jstor.org/stable/2444695.

CROSSA, José et al. (2010). "Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers". In: *Genetics*.

CROSSA, José et al. (2016). "Genomic Prediction of Gene Bank Wheat Landraces". In: *G3: Genes, Genomes, Genetics* 6.7, pp. 1819–1834. DOI: 10.1534/g3.116.029637. URL: https://doi.org/10.1534.

CROSSA, José et al. (2017). "Genomic selection in plant breeding: Methods, models, and perspectives". In: *Trends in plant science*.

CROSSA, Jose et al. (2019). "DEEP KERNEL AND DEEP LEARNING FOR GENOME-BASED PREDICTION OF SINGLE TRAITS IN MULTI-ENVIRONMENT BREEDING TRIALS". In: *Frontiers in Genetics* 10, p. 1168.

CUEVAS, Jaime et al. (2017). "Bayesian genomic prediction with genotype× environment interaction kernel models". In: *G3: Genes, Genomes, Genetics* 7.1, pp. 41–53.

CUEVAS, Jaime et al. (2019a). "Deep Kernel for Genomic and Near Infrared Predictions in Multi-environment Breeding Trials". In: *G3: Genes, Genomes, Genetics* 9.9, pp. 2913–2924. DOI: 10.1534/g3.119.400493. URL: https://doi.org/10.1534.

CUEVAS, Jaime et al. (2019b). "Deep Kernel for Genomic and Near Infrared Predictions in Multi-environment Breeding Trials". In: *G3: Genes, Genomes, Genetics* 9.9, pp. 2913–2924.

CUYABANO, Beatriz CD, Guosheng SU, and Mogens S LUND (2014). "Genomic prediction of genetic merit using LD-based haplotypes in the Nordic Holstein population". In: *BMC genomics* 15.1, p. 1171.

DANIELL, Henry et al. (June 23, 2016). "Chloroplast genomes: diversity, evolution, and applications in genetic engineering". In: *Genome Biology* 17. ISSN: 1474-7596. DOI: 10.1186/s13059-016-1004-2. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4918201/ (visited on 05/20/2019).

DARWIN, Charles (1859). *On the Origin of Species by Means of Natural Selection*. or the Preservation of Favored Races in the Struggle for Life. London: Murray.

DE LOS CAMPOS, Gustavo et al. (2009). "Predicting quantitative traits with regression models for dense molecular markers and pedigree". In: *Genetics* 182.1, pp. 375–385.

DE RUBEIS, Silvia et al. (2014). "Synaptic, transcriptional and chromatin genes disrupted in autism". In: *Nature* 515.7526, p. 209.

DEINER, Kristy et al. (2017). "Environmental DNA metabarcoding: Transforming how we survey animal and plant communities". In: *Molecular Ecology* 26.21, pp. 5872–5895. ISSN: 1365-294X. DOI: 10.1111/mec.14350. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/mec.14350 (visited on 05/21/2019).

DESTA, Zeratsion Abera and Rodomiro ORTIZ (2014). "Genomic selection: genome-wide prediction in plant improvement". In: *Trends in plant science* 19.9, pp. 592–601.

DIERCKXSENS, Nicolas, Patrick MARDULYN, and Guillaume SMITS (Feb. 28, 2017). "NOVOPlasty: de novo assembly of organelle genomes from whole genome data". In: *Nucleic Acids Research* 45.4, e18–e18. ISSN: 0305-1048. DOI: 10.1093/nar/gkw955. URL: https://academic.oup.com/nar/article/45/4/e18/2290925 (visited on 01/26/2018).

DITTBERNER, Hannes et al. (2018). "Natural variation in stomata size contributes to the local adaptation of water-use efficiency in Arabidopsis thaliana". In: *Molecular ecology* 27.20, pp. 4052–4065.

Docker Hub Group for Benchmark Project. URL: https://cloud.docker.com/u/chloroextractorteam/.

DOS SANTOS, Cicero and Maira GATTI (2014). "Deep convolutional neural networks for sentiment analysis of short texts". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78.

DOZAT, Timothy (2016). "Incorporating nesterov momentum into adam". In:

EL-DIEN, Omnia Gamal et al. (2016). "Implementation of the Realized Genomic Relationship Matrix to Open-Pollinated White Spruce Family Testing for Disentangling Additive from Nonadditive Genetic Effects". In: *G3: Genes, Genomes, Genetics* 6.3, pp. 743–753. DOI: 10.1534/g3.115.025957. URL: https://doi.org/10.1534.

ELIAS, Ani A et al. (2018a). "Improving genomic prediction in cassava field experiments by accounting for interplot competition". In: *G3: Genes, Genomes, Genetics* 8.3, pp. 933–944.

— (2018b). "Improving genomic prediction in cassava field experiments using spatial analysis". In: *G3: Genes, Genomes, Genetics* 8.1, pp. 53–62.

ENCISO-RODRIGUEZ, Felix et al. (2018). "Genomic selection for late blight and common scab resistance in tetraploid potato (Solanum tuberosum)". In: *G3: Genes, Genomes, Genetics* 8.7, pp. 2471–2481.

ENDELMAN, Jeffrey B. et al. (Mar. 2018). "Genetic Variance Partitioning and Genome-Wide Prediction with Allele Dosage Information in Autotetraploid Potato". In: *Genetics* 209.1, pp. 77–87. DOI: 10.1534/genetics.118.300685. URL: https://doi.org/10.1534/genetics.118.300685.

FALCONER, DS and TFC MACKAY (1996). "Introduction to quantitative genetics. 1996". In: *Harlow, Essex, UK: Longmans Green* 3.

FAN, Jianqing, Fang HAN, and Han LIU (2014). "Challenges of big data analysis". In: *National science review* 1.2, pp. 293–314.

FELDMAN, Moshe and Avraham A LEVY (2012). "Genome evolution due to allopolyploidization in wheat". In: *Genetics* 192.3, pp. 763–774.

FISHER, Ronald A (1919). "XV.—The correlation between relatives on the supposition of Mendelian inheritance." In: *Earth and Environmental Science Transactions of the Royal Society of Edinburgh* 52.2, pp. 399–433.

FISHER, Ronald A and Winifred A MACKENZIE (1923). "Studies in crop variation. II. The manurial response of different potato varieties". In: *The Journal of Agricultural Science* 13.3, pp. 311–320.

FORSBERG, Simon K. G. et al. (2015). "The Multi-allelic Genetic Architecture of a Variance-Heterogeneity Locus for Molybdenum Concentration in Leaves Acts as a Source of Unexplained Additive Genetic Variance". In: *PLOS Genetics* None. DOI: 10.1371/journal.pgen.1005648.

FÖRSTER, Frank and Markus J. ANKENBRAND (May 2019). *chloroExtractorTeam/benchmark: Benchmark container setup v2.0.1*. DOI: 10.5281/zenodo.2628061. URL: https://doi.org/10.5281/zenodo.2628061.

FREUDENTHAL, Jan A. et al. (2019a). "GWAS-Flow: A GPU accelerated framework for efficient permutation based genome-wide association studies". In: DOI: 10.1101/783100. URL: https://doi.org/10.1101.

FREUDENTHAL, Jan A et al. (2019b). "The landscape of chloroplast genome assembly tools". In: *bioRxiv*, p. 665869.

FRIEBE, B et al. (2000). "Development of a complete set of Triticum aestivum-Aegilops speltoides chromosome addition lines". In: *Theoretical and Applied Genetics* 101.1-2, pp. 51–58.

FUENTES, Ignacia et al. (2014). "Horizontal genome transfer as an asexual path to the formation of new species". In: *Nature* 511.7508, p. 232.

GAPARE, Washington et al. (2018). "Historical Datasets Support Genomic Selection Models for the Prediction of Cotton Fiber Quality Phenotypes Across Multiple Environments". In: *G3: Genes, Genomes, Genetics* 8.5, pp. 1721–1732.

GERLAND, Patrick et al. (2014). "World population stabilization unlikely this century". In: *Science* 346.6206, pp. 234–237.

GHOSH, Sreya et al. (2018). "Speed breeding in growth chambers and glasshouses for crop breeding and model plant research". In: *Nature protocols* 13.12, p. 2944.

GIANOLA, Daniel (2013). "Priors in whole-genome regression: the Bayesian alphabet returns". In: *Genetics* 194.3, pp. 573–596.

GIANOLA, Daniel and Johannes BCHM van KAAM (2008). "Reproducing kernel Hilbert spaces regression methods for genomic assisted prediction of quantitative traits". In: *Genetics* 178.4, pp. 2289–2303.

GIANOLA, Daniel and Guilherme JM ROSA (2015). "One hundred years of statistical developments in animal breeding". In: *Annu. Rev. Anim. Biosci.* 3.1, pp. 19–56.

GIANOLA, Daniel et al. (2009). "Additive genetic variability and the Bayesian alphabet". In: *Genetics* 183.1, pp. 347–363.

GIANOLA, Daniel et al. (2016). "Genome-Wide Association Studies with a Genomic Relationship Matrix: A Case Study with Wheat and Arabidopsis". In: *G3: Genes, Genomes, Genetics* 6.10, pp. 3241–3256. DOI: 10.1534/g3.116.034256. URL: https://doi.org/10.1534.

GitHub Repository for Benchmark Project. URL: https://github.com/chloroExtractorTeam/benchmark.

GLOROT, Xavier, Antoine BORDES, and Yoshua BENGIO (2011). "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323.

GODDARD, Michael E and Ben J HAYES (2009). "Mapping genes for complex traits in domestic animals and their use in breeding programmes". In: *Nature Reviews Genetics* 10.6, p. 381.

GODDARD, Michael E, Ben J HAYES, and Theo HE MEUWISSEN (2011). "Using the genomic relationship matrix to predict the accuracy of genomic selection". In: *Journal of animal breeding and genetics* 128.6, pp. 409–421.

GONZÁLEZ-CAMACHO, JM et al. (2012). "Genome-enabled prediction of genetic values using radial basis function neural networks". In: *Theoretical and Applied Genetics* 125.4, pp. 759–771.

GONZÁLEZ-CAMACHO, Juan Manuel et al. (2016). "Genome-enabled prediction using probabilistic neural network classifiers". In: *BMC genomics* 17.1, p. 208.

GONZÁLEZ-CAMACHO, Juan Manuel et al. (2018). "Applications of machine learning methods to genomic selection in breeding wheat for rust resistance". In: *The plant genome* 11.2.

GOODFELLOW, Ian, Yoshua BENGIO, and Aaron COURVILLE (2016). *Deep learning*. MIT press.

GOUY, Matthieu et al. (2013). "Experimental assessment of the accuracy of genomic selection in sugarcane". In: *Theoretical and applied genetics* 126.10, pp. 2575–2586.

GREEN, Beverley R. (2011). "Chloroplast genomes of photosynthetic eukaryotes". In: *The Plant Journal* 66.1, pp. 34–44. ISSN: 1365-313X. DOI: 10.1111/j.1365-313X.2011.04541.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-313X.2011.04541.x (visited on 05/16/2019).

GRENIER, Cécile et al. (2015). "Accuracy of genomic selection in a rice synthetic population developed for recurrent selection breeding". In: *PloS one* 10.8, e0136594.

GRINBERG, Nastasiya F, Oghenejokpeme I ORHOBOR, and Ross D KING (2018). "An Evaluation of Machine-learning for Predicting Phenotype: Studies in Yeast, Rice and Wheat". In: *BioRxiv*, p. 105528.

GUO, Zhigang et al. (2013). "Accuracy of across-environment genome-wide prediction in maize nested association mapping populations". In: *G3: Genes, Genomes, Genetics* 3.2, pp. 263–272.

HABIER, David, Rohan L FERNANDO, and Jack CM DEKKERS (2007). "The impact of genetic relationship information on genome-assisted breeding values". In: *Genetics* 177.4, pp. 2389–2397.

HABIER, David et al. (2011). "Extension of the Bayesian alphabet for genomic selection". In: *BMC bioinformatics* 12.1, p. 186.

HAHNLOSER, Richard HR et al. (2000). "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit". In: *Nature* 405.6789, p. 947.

HARRIS, BL, DL JOHNSON, RJ SPELMAN, et al. (2009). "Genomic selection in New Zealand and the implications for national genetic evaluation." In: *ICAR Technical Series* 13, pp. 325–330.

HASSEN, Manel Ben et al. (May 2018). "Genomic Prediction Accounting for Genotype by Environment Interaction Offers an Effective Framework for Breeding Simultaneously for Adaptation to an Abiotic Stress and Performance Under Normal Cropping Conditions in Rice". In: *G3: Genes, Genomes, Genetics* 8.7, pp. 2319–2332. DOI: 10.1534/g3.118.200098. URL: https://doi.org/10.1534/g3.118.200098.

HAWKINS, Charles and Long-Xi YU (2018). "Recent progress in alfalfa (Medicago sativa L.) genomics and genomic selection". In: *The Crop Journal* 6.6, pp. 565–575.

HAYES, Ben and Mike GODDARD (2010). "Genome-wide association and genomic selection in animal breeding". In: *Genome* 53.11, pp. 876–883.

HAYES, BJ, ME GODDARD, et al. (2001). "Prediction of total genetic value using genome-wide dense marker maps". In: *Genetics* 157.4, pp. 1819–1829.

HE, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

HEFFNER, Elliot L, Jean-Luc JANNINK, and Mark E SORRELLS (2011). "Genomic selection accuracy using multifamily prediction models in a wheat breeding program". In: *The Plant Genome* 4.1, pp. 65–75.

HEFFNER, Elliot L et al. (2010). "Plant breeding with genomic selection: gain per unit time and cost". In: *Crop science* 50.5, pp. 1681–1690.

HENDERSON, Charles R (1975). "Best linear unbiased estimation and prediction under a selection model". In: *Biometrics*, pp. 423–447.

HESLOT, Nicolas et al. (2012). "Genomic selection in plant breeding: a comparison of models". In: *Crop science* 52.1, pp. 146–160.

HILL, William G, Michael E GODDARD, and Peter M VISSCHER (2008). "Data and theory point to mainly additive genetic variance for complex traits". In: *PLoS genetics* 4.2, e1000008.

HINTON, Geoffrey, Nitish SRIVASTAVA, and Kevin SWERSKY (2012). "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent". In: *Cited on* 14, p. 8.

HIRSCHHORN, Joel N. and Mark J. DALY (2005). "Genome-wide association studies for common diseases and complex traits". In: *Nature Reviews Genetics* 6.2, pp. 95–108. ISSN: 1471-0064. DOI: 10.1038/nrg1521. URL: https://doi.org/10.1038/nrg1521.

HÖLKER, Armin C et al. (2019). "European maize landraces made accessible for plant breeding and genome-based studies". In: *Theoretical and Applied Genetics*, pp. 1–13.

HOLLIDAY, Jason A., Tongli WANG, and Sally AITKEN (2012). "Predicting Adaptive Phenotypes From Multilocus Genotypes in Sitka Spruce (Picea sitchensis) Using Random Forest". In: *G3: Genes, Genomes, Genetics* 2.9, pp. 1085–1093. DOI: 10.1534/g3.112.002733. URL: https://doi.org/10.1534.

HORTON, Matthew W et al. (2012). "Genome-wide patterns of genetic variation in worldwide Arabidopsis thaliana accessions from the RegMap panel". In: *Nature genetics* 44.2, p. 212.

HOWARD, Réka et al. (2019). "Joint Use of Genome, Pedigree, and Their Interaction with Environment for Predicting the Performance of Wheat Lines in New Environments". In: *G3: Genes, Genomes, Genetics* 9.9, pp. 2925–2934. DOI: 10.1534/g3.119.400508. URL: https://doi.org/10.1534.

HU, Yaodong et al. (2015). "Prediction of plant height in Arabidopsis thaliana using DNA methylation data". In: *Genetics* 201.2, pp. 779–793.

ISIK, F (2013). *Genomic Relationships and GBLUP*.

JAN, Habib U et al. (2016). "Genomic prediction of testcross performance in canola (Brassica napus)". In: *PLoS One* 11.1, e0147769.

JANOCHA, Katarzyna and Wojciech Marian CZARNECKI (2017). "On loss functions for deep neural networks in classification". In: *arXiv preprint arXiv:1702.05659*.

JARAMILLO-CORREA, Juan-Pablo et al. (2014). "Molecular Proxies for Climate Maladaptation in a Long-Lived Tree (Pinus pinasterAiton, Pinaceae)". In: *Genetics* 199.3, pp. 793–807. DOI: 10.1534/genetics.114.173252. URL: https://doi.org/10.1534.

JARQUIN, Diego, James SPECHT, and Aaron LORENZ (2016). "Prospects of Genomic Prediction in the USDA Soybean Germplasm Collection: Historical Data Creates Robust Models for Enhancing Selection of Accessions". In: *G3: Genes, Genomes, Genetics* 6.8, pp. 2329–2341. DOI: 10.1534/g3.116.031443. URL: https://doi.org/10.1534.

JETTE, Morris A., Andy B. YOO, and Mark GRONDONA (2002). "SLURM: Simple Linux Utility for Resource Management". In: *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*. Springer-Verlag, pp. 44–60.

JIANG, Yong and Jochen C REIF (2015). "Modeling epistasis in genomic selection". In: *Genetics* 201.2, pp. 759–768.

JIN, Jian-Jun et al. (Mar. 2018). "GetOrganelle: a simple and fast pipeline for de novo assembly of a complete circular chloroplast genome using genome skimming data". In: *bioRxiv*. DOI: 10.1101/256479. URL: http://biorxiv.org/lookup/doi/10.1101/256479.

KADAM, Dnyaneshwar C et al. (2016). "Genomic prediction of single crosses in the early stages of a maize hybrid breeding pipeline". In: *G3: Genes, Genomes, Genetics* 6.11, pp. 3443–3453.

KAINER, David et al. (2018). "Accuracy of Genomic Prediction for Foliar Terpene Traits in Eucalyptus polybractea". In: *G3: Genes, Genomes, Genetics* 8.8, pp. 2573–2583. DOI: 10.1534/g3.118.200443. URL: https://doi.org/10.1534.

KALO, P et al. (2004). "Comparative mapping between Medicago sativa and Pisum sativum". In: *Molecular Genetics and Genomics* 272.3, pp. 235–246.

KANG, Hyun Min et al. (2010). "Variance component model to account for sample structure in genome-wide association studies". In: *Nature genetics* 42.4, p. 348.

KÄRKKÄINEN, Hanni P and Mikko J SILLANPÄÄ (2012). "Back to basics for Bayesian model building in genomic selection". In: *Genetics* 191.3, pp. 969–987.

KHOURY, Colin K et al. (2016). "Origins of food crops connect countries worldwide". In: *Proceedings of the Royal Society B: Biological Sciences* 283.1832, p. 20160792.

KINGMA, Diederik P and Jimmy BA (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

KINGSBURY, Noel (2009). *Hybrid: the history and science of plant breeding*. University of Chicago Press.

KINGSOLVER, Joel G et al. (2001). "The strength of phenotypic selection in natural populations". In: *The American Naturalist* 157.3, pp. 245–261.

KOCH, Marcus A and Michaela MATSCHINGER (2007). "Evolution and genetic differentiation among relatives of Arabidopsis thaliana". In: *Proceedings of the National Academy of Sciences* 104.15, pp. 6272–6277.

KORTE, Arthur and Ashley FARLOW (2013). "The advantages and limitations of trait analysis with GWAS: a review". In: *Plant methods* 9.1, p. 29.

KORTE, Arthur et al. (2012). "A mixed-model approach for genome-wide association studies of correlated traits in structured populations". In: *Nature genetics* 44.9, p. 1066.

KRAUSE, Margaret R. et al. (2019). "Hyperspectral Reflectance-Derived Relationship Matrices for Genomic Prediction of Grain Yield in Wheat". In: *G3: Genes, Genomes, Genetics*, g3.200856.2018. DOI: 10.1534/g3.118.200856. URL: https://doi.org/10.1534.

KUMAR, Rachana A., Delene J. OLDENBURG, and Arnold J. BENDICH (Sept. 2014). "Changes in DNA damage, molecular integrity, and copy number for plastid DNA and mitochondrial DNA during maize development". In: *Journal of Experimental Botany* 65.22, pp. 6425–6439. ISSN: 0022-0957. DOI: 10.1093/jxb/eru359. eprint: http://oup.prod.sis.lan/jxb/article-pdf/65/22/6425/16935653/eru359.pdf. URL: https://doi.org/10.1093/jxb/eru359.

KUMAR, Satish et al. (2015). "Genome-Enabled Estimates of Additive and Nonadditive Genetic Variances and Prediction of Apple Phenotypes Across Environments". In: *G3: Genes, Genomes, Genetics* 5.12, pp. 2711–2718. DOI: 10.1534/g3.115.021105. URL: https://doi.org/10.1534.

KURTZER, Gregory M, Vanessa SOCHAT, and Michael W BAUER (2017). "Singularity: Scientific containers for mobility of compute". In: *PloS one* 12.5, e0177459.

KUTSCHERA, Ulrich and Karl J NIKLAS (2005). "Endosymbiosis, cell evolution, and speciation". In: *Theory in Biosciences* 124.1, pp. 1–24.

KWONG, Qi Bin et al. (2017). "Genomic selection in commercial perennial crops: applicability and improvement in oil palm (Elaeis Guineensis Jacq.)" In: *Scientific reports* 7.1, p. 2872.

LAN, Kun et al. (2018). "A survey of data mining and deep learning in bioinformatics". In: *Journal of medical systems* 42.8, p. 139.

LECUN, Yann, Yoshua BENGIO, and Geoffrey HINTON (2015). "Deep learning". In: *nature* 521.7553, p. 436.

LECUN, Yann et al. (1999). "Object recognition with gradient-based learning". In: *Shape, contour and grouping in computer vision*. Springer, pp. 319–345.

LEGARRA, Andres, Danila A.L LOURENCO, and Zulma G. VITEZICA (2018). "Bases for Genomic Prediction". In:

LEHERMEIER, Christina et al. (2014). "Usefulness of multiparental populations of maize (Zea mays L.) for genome-based prediction". In: *Genetics* 198.1, pp. 3–16.

LEINONEN, Rasko et al. (Nov. 2010). "The Sequence Read Archive". In: *Nucleic Acids Research* 39.suppl_1, pp. D19–D21. ISSN: 0305-1048. DOI: 10.1093/nar/gkq1019. eprint: http://oup.prod.sis.lan/nar/article-pdf/39/suppl_1/D19/7624335/gkq1019.pdf. URL: https://doi.org/10.1093/nar/gkq1019.

LEUTENEGGER, Anne-Louise et al. (2003). "Estimation of the inbreeding coefficient through use of genomic data". In: *The American Journal of Human Genetics* 73.3, pp. 516–523.

LI, Bo et al. (2018). "Genomic prediction of breeding values using a subset of SNPs identified by three machine learning methods". In: *Frontiers in genetics* 9, p. 237.

LI, Heng (2018). "Minimap2: pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18, pp. 3094–3100.

LI, Jia-Yang, Jun WANG, and Robert S ZEIGLER (2014). "The 3,000 rice genomes project: new opportunities and challenges for future rice research". In: *GigaScience* 3.1, p. 8.

LI, Peijin et al. (2014). "Multiple FLC haplotypes defined by independent cis-regulatory variation underpin life history diversity in Arabidopsis thaliana". In: *Genes & Development* 28.15, pp. 1635–1640.

LI, Xuehui and E Charles BRUMMER (2012). "Applied genetics and genomics in alfalfa breeding". In: *Agronomy* 2.1, pp. 40–61.

LI, Xuehui et al. (2015). "Genomic prediction of biomass yield in two selection cycles of a tetraploid alfalfa breeding population". In: *The Plant Genome* 8.2.

LI, Yan et al. (2010). "Association mapping of local climate-sensitive quantitative trait loci in Arabidopsis thaliana". In: *PNAS* 107. DOI: 10.1073/pnas.1007431107.

LIN, Peng et al. (2010). "A new statistic to evaluate imputation reliability". In: *PloS one* 5.3, e9697.

LIPPERT, Christoph et al. (2014). "LIMIX: genetic analysis of multiple traits". In: *bioRxiv*. DOI: 10.1101/003905. eprint: https://www.biorxiv.org/content/

early/2014/05/22/003905.full.pdf. URL: https://www.biorxiv.org/content/early/2014/05/22/003905.

LITJENS, Geert et al. (2017). "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42, pp. 60–88.

LIU, Shengyi, Rod SNOWDON, and Boulos CHALHOUB (2018). *The Brassica Napus Genome*. Springer.

LOPEZ-CRUZ, Marco et al. (2015). "Increased Prediction Accuracy in Wheat Breeding Trials Using a Marker x Environment Interaction Genomic Selection Model". In: *G3: Genes, Genomes, Genetics* 5.4, pp. 569–582. DOI: 10.1534/g3.114.016097. URL: https://doi.org/10.1534.

LUO, Xiang et al. (2017). "Genomic prediction of genotypic effects with epistasis and environment interactions for yield-related traits of rapeseed (Brassica napus L.)" In: *Frontiers in genetics* 8, p. 15.

LYNCH, Michael, Bruce WALSH, et al. (1998). *Genetics and analysis of quantitative traits*. Vol. 1. Sinauer Sunderland, MA.

MA, Wenlong et al. (2017). "DeepGS: Predicting phenotypes from genotypes using Deep Learning". In: *bioRxiv*, p. 241414.

MAMOSHINA, Polina et al. (2016). "Applications of deep learning in biomedicine". In: *Molecular pharmaceutics* 13.5, pp. 1445–1454.

MARTIN, William et al. (Sept. 17, 2002). "Evolutionary analysis of Arabidopsis, cyanobacterial, and chloroplast genomes reveals plastid phylogeny and thousands of cyanobacterial genes in the nucleus". In: *Proceedings of the National Academy of Sciences of the United States of America* 99.19, pp. 12246–12251. ISSN: 0027-8424. DOI: 10.1073/pnas.182432999. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC129430/ (visited on 05/20/2019).

MARTINI, Johannes WR et al. (2017). "Genomic prediction with epistasis models: on the marker-coding-dependent performance of the extended GBLUP and properties of the categorical epistasis model (CE)". In: *BMC bioinformatics* 18.1, p. 3.

MARULANDA, Jose J et al. (2016). "Optimum breeding strategies using genomic selection for hybrid breeding in wheat, maize, rye, barley, rice and triticale". In: *Theoretical and applied genetics* 129.10, pp. 1901–1913.

MARVIN, Minsky and Papert SEYMOUR (1969). *Perceptrons*.

MAYER, Manfred et al. (2017). "Is there an optimum level of diversity in utilization of genetic resources?" In: *Theoretical and applied genetics* 130.11, pp. 2283–2295.

MCKAIN, Michael and AFINIT (Sept. 2017). *Mrmckain/Fast-Plast: Fast-Plast V.1.2.6*. DOI: 10.5281/zenodo.973887. URL: https://zenodo.org/record/973887.

MCKINNEY, Brett and Nicholas PAJEWSKI (2012). "Six degrees of epistasis: statistical network models for GWAS". In: *Frontiers in genetics* 2, p. 109.

MCKINNEY, Wes (2010). "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der WALT and Jarrod MILLMAN, pp. 51 –56.

MEIJÓ, Mónica et al. (2014). "Genome-wide association study using cellular traits identifies a new regulator of root development in Arabidopsis". In: *Nature Genetics* 46. DOI: 10.1038/ng.2824.

MERESCHKOWSKY, Constantin (1905). "Uber natur und ursprung der chromatophoren im pflanzenreiche". In: *Biologisches Centralblatt* 25, pp. 293–604.

MERKEL, Dirk (2014). "Docker: lightweight linux containers for consistent development and deployment". In: *Linux Journal* 2014.239, p. 2.

MICHIE, Donald, David J SPIEGELHALTER, CC TAYLOR, et al. (1994). "Machine learning". In: *Neural and Statistical Classification* 13.

MIN, Seonwoo, Byunghan LEE, and Sungroh YOON (2017). "Deep learning in bioinformatics". In: *Briefings in bioinformatics* 18.5, pp. 851–869.

MISZTAL, I et al. (2013). "Methods to approximate reliabilities in single-step genomic evaluation". In: *Journal of Dairy Science* 96.1, pp. 647–654.

MOEINIZADE, Saba et al. (2019). "Optimizing Selection and Mating in Genomic Selection with a Look-Ahead Approach: An Operations Research Framework". In: *G3: Genes, Genomes, Genetics*, g3–200842.

MOMEN, Mehdi et al. (Aug. 2019). "Predicting Longitudinal Traits Derived from High-Throughput Phenomics in Contrasting Environments Using Genomic Legendre Polynomials and B-Splines". In: *G3: Genes, Genomes, Genetics* 9.10, pp. 3369–3380. DOI: 10.1534/g3.119.400346. URL: https://doi.org/10.1534/g3.119.400346.

MONIR, Md Mamun and Jun ZHU (2018). "Dominance and epistasis interactions revealed as important variants for leaf traits of maize NAM population". In: *Frontiers in plant science* 9, p. 627.

MONTESINOS-LÓPEZ, Osval A et al. (2015). "Threshold models for genome-enabled prediction of ordinal categorical traits in plant breeding". In: *G3: Genes, Genomes, Genetics* 5.2, pp. 291–300.

MONTESINOS-LÓPEZ, Osval A et al. (2019a). "A benchmarking between deep learning, support vector machine and Bayesian threshold best linear unbiased prediction for predicting ordinal traits in plant breeding". In: *G3: Genes, Genomes, Genetics* 9.2, pp. 601–618.

— (2019b). "New deep learning genomic-based prediction model for multiple traits with binary, ordinal, and continuous phenotypes". In: *G3: Genes, Genomes, Genetics* 9.5, pp. 1545–1556.

MORGANTE, Fabio et al. (2018). "Effect of genetic architecture on the prediction accuracy of quantitative traits in samples of unrelated individuals". In: *Heredity* 120.6, p. 500.

MOROTA, Gota and Daniel GIANOLA (2014). "Kernel-based whole-genome prediction of complex traits: a review". In: *Frontiers in genetics* 5, p. 363.

MOSER, Gerhard et al. (2009). "A comparison of five methods to predict genomic breeding values of dairy bulls from genome-wide SNP markers". In: *Genetics Selection Evolution* 41.1, p. 56.

MOUSSEAU, Timothy A and Derek A ROFF (1987). "Natural selection and the heritability of fitness components". In: *Heredity* 59.2, p. 181.

NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION. *NCBI Taxonomy*. Accessed: 2019-10-01. URL: `https://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/`.

NEYHART, Jeffrey, Aaron J LORENZ, and Kevin P SMITH (2019). "Multi-Trait Improvement by Predicting Genetic Correlations in Breeding Crosses". In: *bioRxiv*, p. 593210.

NGUYEN, Derrick and Bernard WIDROW (1990). "The truck backer-upper: An example of self-learning in neural networks". In: *Advanced neural computers*. Elsevier, pp. 11–19.

NORMAN, Adam et al. (2018). "Optimising Genomic Selection in Wheat: Effect of Marker Density, Population Size and Population Structure on Prediction Accuracy". In: *G3: Genes, Genomes, Genetics* 8.9, pp. 2889–2899. DOI: 10.1534/g3.118.200311. URL: https://doi.org/10.1534.

OAKEY, Helena et al. (2016). "Genomic selection in multi-environment crop trials". In: *G3: Genes, Genomes, Genetics* 6.5, pp. 1313–1326.

OGUTU, Joseph O, Hans-Peter PIEPHO, and Torben SCHULZ-STREECK (2011). "A comparison of random forests, boosting and support vector machines for genomic selection". In: *BMC proceedings*. Vol. 5. 3. BioMed Central, S11.

OHYAMA, Kanji et al. (Aug. 1986). "Chloroplast gene organization deduced from complete sequence of liverwort Marchantia polymorpha chloroplast DNA". In: *Nature* 322.6079, p. 572. ISSN: 1476-4687. DOI: 10.1038/322572a0. URL: https://www.nature.com/articles/322572a0 (visited on 05/20/2019).

OLEJNICZAK, Szymon Adam et al. (2016). "Chloroplasts: state of research and practical applications of plastome sequencing". In: *Planta* 244.3, pp. 517–527.

OLIPHANT, Travis E (2006). *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.

OVENDEN, Ben et al. (2018). "Accounting for Genotype-by-Environment Interactions and Residual Genetic Variation in Genomic Selection for Water-Soluble Carbohydrate Concentration in Wheat". In: *G3: Genes, Genomes, Genetics* 8.6, pp. 1909–1919. DOI: 10.1534/g3.118.200038. URL: https://doi.org/10.1534.

OWENS, Brenda F et al. (2014). "A foundation for provitamin A biofortification of maize: genome-wide association and genomic prediction models of carotenoid levels". In: *Genetics* 198.4, pp. 1699–1716.

OZKAN, Hakan, Avraham A LEVY, and Moshe FELDMAN (2001). "Allopolyploidy-induced rapid genome evolution in the wheat (Aegilops–Triticum) group". In: *The Plant Cell* 13.8, pp. 1735–1747.

PALMER, Jeffrey D. (1985). "COMPARATIVE ORGANIZATION OF CHLOROPLAST GENOMES". In: *Annual Review of Genetics* 19.1. PMID: 3936406, pp. 325–354. DOI: 10.1146/annurev.ge.19.120185.001545. eprint: https://doi.org/10.1146/annurev.ge.19.120185.001545. URL: https://doi.org/10.1146/annurev.ge.19.120185.001545.

PEIFFER, Jason A et al. (2014). "The genetic architecture of maize height". In: *Genetics* 196.4, pp. 1337–1356.

PIASKOWSKI, Julia et al. (2018). "Genomic heritability estimates in sweet cherry reveal non-additive genetic variance is relevant for industry-prioritized traits". In: *BMC genetics* 19.1, p. 23.

POLYAK, Boris T (1964). "Some methods of speeding up the convergence of iteration methods". In: *USSR Computational Mathematics and Mathematical Physics* 4.5, pp. 1–17.

POOK, Torsten et al. (2019). "Improving imputation quality in BEAGLE for crop and livestock data". In: *bioRxiv*, p. 577338.

POUDEL, Hari P. et al. (2019). "Genomic Prediction for Winter Survival of Lowland Switchgrass in the Northern USA". In: *G3: Genes, Genomes, Genetics*, g3.400094.2019. DOI: 10.1534/g3.119.400094. URL: https://doi.org/10.1534.

PURCELL, Shaun et al. (2007). "PLINK: a tool set for whole-genome association and population-based linkage analyses". In: *The American journal of human genetics* 81.3, pp. 559–575.

PURCELL, Shaun M et al. (2014). "A polygenic burden of rare disruptive mutations in schizophrenia". In: *Nature* 506.7487, p. 185.

QIAN, Lunwen, Wei QIAN, and Rod J SNOWDON (2014). "Sub-genomic selection patterns as a signature of breeding in the allopolyploid Brassica napus genome". In: *BMC genomics* 15.1, p. 1170.

QIU, Zhixu et al. (2016). "Application of machine learning-based classification to genomic selection and performance improvement". In: *International Conference on Intelligent Computing*. Springer, pp. 412–421.

R CORE TEAM (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

— (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

RAMPASEK, Ladislav and Anna GOLDENBERG (2016). "Tensorflow: Biology's gateway to deep learning?" In: *Cell systems* 2.1, pp. 12–14.

RAMSTEIN, Guillaume P. and Michael D. CASLER (2019). "Extensions of BLUP Models for Genomic Prediction in Heterogeneous Populations: Application in a Diverse Switchgrass Sample". In: *G3: Genes, Genomes, Genetics*, g3.200969.2018. DOI: 10.1534/g3.118.200969. URL: https://doi.org/10.1534.

RAMSTEIN, Guillaume P. et al. (2016). "Accuracy of Genomic Prediction in Switchgrass (Panicum virgatumL.) Improved by Accounting for Linkage Disequilibrium". In: *G3: Genes, Genomes, Genetics* 6.4, pp. 1049–1062. DOI: 10.1534/g3.115.024950. URL: https://doi.org/10.1534.

RATCLIFFE, Blaise et al. (2017). "Single-Step BLUP with Varying Genotyping Effort in Open-Pollinated Picea glauca". In: *G3: Genes, Genomes, Genetics* 7.3, pp. 935–942. DOI: 10.1534/g3.116.037895. URL: https://doi.org/10.1534.

RESENDE, M. F. R. et al. (2012). "Accuracy of Genomic Selection Methods in a Standard Data Set of Loblolly Pine (Pinus taeda L.)" In: *Genetics* 190.4, pp. 1503–1510. DOI: 10.1534/genetics.111.137026. URL: https://doi.org/10.1534.

RESENDE, Rafael Tassinari et al. (2019). "Enviromics in breeding: applications and perspectives on envirotypic-assisted selection". In: *BioRxiv*, p. 726513.

RHEE, Seung Yon et al. (2003). "The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community". In: *Nucleic acids research* 31.1, pp. 224–228.

RIEDELSHEIMER, Christian et al. (2013). "Genomic predictability of interconnected biparental maize populations". In: *Genetics* 194.2, pp. 493–503.

RINCENT, Renaud et al. (2012). "Maximizing the reliability of genomic selection by optimizing the calibration set of reference individuals: comparison of methods in two diverse groups of maize inbreds (Zea mays L.)" In: *Genetics* 192.2, pp. 715–728.

RINCENT, Renaud et al. (2018). "Phenomic Selection Is a Low-Cost and High-Throughput Method Based on Indirect Predictions: Proof of Concept on Wheat and Poplar". In: *G3: Genes, Genomes, Genetics*, g3.200760.2018. DOI: 10.1534/g3.118.200760. URL: https://doi.org/10.1534.

RITCHIE, Marylyn D and Kristel VAN STEEN (2018). "The search for gene-gene interactions in genome-wide association studies: challenges in abundance of

methods, practical considerations, and biological interpretation". In: *Annals of translational medicine* 6.8.

ROCAPS LAB. *CpBase*. Accessed: 2019-04-01, Version: 8/20/2017. URL: http://rocaplab.ocean.washington.edu/old_website/tools/cpbase.

RODRÍGUEZ-LEAL, Daniel et al. (2017). "Engineering quantitative trait variation for crop improvement by genome editing". In: *Cell* 171.2, pp. 470–480.

ROEBER, FK, GA GORDILLO, and HH GEIGER (2005). "In vivo haploid induction in maize. Performance of new inducers and significance of doubled haploid lines in hybrid breeding [Zea mays L.]" In: *Maydica (Italy)*.

ROORKIWAL, Manish et al. (2016). "Genome-enabled prediction models for yield related traits in chickpea". In: *Frontiers in plant science* 7, p. 1666.

ROSENBLATT, Frank (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY.

RUDER, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.

RUMELHART, David E, Geoffrey E HINTON, Ronald J WILLIAMS, et al. (1988). "Learning representations by back-propagating errors". In: *Cognitive modeling* 5.3, p. 1.

SANCHO, Rubén et al. (June 2018). "Comparative plastome genomics and phylogenomics of *Brachypodium* : flowering time signatures, introgression and recombination in recently diverged ecotypes". en. In: *New Phytologist* 218.4, pp. 1631–1644. ISSN: 0028646X. DOI: 10.1111/nph.14926. URL: http://doi.wiley.com/10.1111/nph.14926.

SANTOS DIAS, Luiz Antônio dos et al. (2004). "A priori choice of hybrid parents in plants". In: *Genet. Mol. Res* 3.3, pp. 356–368.

SCARCELLI, N. et al. (2016). "Intra-individual polymorphism in chloroplasts from NGS data: where does it come from and how to handle it?" In: *Molecular Ecology Resources* 16.2, pp. 434–445. DOI: 10.1111/1755-0998.12462. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1755-0998.12462. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1755-0998.12462.

SCHMIDHUBER, Jürgen (2015). "Deep learning in neural networks: An overview". In: *Neural networks* 61, pp. 85–117.

SCHOPP, Pascal et al. (2017a). "Accuracy of genomic prediction in synthetic populations depending on the number of parents, relatedness, and ancestral linkage disequilibrium". In: *Genetics* 205.1, pp. 441–454.

SCHOPP, Pascal et al. (2017b). "Genomic prediction within and across biparental families: means and variances of prediction accuracy and usefulness of deterministic equations". In: *G3: Genes, Genomes, Genetics* 7.11, pp. 3571–3586.

SCHRAG, Tobias A et al. (2018). "Beyond genomic prediction: combining different types of omics data can improve prediction of hybrid performance in maize". In: *Genetics* 208.4, pp. 1373–1385.

SEGURA, Vincent et al. (2012). "An efficient multi-locus mixed-model approach for genome-wide association studies in structured populations". In: *Nature genetics* 44.7, p. 825.

SEREN, Ümit et al. (2016). "AraPheno: a public database for Arabidopsis thaliana phenotypes". In: *Nucleic acids research*, gkw986.

SHAPIRO, Samuel Sanford and Martin B WILK (1965). "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3/4, pp. 591–611.

SHEN, Wei et al. (Oct. 2016). "SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation". In: *PLOS ONE* 11.10, pp. 1–10. DOI: 10.1371/journal.pone.0163962. URL: https://doi.org/10.1371/journal.pone.0163962.

SHEN, Xia et al. (2013). "A novel generalized ridge regression method for quantitative genetics". In: *Genetics* 193.4, pp. 1255–1268.

SHINOZAKI, K. et al. (1986). "The complete nucleotide sequence of the tobacco chloroplast genome: its gene organization and expression". In: *The EMBO Journal* 5.9, pp. 2043–2049. ISSN: 1460-2075. DOI: 10.1002/j.1460-2075.1986.tb04464.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1460-2075.1986.tb04464.x (visited on 05/20/2019).

SIVA, Nayanah (2008). *1000 Genomes project*.

SNOWDON, Rod J and Federico L INIGUEZ LUY (2012). "Potential to improve oilseed rape and canola breeding in the genomics era". In: *Plant breeding* 131.3, pp. 351–360.

SOPER, HE et al. (1917). "On the distribution of the correlation coefficient in small samples. Appendix II to the papers of" Student" and RA Fisher". In: *Biometrika* 11.4, pp. 328–413.

SOUSA, Massaine Bandeira e et al. (2017). "Genomic-enabled prediction in maize using kernel models with genotype× environment interaction". In: *G3: Genes, Genomes, Genetics* 7.6, pp. 1995–2014.

STEGEMANN, Sandra and Ralph BOCK (2009). "Exchange of genetic material between cells in plant tissue grafts". In: *science* 324.5927, pp. 649–651.

STEWART-BROWN, Benjamin B. et al. (2019). "Genomic Selection for Yield and Seed Composition Traits Within an Applied Soybean Breeding Program". In: *G3: Genes, Genomes, Genetics* 9.7, pp. 2253–2265. DOI: 10.1534/g3.118.200917. URL: https://doi.org/10.1534.

STOREY, John D. and Robert TIBSHIRANI (2003). "Statistical significance for genomewide studies". In: *Proceedings of the National Academy of Sciences* 100.16, pp. 9440–9445. ISSN: 0027-8424. DOI: 10.1073/pnas.1530509100. eprint: https://www.pnas.org/content/100/16/9440.full.pdf. URL: https://www.pnas.org/content/100/16/9440.

STRAUCH, Rene et al. (2015). "Discovery of a novel amino acid racemase through exploration of natural variation in Arabidopsis thaliana". In: *PNAS* 112. DOI: 10.1073/pnas.1503272112.

STRINGER, Sven et al. (2011). "Underestimated effect sizes in GWAS: fundamental limitations of single SNP analysis for dichotomous phenotypes". In: *PloS one* 6.11, e27964.

SUKUMARAN, Sivakumar et al. (2016). "Genomic Prediction with Pedigree and Genotype Environment Interaction in Spring Wheat Grown in South and West Asia, North Africa, and Mexico". In: *G3: Genes, Genomes, Genetics* 7.2, pp. 481–495. DOI: 10.1534/g3.116.036251. URL: https://doi.org/10.1534.

TECHNOW, Frank, Anna BÜRGER, and Albrecht E MELCHINGER (2013). "Genomic prediction of northern corn leaf blight resistance in maize with combined or

separated training sets for heterotic groups". In: *G3: Genes, Genomes, Genetics* 3.2, pp. 197–203.

TECHNOW, Frank et al. (2014). "Genome properties and prospects of genomic prediction of hybrid performance in a breeding program of maize". In: *Genetics* 197.4, pp. 1343–1355.

TETKO, Igor V, David J LIVINGSTONE, and Alexander I LUIK (1995). "Neural network studies. 1. Comparison of overfitting and overtraining". In: *Journal of chemical information and computer sciences* 35.5, pp. 826–833.

THAVAMANIKUMAR, Saravanan, Rudy DOLFERUS, and Bala R. THUMMA (2015). "Comparison of Genomic Selection Models to Predict Flowering Time and Spike Grain Number in Two Hexaploid Wheat Doubled Haploid Populations". In: *G3: Genes, Genomes, Genetics* 5.10, pp. 1991–1998. DOI: 10.1534/g3.115.019745. URL: https://doi.org/10.1534.

THORWARTH, Patrick, Eltohamy AA YOUSEF, and Karl J SCHMID (2018). "Genomic Prediction and Association Mapping of Curd-Related Traits in Gene Bank Accessions of Cauliflower". In: *G3: Genes, Genomes, Genetics* 8.2, pp. 707–718.

TIMPSON, Nicholas J et al. (2018). "Genetic architecture: the shape of the genetic contribution to human traits and disease". In: *Nature Reviews Genetics* 19.2, p. 110.

TOGNINALLI, Matteo et al. (2017). "The AraGWAS Catalog: a curated and standardized Arabidopsis thaliana GWAS catalog". In: *Nucleic acids research* 46.D1, pp. D1150–D1156.

TOGNINALLI, Matteo et al. (Oct. 2019). "AraPheno and the AraGWAS Catalog 2020: a major database update including RNA-Seq and knockout mutation data for Arabidopsis thaliana". In: *Nucleic Acids Research*. gkz925. ISSN: 0305-1048. DOI: 10.1093/nar/gkz925. eprint: http://oup.prod.sis.lan/nar/advance-article-pdf/doi/10.1093/nar/gkz925/30264131/gkz925.pdf. URL: https://doi.org/10.1093/nar/gkz925.

TSCHERMAK, Erich (1900). *Über künstliche Kreuzung bei Pisum sativum*. E. Tschermak.

TWYFORD, Alex D. and Rob W. NESS (Sept. 1, 2017). "Strategies for complete plastid genome sequencing". In: *Molecular Ecology Resources* 17.5, pp. 858–868. ISSN:

1755-0998. DOI: 10.1111/1755-0998.12626. URL: http://onlinelibrary.wiley.com/doi/10.1111/1755-0998.12626/abstract (visited on 01/26/2018).

UNTERSEER, Sandra et al. (2014). "A powerful tool for genome analysis in maize: development and evaluation of the high density 600 k SNP genotyping array". In: *BMC genomics* 15.1, p. 823.

VAN ROSSUM, Guido and Fred L DRAKE JR (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

VANRADEN, Paul M (2008). "Efficient methods to compute genomic predictions". In: *Journal of dairy science* 91.11, pp. 4414–4423.

VANRADEN, PM et al. (2008). "Reliability of genomic predictions for North American dairy bulls". In: *J. Dairy Sci* 91.Suppl 1, p. 305.

VERBYLA, Klara L et al. (2009). "Accuracy of genomic selection using stochastic search variable selection in Australian Holstein Friesian dairy cattle". In: *Genetics research* 91.5, pp. 307–311.

VIEIRA, IC et al. (2017). "Assessing non-additive effects in GBLUP model". In: *Genetics and molecular research: GMR* 16.2.

VINGA, Susana et al. (2012). "Pattern matching through Chaos Game Representation: bridging numerical and discrete data structures for biological sequence analysis". In: *Algorithms for molecular biology : AMB* 7.1. 22551152[pmid], pp. 10–10. ISSN: 1748-7188. DOI: 10.1186/1748-7188-7-10. URL: https://www.ncbi.nlm.nih.gov/pubmed/22551152.

WALLACE, Jason G, Eli RODGERS-MELNICK, and Edward S BUCKLER (2018). "On the road to breeding 4.0: unraveling the good, the bad, and the boring of crop quantitative genomics". In: *Annual review of genetics*.

WALSH, B and M LYNCH (2018a). "Short-term Changes in the Mean: 2. Truncation and Threshold Selection". In:

WALSH, Bruce and Michael LYNCH (2018b). *Evolution and selection of quantitative traits*. Oxford University Press.

WANG, Weiwen et al. (2018). "Assembly of chloroplast genomes with long- and short-read data: a comparison of approaches using Eucalyptus pauciflora as

a test case". In: *BMC genomics* 19.1. 30594129[pmid], pp. 977–977. ISSN: 1471-2164. DOI: 10.1186/s12864-018-5348-8. URL: https://www.ncbi.nlm.nih.gov/pubmed/30594129.

WANG, Yu et al. (2014). "The accuracy of prediction of genomic selection in elite hybrid rye populations surpasses the accuracy of marker-assisted selection and is equally augmented by multiple field evaluation locations and test years". In: *BMC genomics* 15.1, p. 556.

WARNER, Brad and Manavendra MISRA (1996). "Understanding neural networks as statistical tools". In: *The american statistician* 50.4, pp. 284–293.

WATSON, Amy et al. (2018). "Speed breeding is a powerful tool to accelerate crop research and breeding". In: *Nature plants* 4.1, p. 23.

WEBB, Sarah (2018). "Deep learning for biology". In: *Nature* 554.7693.

WERNER, Christian R et al. (2018). "Effective genomic selection in a narrow-genepool crop with low-density markers: Asian rapeseed as an example". In: *The plant genome* 11.2.

WICKE, Susann et al. (July 1, 2011). "The evolution of the plastid chromosome in land plants: gene content, gene order, gene function". In: *Plant Molecular Biology* 76.3, pp. 273–297. ISSN: 1573-5028. DOI: 10.1007/s11103-011-9762-4. URL: https://doi.org/10.1007/s11103-011-9762-4 (visited on 05/16/2019).

WINDHAUSEN, Vanessa S et al. (2012). "Effectiveness of genomic prediction of maize hybrid performance in different breeding populations and environments". In: *G3: Genes, Genomes, Genetics* 2.11, pp. 1427–1436.

WRIGHT, Sewall (1922). "Coefficients of inbreeding and relationship". In: *The American Naturalist* 56.645, pp. 330–338.

WÜRSCHUM, Tobias (2012). "Mapping QTL for agronomic traits in breeding populations". In: *Theoretical and Applied Genetics* 125.2, pp. 201–210.

WÜRSCHUM, Tobias, Stefan ABEL, and Yusheng ZHAO (2014). "Potential of genomic selection in rapeseed (B rassica napus L.) breeding". In: *Plant Breeding* 133.1, pp. 45–51.

WÜRSCHUM, Tobias et al. (2013). "Genomic selection in sugar beet breeding populations". In: *BMC genetics* 14.1, p. 85.

XAVIER, Alencar, William M. MUIR, and Katy Martin RAINEY (2016). "Assessing Predictive Properties of Genome-Wide Selection in Soybeans". In: *G3* 6.8, pp. 2611–2616. DOI: 10.1534/g3.116.032268. URL: https://doi.org/10.1534.

XIAO-MING, Zheng et al. (May 8, 2017). "Inferring the evolutionary mechanism of the chloroplast genome size by comparing whole-chloroplast genome sequences in seed plants". In: *Scientific Reports* 7.1, p. 1555. ISSN: 2045-2322. DOI: 10.1038/s41598-017-01518-5. URL: https://www.nature.com/articles/s41598-017-01518-5 (visited on 05/16/2019).

XU, S (2010). "An expectation–maximization algorithm for the Lasso estimation of quantitative trait locus effects". In: *Heredity* 105.5, p. 483.

XU, Shizhong (Aug. 2013). "Genetic Mapping and Genomic Selection Using Recombination Breakpoint Data". In: *Genetics* 195.3, pp. 1103–1115. DOI: 10.1534/genetics.113.155309. URL: https://doi.org/10.1534/genetics.113.155309.

YANG, Jian et al. (2010). "Common SNPs explain a large proportion of the heritability for human height". In: *Nature genetics* 42.7, p. 565.

YAP, Chloe X et al. (2018). "Misestimation of heritability and prediction accuracy of male-pattern baldness". In: *Nature communications* 9.1, p. 2537.

ZAPATA-VALENZUELA, Jaime et al. (2013). "Genomic Estimated Breeding Values Using Genomic Relationship Matrices in a Cloned Population of Loblolly Pine". In: *G3: Genes, Genomes, Genetics* 3.5, pp. 909–916. DOI: 10.1534/g3.113.005975. URL: https://doi.org/10.1534.

ZEILER, Matthew D (2012). "ADADELTA: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701*.

ZHANG, Haohao et al. (2019). "Factors affecting the accuracy of genomic selection for agricultural economic traits in maize, cattle, and pig populations". In: *Frontiers in genetics* 10, p. 189.

ZHANG, Wenyu et al. (2011). "A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies". In: *PloS one* 6.3, e17915.

ZHANG, Yan et al. (2018). "Estimation of complex effect-size distributions using summary-level statistics from genome-wide association studies across 32 complex traits". In: *Nature genetics* 50.9, p. 1318.

ZHANG, Zhiwu et al. (Mar. 2010). "Mixed linear model approach adapted for genome-wide association studies". In: *Nature Genetics* 42.4, pp. 355–360. DOI: 10.1038/ng.546. URL: https://doi.org/10.1038/ng.546.

ZHENG, Xiuwen (2013). "A Tutorial for the R Package SNPRelate". In: *University of Washington, Washington, USA*.

ZHONG, Shengqiang et al. (2009). "Factors affecting accuracy from genomic selection in populations derived from multiple inbred lines: a barley case study". In: *Genetics* 182.1, pp. 355–364.

ZHOU, Xiang and Matthew STEPHENS (June 2012). "Genome-wide efficient mixed-model analysis for association studies". In: *Nature Genetics* 44.7, pp. 821–824. DOI: 10.1038/ng.2310. URL: https://doi.org/10.1038/ng.2310.

ŽILINSKAS, A (2006). *Practical mathematical optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms*.

Curriculum Vitae
Jan Alexander Freudenthal

25. June 1988	Born in Lübeck, Germany
2004 - 2006	International Baccalaureate, Princess Anne High School, VA, USA
2010 - 2014	Bachelor of Science in Agricultural Sciences, CAU Kiel
2014 - 2016	Master of Science in Agricultural Sciences, GAU Göttingen
2016 - 2019	Ph.D student at the GSLS, JMU Würzburg.