JULIUS-MAXIMILIANS UNIVERSIÄT WÜRZBURG

DOCTORAL THESIS

Quantitative genetics - from genome assemlbies to neural network aided omics based prediction of quantitative traits

Author:
Jan Alexander FREUDENTHAL

Supervisor:
Prof. Arthur KORTE

A thesis submitted in fulfillment of the requirements *for the degree of Ph.D.*

in the

Research group for evolutionary genomics

GSLS

October 21, 2019

Declaration of Authorship

I, Jan Alexander FREUDENTHAL, declare that this thesis titled, “Quantitative genetics - from genome assemlbies to neural network aided omics based prediction of quantitative traits” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

____ Date:

____

"Wit beyond messure is men's greatest treassure"

Rawenclaw

JULIUS-MAXIMILIANS UNIVERSIÄT WÜRZBURG

Abstract

Faculty Name

GSLS

Ph.D.

Quantitative genetics - from genome assemlbies to neural network aided omics based prediction of quantitative traits

by Jan Alexander FREUDENTHAL

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

The acknowledgments and the people to thank go here, don’t forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Abbreviations

ANN	**A**rtificial **N**eural **N**etwork
BLUP	**B**est **L**inear **U**nbiased **P**redictor
BLUE	**B**est **L**inear **U**nbiased **E**stimator
CPU	**C**ore **P**rocessing **U**nit
EMMA	**E**fficient **M**ixed **M**odel **A**ssociations
FCL	**F**ully **C**onnected **L**ayer
GP	**G**enomic **P**redicton
GPU	**G**raphical **P**rocessing **U**nit
GS	**G**enomic **S**election
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
LCL	**L**ocally **C**onnected **L**ayer
LD	**L**inkage **D**isequilibrium
LMM	**L**inear **M**ixed **M**odel
MLP	**M**ulti **L**ayer **P**erceptron
ML	**M**achine **L**earning
QTL	**Q**uantitative **T**rait **L**ocus
RKHS	**R**erproducing **K**ernel **H**ilbert **S**paces
RSS	**R**esidual **S**um of **S**quares
SNP	**S**ingle **N**ucleotide **P**olymorphism
TRN	**TR**ai**N**ing subset
TST	**T**e**ST**ing subset
WGS	**W**hole **G**enome **S**equencing
LSC	**L**arge **S**ingle **C**opy
SSC	**S**mall **S**ingle **C**opy
IR	**I**nverted **R**epeat
DNA	**D**eoxyribo**N**ucleic **A**cid
DNA	**R**ibo**N**ucleic **A**cid
GUI	**G**raphical **U**ser **I**nterface
BP	**B**ase **Pair**
DH	**D**oubled **H**aploid
GBLUP	**G**enomic **B**est **L**inear **U**nbiased **P**redictor
XOR	e**X**clusive **OR**
ReLU	**Re**ctified **L**inear **U**nits
MSE	**M**ean **S**quare **E**rror
RMSE	**R**oot **M**ean **S**quare **E**rror

For/Dedicated to/To my…

1 Benchmarking of Chloroplast Genome Assembly tools

1.1 Introduction

Here I will but the introduction to from the paper

1.2 Material and Methods

1.2.1 Methods

1.2.2 Tools

1.2.3 Evaluation

Quantitative

$$score = \frac{1}{4} \cdot \left(cov_{ref} + cov_{qry} + min \left\{ \frac{cov_{qry}}{cov_{ref}}, \frac{cov_{ref}}{cov_{qry}} \right\} + \frac{1}{n_{contigs}} \right) \cdot 100 \tag{1.1}$$

Qualitative

Consisteny

1.2.4 Data

Simulated

Real data set

Novel data set

1.3 Results

1.3.1 Qualitative

1.3.2 Quantitatve

Simulated data

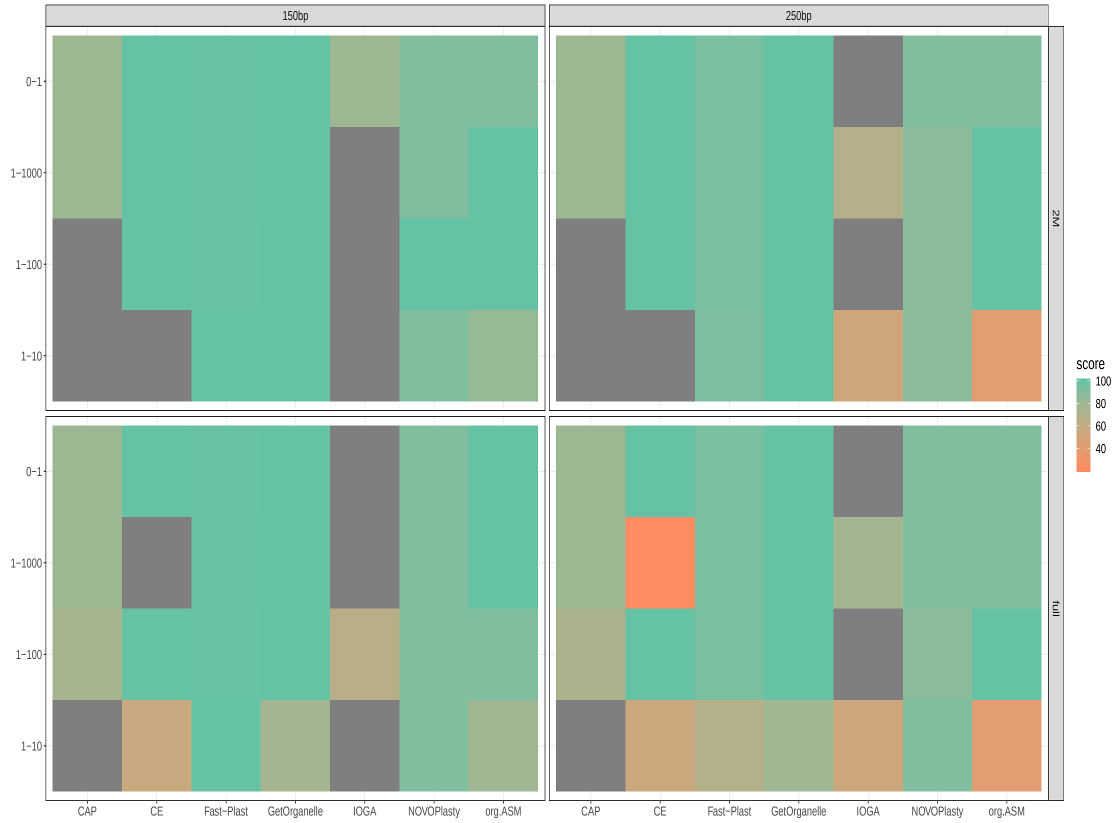

FIGURE 1.1: Results of assemblies exectuted with simulated data sets.

Real data sets

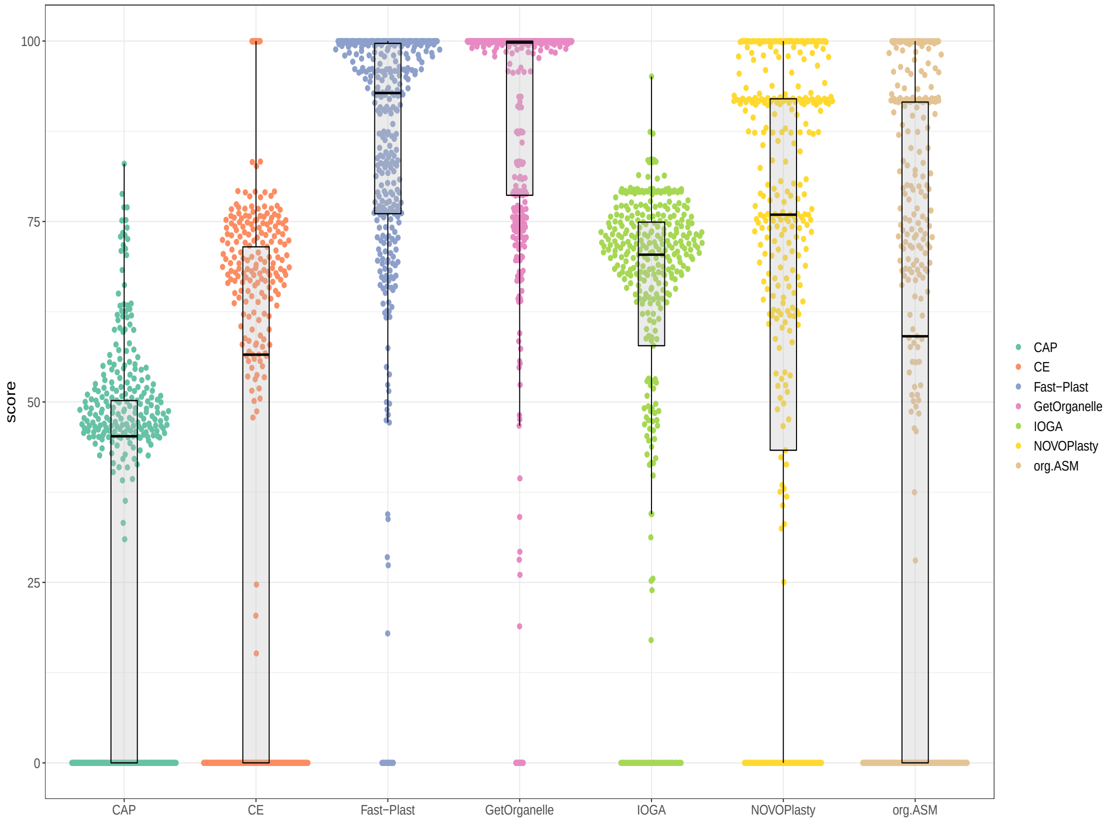

FIGURE 1.2: Box and swarm plots depict the resluts from the scoring shown in 1.1

Consistency

Real data sets

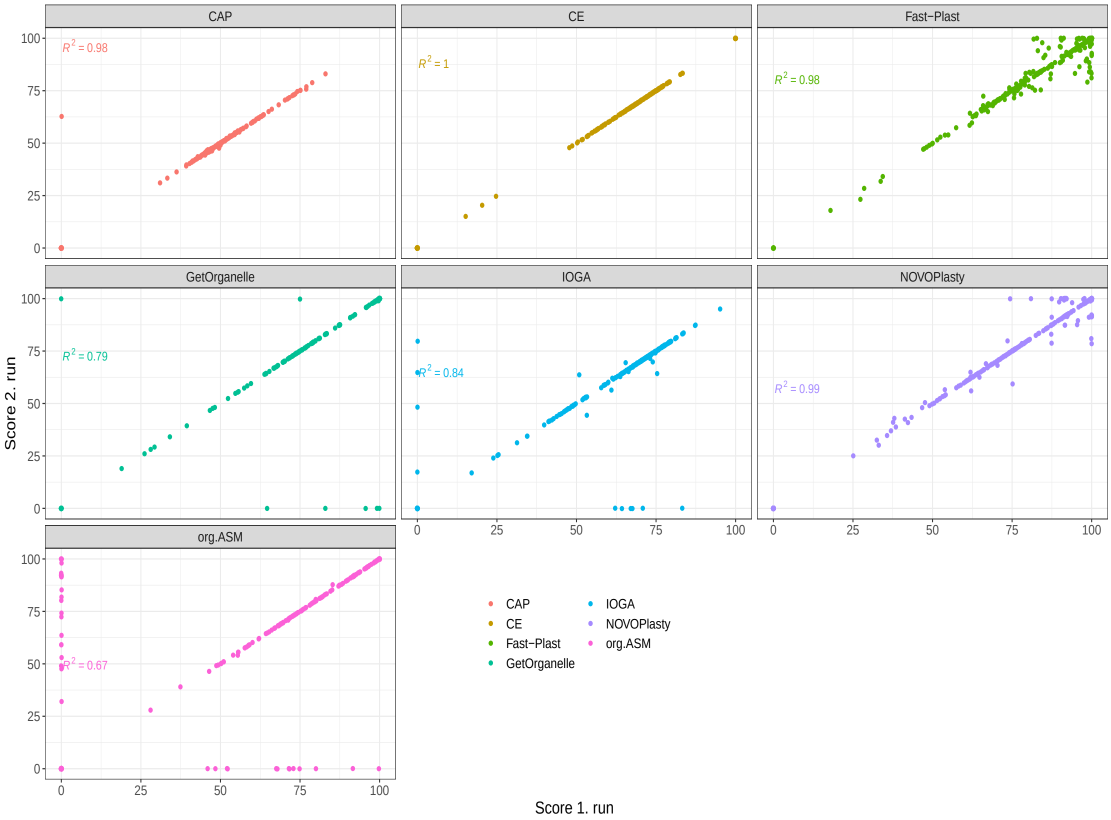

FIGURE 1.3: Swarm plots depict the resluts from the scoring shown in **??**

TABLE 1.1: The effects of treatments X and Y on the four groups studied.

Groups	Treatment X	Treatment Y
1	0.2	0.8
2	0.17	0.7
3	0.24	0.75
4	0.68	0.3

Novel

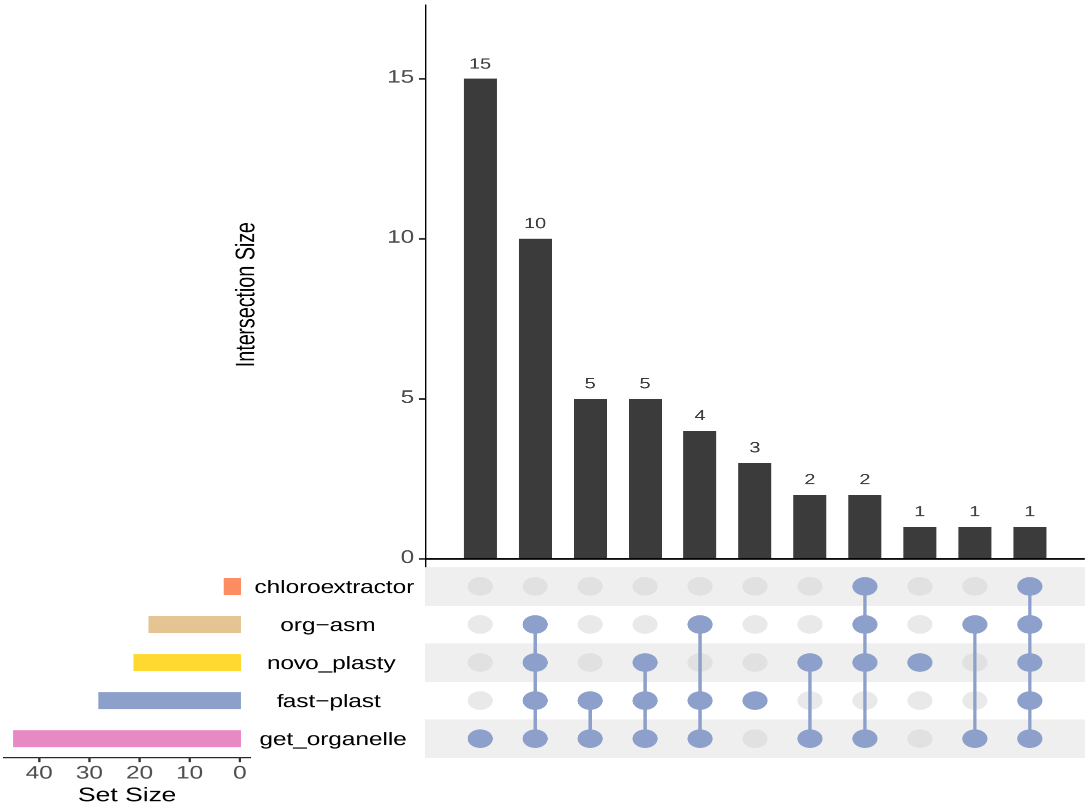

FIGURE 1.4: nls

1.4 Disucssion

FIGURE 1.5: Upset plot showing the intersections of sucess rates between assemblers. A successful assembly was defined with a score > 99 according to equation 1.1

2 Understanding the hapoltype structure of Arabidopisis thaliana

2.1 Introduction

2.2 Haplotyping of A. thaliana

2.3 Results

2.4 Disucssion

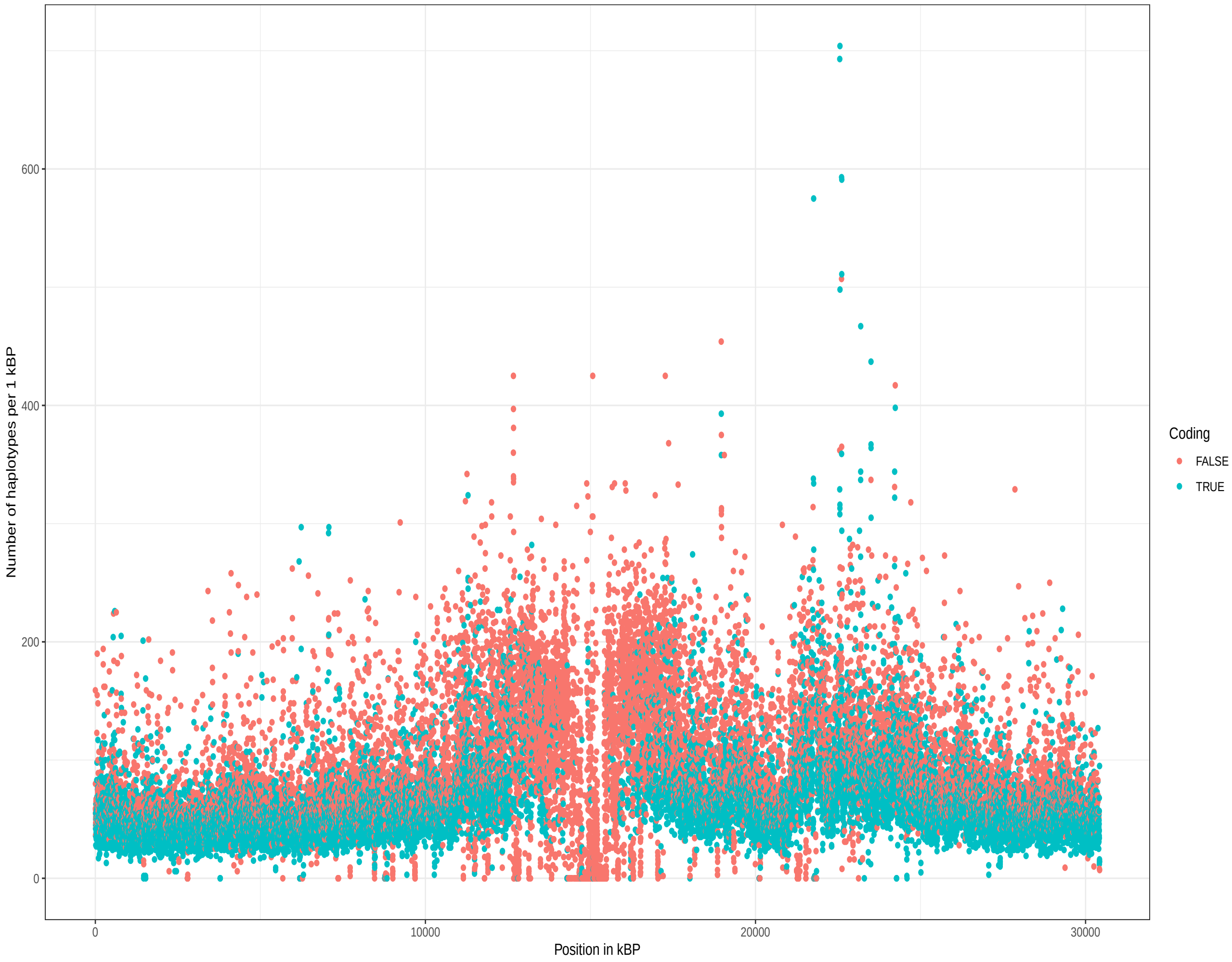

FIGURE 2.1: The number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

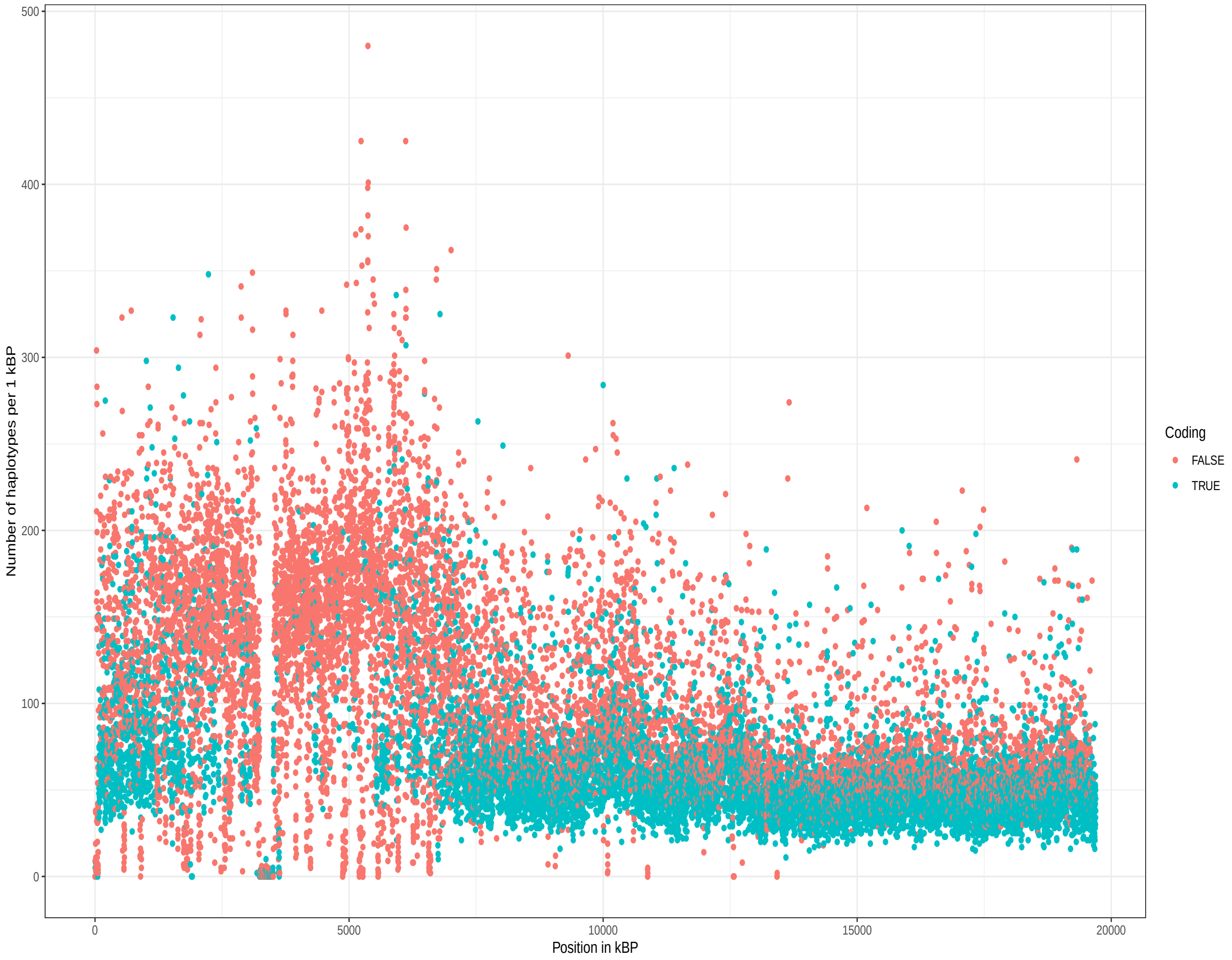

FIGURE 2.2: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

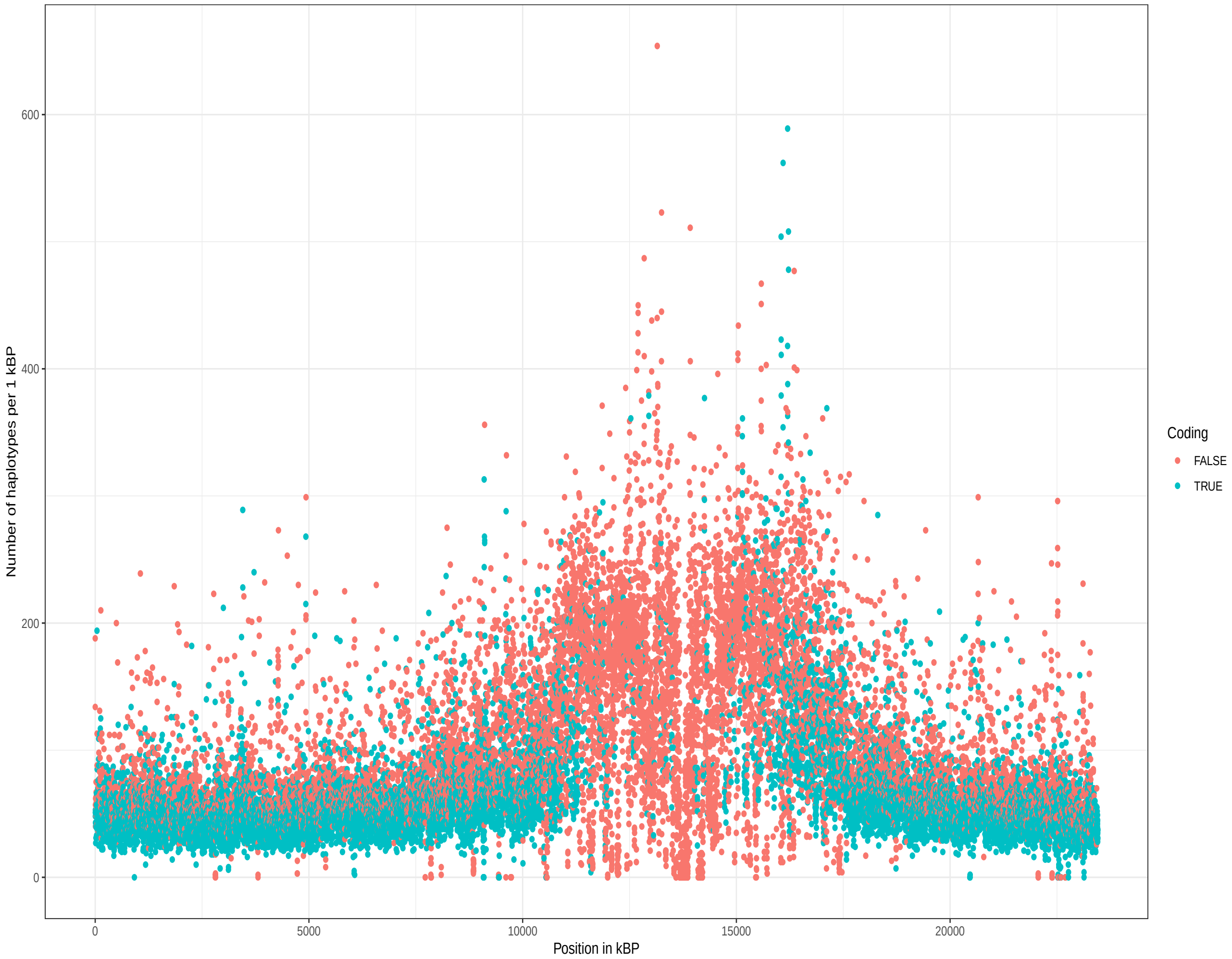

FIGURE 2.3: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

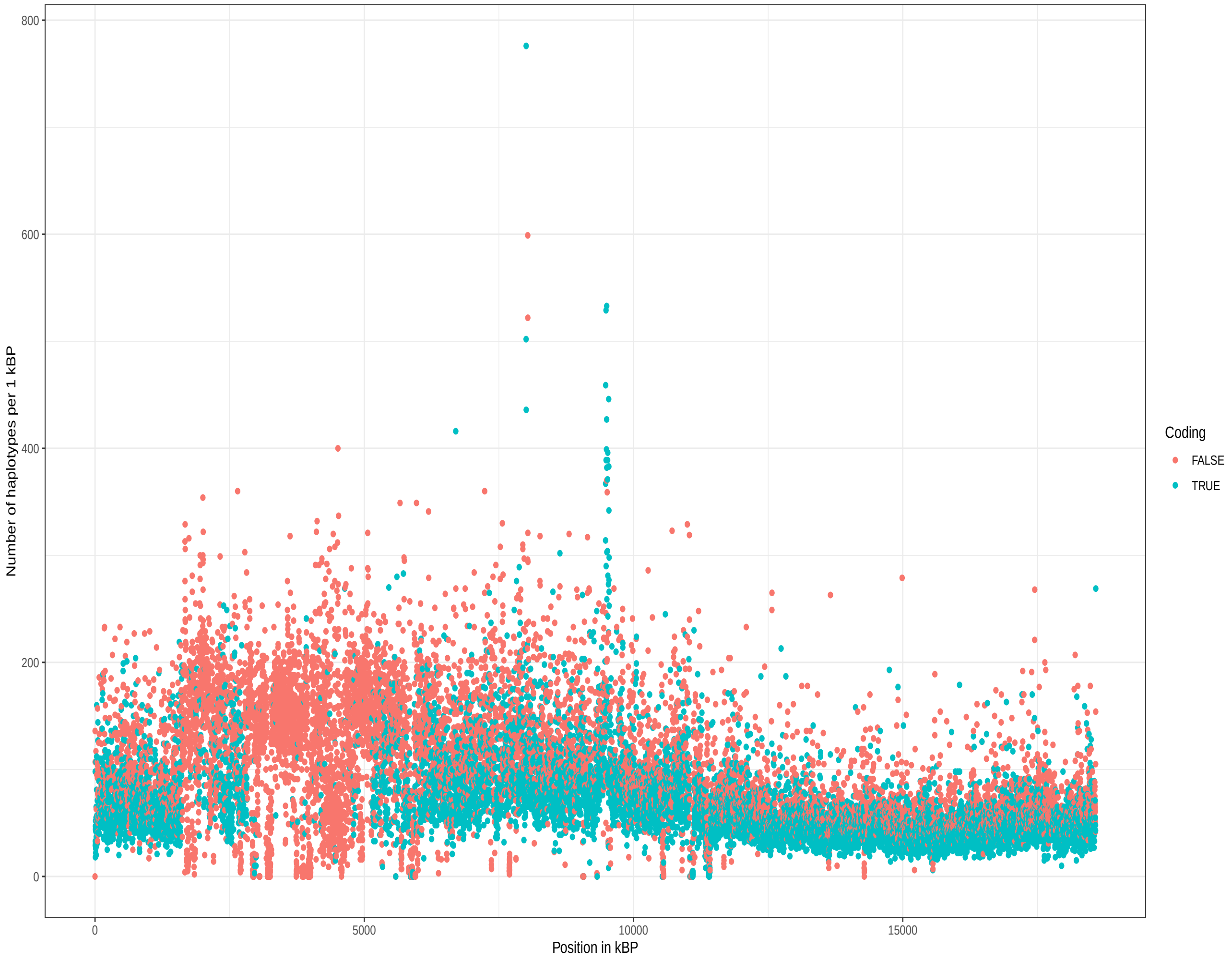

FIGURE 2.4: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

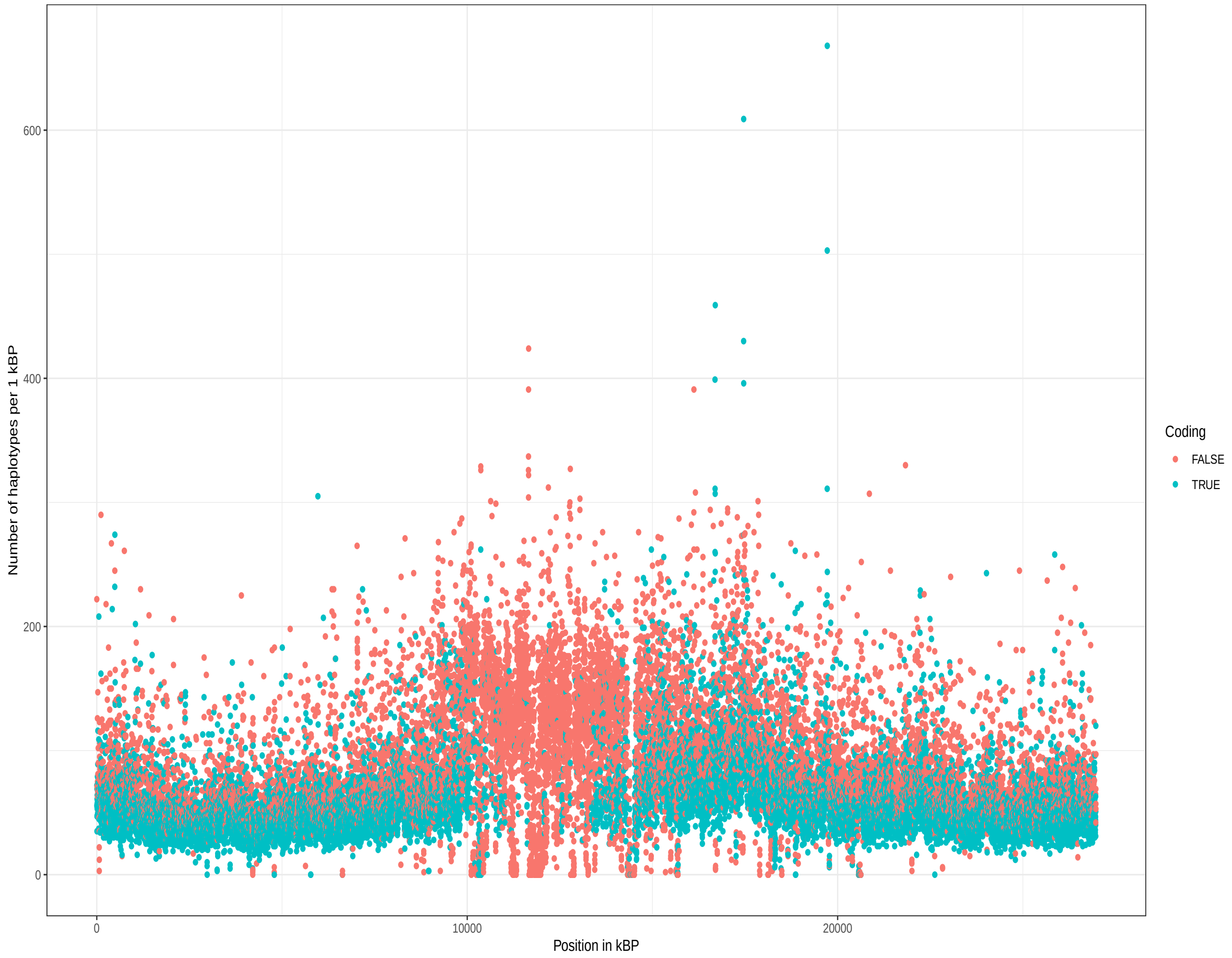

FIGURE 2.5: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

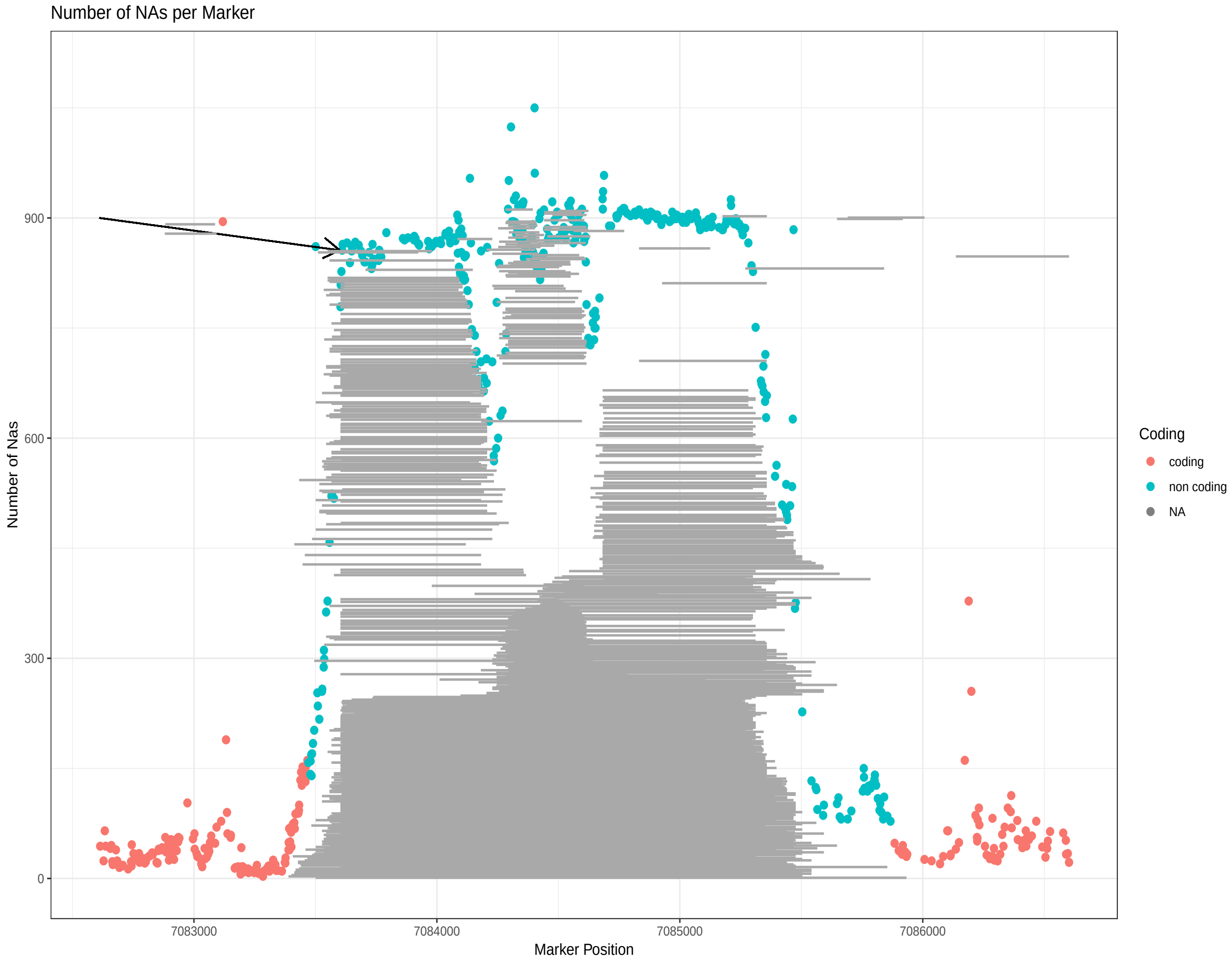

FIGURE 2.6: Number of segregating haplotypes with a polymorphism in at least one position over a stretch of 1 kBP.

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

The following chapter is written by the author and has been published on bioχiv preprint server Freudenthal et al., 2019

3.1 Introduction

Genome-wide association studies, pioneered in human genetics Hirschhorn and Daly, 2005 in the last decade, have become the predominant method to detect associations between phenotypes and the genetic variations present in a population. 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 cost 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 density 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 *Arabidopsis 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 use significance thresholds that are based on the false-discovery rate (FDR) Storey and Tibshirani, 2003. An alternative approach are permutation-based thresholds Che et al., 2014. Permutation-based thresholds estimate th e 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

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.

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 Abadi et al., 2015, a multi-language machine learning framework published and developed by Google. 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 slightly altered function based on the Python package *limix* Lippert et al., 2014. The GWAS model is based on the following linear mixed model 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 calcucated as descirebed 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 that 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 report summary statistics, such as the estimated effect size (β_i) and its standard error for each marker.

Calculation of permutation-based thresholds for GWAS

To calculate a permuation-based threshold, we essentially perform n repetitions ($n > 100$) of the GWAS on the same data with the sole difference that before each GWAS we randomize the phenotypic values. 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 we take the lowest p-value (after filtering for the desired minor allele count) for each permutation and take the 5% lowest value as the permutation-based threshold for the GWAS.

Benchmarking

For benchmarking of `GWAS-Flow` we used data from the *Arabidopsis* 1001 Genomes Project Alonso-Blanco et al., 2016. The genomic data we used were subsets between 10,000 and 100,000 markers. We chose not to include subsets that exceed 100,000 markers, because there is a linear relationship between the number of markers and the computational time demanded, as all markers are tested independently. We used phenotypic data for flowering time at ten degrees (FT10) for *A. thaliana*, published and downloaded from the AraPheno database Seren et al., 2016. We down- and up-sampled sets to generate phenotypes for sets between 100 and 5000 accessions. For each set of phenotypes and markers we ran 10 permutations to assess the computational time needed. 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, we compared the times required by *GEMMA* and both `GWAS-Flow` implementations for $>$ 200 different real datasets from *A. thaliana* that have been downloaded from the AraPheno Seren et al., 2016 database and have been analyzed with the available fully imputed genomic dataset 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 1A shows the time demand as a function of the number of accessions used in the analysis with 10,000 markers. The quadratic increase in time demand is clearly visible for the custom R implementation, as well as for the CPU-based `GWAS-Flow` implementation and *GEMMA*. The `GWAS-Flow` implementation and *GEMMA* clearly outperforms the R implementation in general, while for a small 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 1B shows the computational time in relation to the number of markers and a fixed amount 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, we also run both implementations on more than 200 different real datasets downloaded from the AraPheno database. Figure 1C shows the computational time demands for all analyses comparing both `GWAS-Flow` implementation 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.

3.4 Disucssion

We made use of recent developments of computational architecture and software to cope with the increasing computational demand in analyzing large GWAS datasets. With `GWAS-Flow` we implemented both, a CPU- and a GPU-based version of the classical linear mixed model commonly used for GWAS. Both implementations outperform custom R scripts on simulated and real data. While the CPU-based version performs nearly identical compared to *GEMMA*, a commonly used GWAS implementation, 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

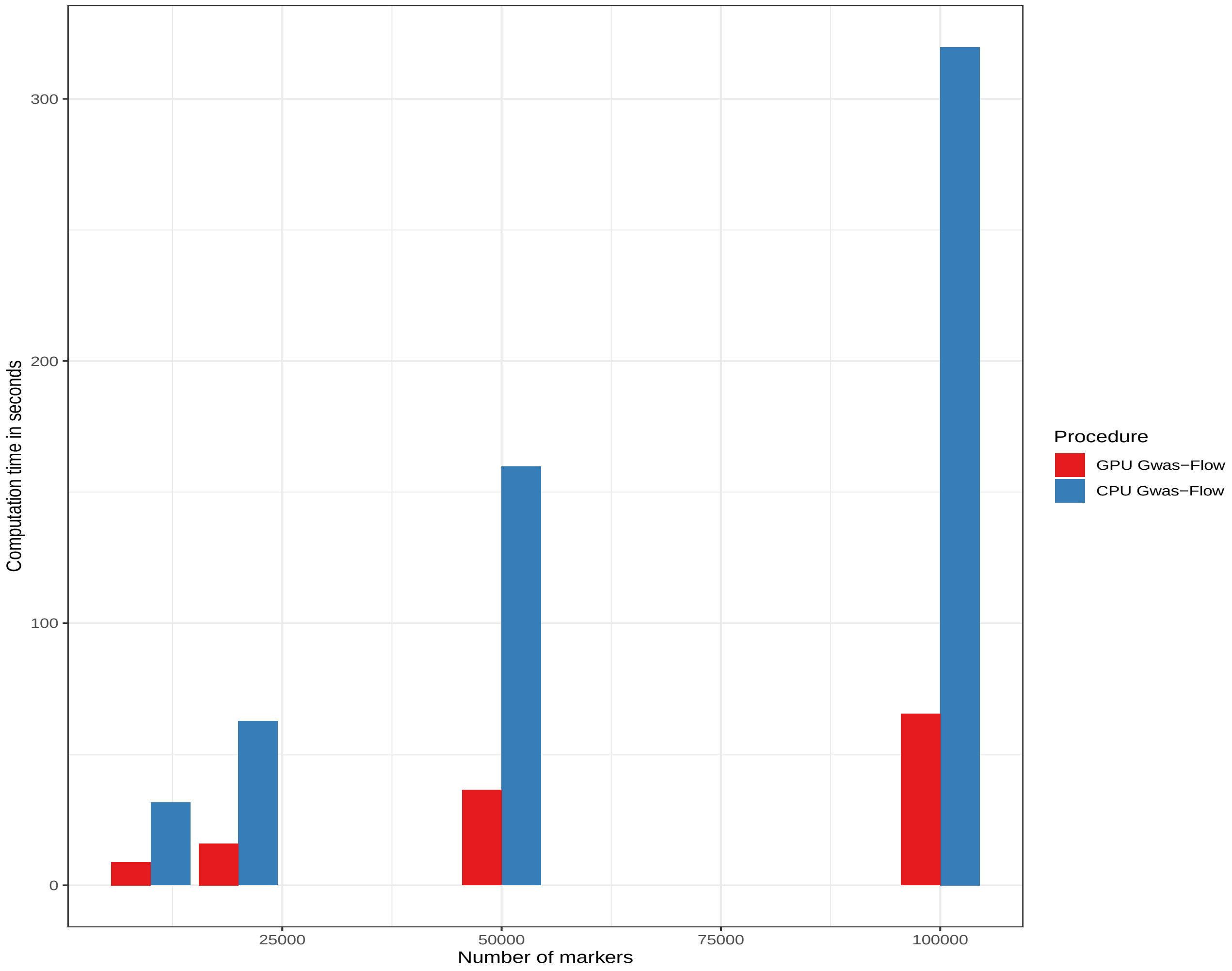

FIGURE 3.1: Computational time as a function of the number of genetic markers with constantly 2000 accessions for both `GWAS-Flow` versions

marker are 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 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

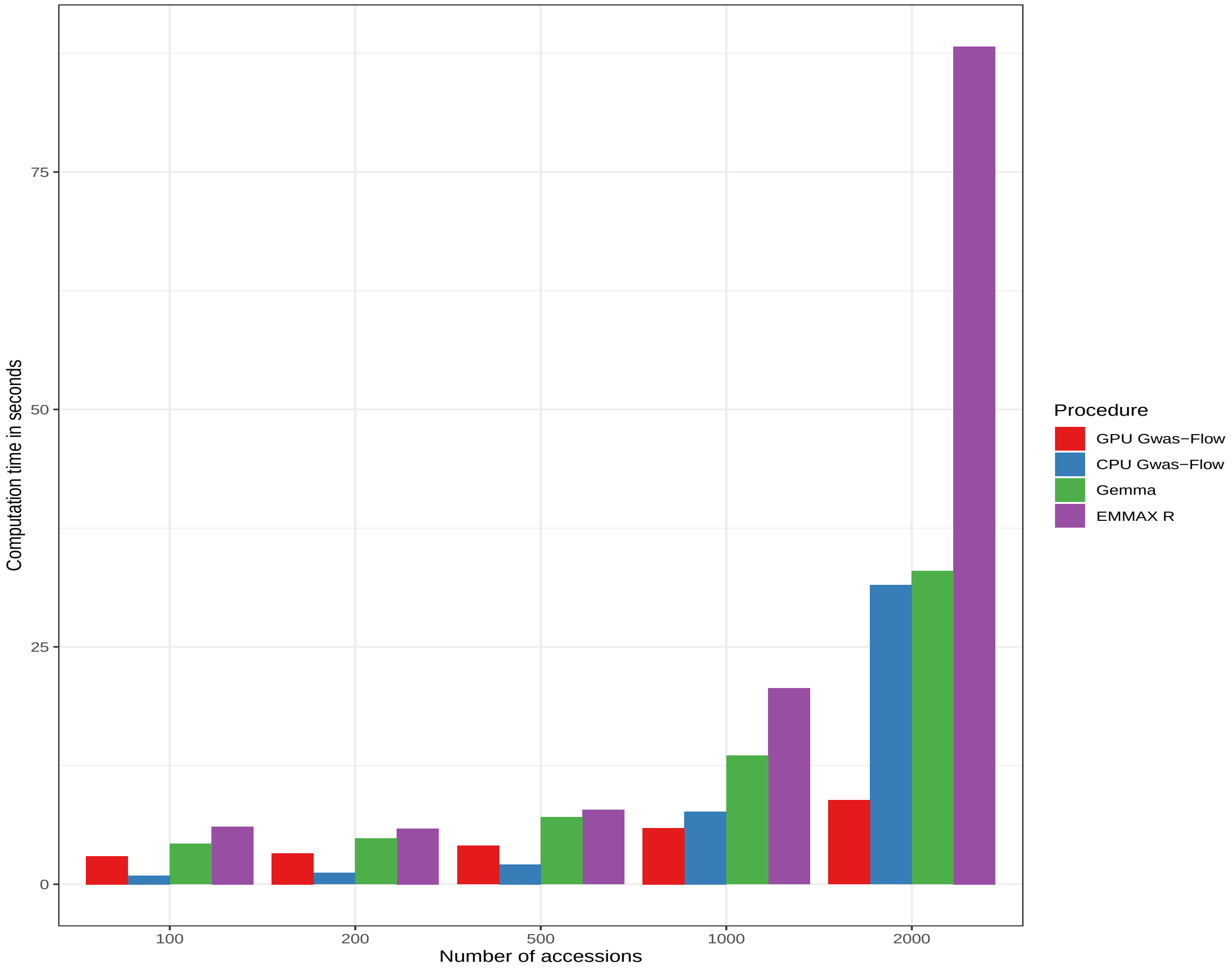

FIGURE 3.2: Computational time as a function of the number of accessions with 10000 markers each.

count. This could help to distinguish false and true associations even for rare alleles. `GWAS-Flow` is a versatile and fast software package. Currently `GWAS-Flow` is and will remain under active development to make the software more versatile. This will e.g. include the compatibility with TensorFlow v2.0.0 and enable 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 might be a statistic nightmare, it now would be computationally feasible.

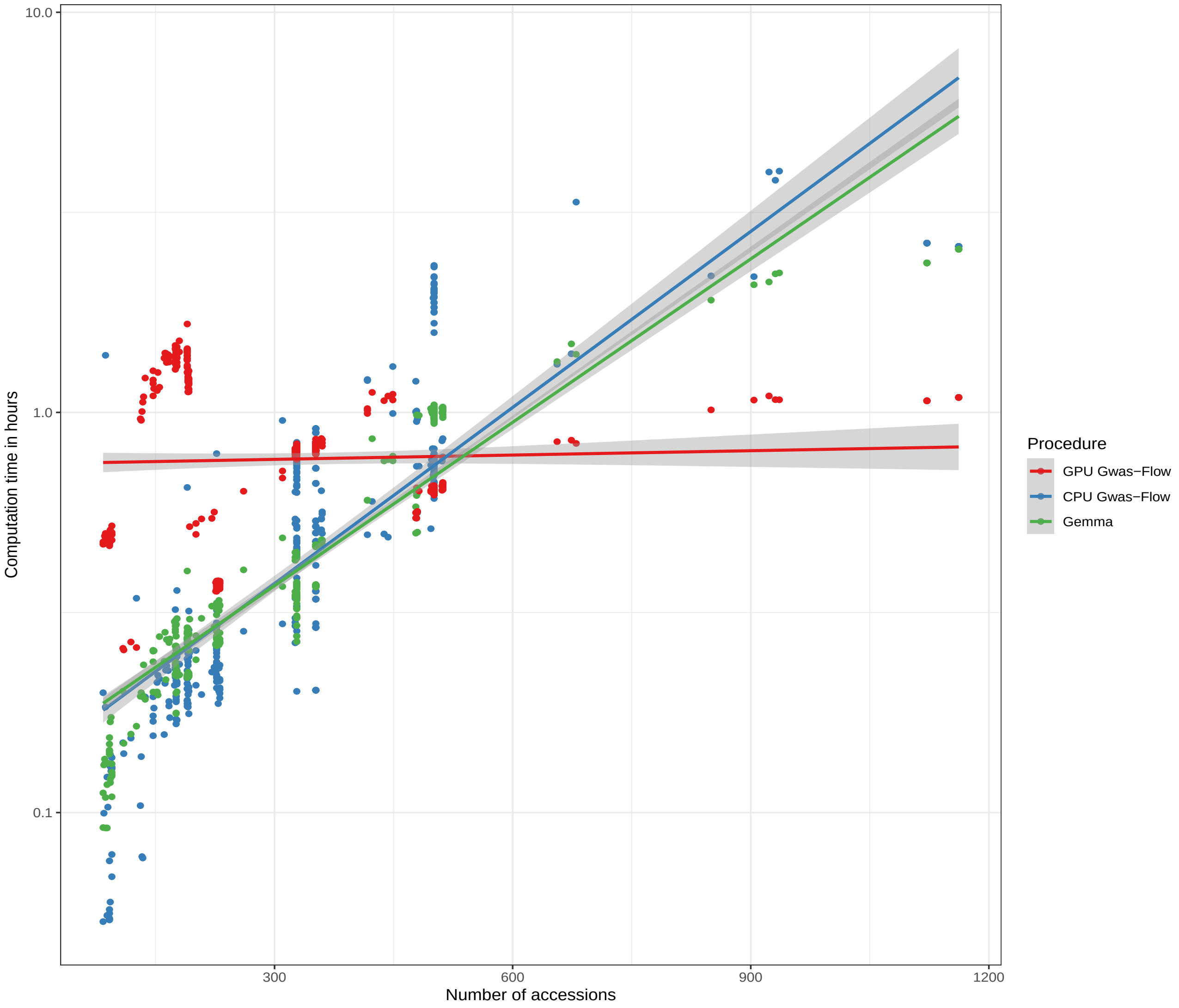

FIGURE 3.3: Comparison of the computational time for the analyses of >200 phenotypes from *Arabidopsis thaliana* as a function of the number of accessions for *GEMMA* and the CPU- and GPU-based version of `GWAS-Flow`. GWAS was performed with a fully imputed genotype matrix containing 10.7 M markers and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

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

4.1 Introduction

4.1.1 A brief history of machine learning

Basic perceptron model

While machine learning, neural networks, deep learning became essential tools for many applications in more recent years, their mathematical principals date back to the early 1950s and 1960s. Figure 4.1 schematically show the basic perceptron model as proposed by Rosenblatt, which was designed to mimic the information flow in biological nervous systems Rosenblatt, 1961

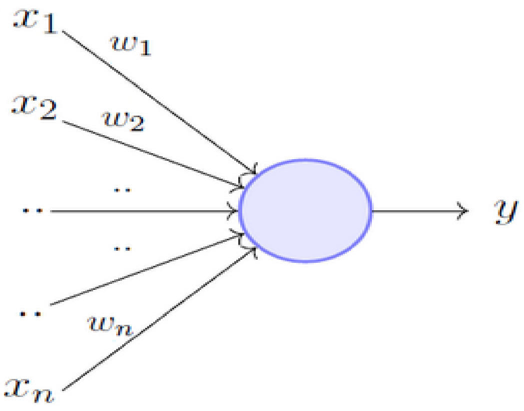

FIGURE 4.1: Basic perceptron model as proposed by Rosenblatt

This basic perceptron, which contrary to perceptrons used nowadays does not have an activation function, takes n binary inputs $x_1, x_2....x_n$ and produces a single, likewise binary, output y after being processed by the perceptron or neuron. To achieve this Rosenblatt introduced the concept of weights which indicated a certain 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 and biases $\sum_i w_i x_i$. If a certain threshold value is met the neuron is either activated and outputs 1 or not and outputs 0. This is algebraically represented in 4.1

$$0 = \text{if} \sum_i^n w_j x_i - \theta \leq 0 \tag{4.1a}$$

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

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 in per definition is negative. A single perceptron is a linear classifier and can only be trained on linearly separable functions and can used as shown by Rosenblatt, 1961 to solve simple logical operations as AND, OR and not. The simple perceptron fails, due to non-linearity, to perform XOR operations as shown by Marvin and Seymour, 1969. This discovery let to a near still stance in the research of artificial neural networks in the 1970s. This time period as now often referred to as the first AI-winter. Another reason that massively hindered the applications and research of machine learning during that time, 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. Continuing the trend of mimicking human neural networks, multiple artificial neurons are stacked into layers and these layers, are connected to each other allowing communication between the many perceptrons in a such generated network. Figure 4.2 shows schematically the basic structure of such a network, now container three types of layers. (i) the input layer, (ii) one or more hidden layers and (iii) one output layer, which in this case only consists of one neuron.

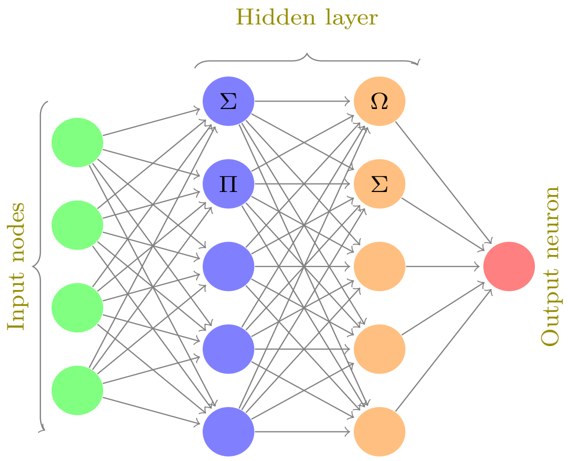

FIGURE 4.2: Schematic layout of a simple multi-layer perceptron

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

Activation functions

The neurons discussed so far are only capable of outputting binary results. Either 0 or 1, depending on the threshold values being met 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 can not be met with binary outputs. Activation functions for a given node provides rules for the output in accordance to the inputs Žilinskas, 2006.

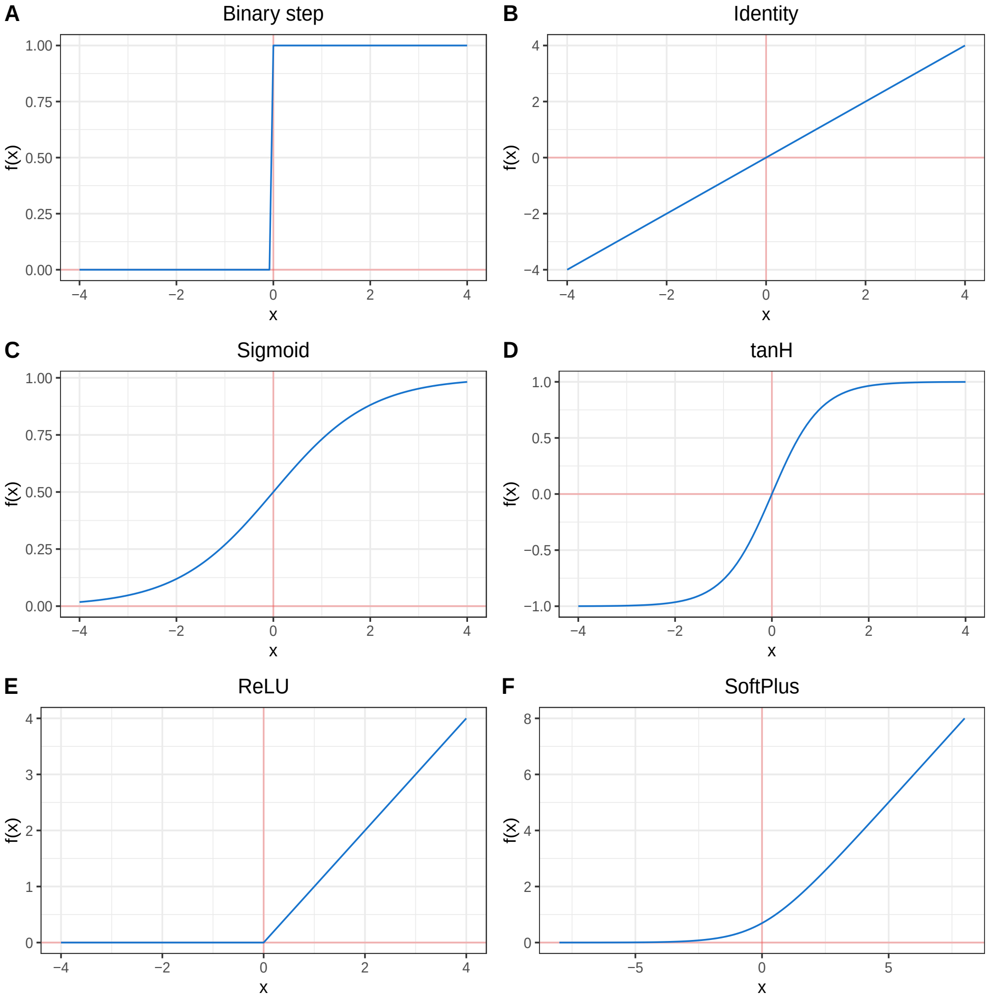

FIGURE 4.3: Popular activation functions used in neural networks. **A** Binary step activation function. **B** Identity activation function. **C** Sigmoid or logistic activation function. **D** tangens hyperbolicus activation function. **E** rectified linear units activation function . **F** SoftPlus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

Figure 4.3 **A** shows six of the most commonly used activation functions Warner and Misra, 1996. The simplest one was introduced, is the binary step activation function equation 4.2, which properties have been discussed along the perceptron model. All other activation produce continuous outputs from any given input. Basically any mathematical function can serve as an activation function in neural nets, starting with a simple identity function 4.3 , 4.3 **B**. Sigmoid figure 4.3 **C**, equation 4.4 and tanh figure 4.3 **D**, equation 4.5, when $x \to \infty$ or $x \to -\infty$ they have similar properties to the binary function, but produce continuous output around 0.

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

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

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

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

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

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

ReLU (equation 4.6) and the 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 2000s Hahnloser et al., 2000 and have since then overtaken all others as the most popular activations functions in neural networks and deep learning today LeCun, Bengio, and Hinton, 2015 and they have proven to be superior in deep-learning algorithms that 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, compared to tanh and sigmoid activation Glorot, Bordes, and Bengio, 2011. All activation functions shown in figure **??**, but the binary step function, share one common property: a small change of the input weight will result in small changes in 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. This property is, as described below, is an important prerequisite for networks being able to learn.

Gradient descent algorithm

Let the network shown in 4.2 be for the classification of a arbitrary phenotype like blue petals with $x_1 ... x_4$ on the input layers being genetic markers as features. And the output layer displaying a value from 0 to 1, meaning yes: blue petals from 0 - 0.5 and no blue petals from 0.5 to 1. To quantify how well the network performs on achieving that goal 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), cross-entropy and many others. In general MSE, MSE are commonly 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 problems Janocha and Czarnecki, 2017. Since all problems presented in due course or regression problems, that use MSE as their loss function, this will be the only one emphasized.

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

Equation 4.8 shows the MSE function which is the sum of the squares of the differences of all the predicted and the real values. The same function can be rewritten with the previously used terminology of weights and biases in equation 4.8.

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

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 the $L(w, b)$ will always be positive. And if $L(w, b) \to 0$ the loss is minimal, meaning that the real and predicted values are close together and the network found weights and biases that explain the output well.

A widely used function to find the optimum for such a loss function is gradient descent. Its objective is to fine the minimum for the loss function Bottou, 1991. The behind gradient descent or other optimizing algorithms is 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 as shown in equation 4.10, making use of partial derivatives.

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

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 ∇C the vector of the partial derivatives: equation 4.11

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

Having defined ∇C and Δd the term 4.10 can be simplified as equation 4.12

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

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

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

Stochastic gradient descent

Backpropagation

Backpropagation Rumelhart, Hinton, and Williams, 1988

Regularization parameters

4.1.2 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 all genes in the genome. Therefore resulting in traits following certain gradual statistical distributions instead of being binned in classes or even binary. Intuitively this might be contradicting with the foundation of modern Genetics - Mendel's three laws. That where derived from observations with where mainly influenced by one locus. But staying with one of Mendel's examples the round or wrinkled surfaces of peas *Pisum sativum*, an assessment of a couple of thousands peas, would most likely inevitably lead to the conclusion that form the "roundest" to the "wrinkliest" pea any gradual step between those is possible and observable. Mendel's third law of independent segregation also only holds true under certain assumptions. The most simplest one being that the traits under investigation have to be located on different linkage groups. Otherwise for the 7 traits used in Mendel's initial studies would not have segregated independently. The odds of 7 randomly selected traits being on 7 different linkage groups are rather small, especially taking into account, that the genome of the *P. sativum* consists of only 7 chromosomes itself Kalo et al., 2004. Mendel probably new about traits not following its own laws, as well as being aware of the quantitative nature of traits such as the constitution of surfaces of peas or the color of petals. But being the pioneer of a then rather unexplored field of science, some of which big questions we fail to satisfactory answer today, he did not have the resources or the knowledge to explain behavior's not "mendeling", that were only able to be deciphered in later decades and centuries based on his ground-breaking work.

Initially thought to be contradicting to Mendel's ideas Darwin proposed the concept's of evolution due to natural selection which also introduce the idea of traits following a gradual distribution 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 1918 Fisher, 1919. His theories combined the then in all fields of science popular research of distributions with genomics. He 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 the biometricians it opened an other long debated question of effect size and the overall architecture of complex traits. While in the theory of monogenic traits the effect size of the single gene on the trait is 1 or 100 % with an increasing number of genes influencing a complex traits the *per sè* contribution of single gene has to decrease with an increasing number of loci determining the value a given trait. In the 1990s it has been thought, that complex traits are predominantly controlled from few genes with a large to medium effect size, while others had a minimal influence Zhang et al., 2018.

With the upcoming popularity of GWAS as the favored method to decipher genetic architectures of traits, or having pioneered in human genetics in became clear that the majority of the 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 Korte and Farlow, 2013, Stringer et al., 2011. This nature of quantitative traits present great challenges to animal Goddard and Hayes, 2009 and plant breeding Würschum, 2012, in further improving crop or livestock performances, as well 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 provide enough challenges as is, all traits will also be influenced by the environment from which an individual originates. 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 4.15.

$$\sigma_P = \sigma_G + \sigma_E \tag{4.15}$$

The genomic and the environmental effects 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 4.15 needs to be extend by the variance of the gene- environment interactions σ_{GxE} **??** , Lynch and Walsh, 1998, Walsh and Lynch, 2018.

$$\sigma_P = \sigma_G + \sigma_E + \sigma_{GxE} \tag{4.16}$$

Equation 4.16 shows the decomposition of the phenotypic variance, to thoroughly understand complex genetic architectures of traits the genetic variance needs to be decomposed further in its additive, dominance and epistatic components 4.17

$$\sigma_G = \sigma_A + \sigma_D + \sigma_I \tag{4.17}$$

The additive effects are caused by single, for this model mostly homozygous, loci while the variance caused by dominance effects, is caused by heterozygous loci and their resulting interactions being full-, over- , co- or underdominant. And lastly the interaction effects that are a result of two or more genes only having an impact if the involved genes co-occur in a certain state. The resulting variance is commonly known as gene-gene interactions and/or epistasis Falconer and Mackay, 1996.
Since possible interactions in a genome can happen between additive or dominant or a combination of those loci. The variance due to interaction effects σ_I can be further dissembled in the variance resulting from additive-additive σ_{AA} dominant-dominant σDD and additive-dominant $sigmaAD$ terms as represented in equation 4.18.

$$\sigma_I = \sigma_{AxA} + \sigma_{DxD} + \sigma_{AxD} \tag{4.18}$$

Knowledge of the variance components involved in the expression of a trait in population, lead up to the estimation of the total influence of all genetic variances and the environmental variance one the phenotypic distribution. This concept if 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 4.19. This is also referred to as broad sense heritability, because all genetic effects including additive, dominance and epistatic effects are included Brooker, 1999.

$$H^2 = \frac{\sigma_A + \sigma_D + \sigma_I}{\sigma_P} \tag{4.19}$$

The concept of narrow-sense heritability 4.20 is similar to the broad-sense heritability, but only the additive genetic effects are included in the genetic part of the equation. This differentiation is import for natural and artificial selection and thus is commonly used in evolutionary genomics and breeding. Because in diploid species each parent only passes down on a single a allele of a give locus. Dominance effects or interaction effects are not commonly inherited from one parent. Therefore the it is mainly the additive genetic effects of a parent that influences its offspring. While the dominance and epistatic variances are controlled by the combination of the parents Falconer and Mackay, 1996, Walsh and Lynch, 2018.

$$h^2 = \frac{\sigma_A}{\sigma_P} \tag{4.20}$$

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

Genomic prediction has been applied to almost all relevant crop and model species. Including: *A. thaliana* Hu et al., 2015, Shen et al., 2013, alfalfa (*Medicago sativa*) Li and Brummer, 2012; Annicchiarico et al., 2015, Li et al., 2015, Biazzi et al., 2017, Hawkins and Yu, 2018; barley Neyhart, Lorenz, and Smith, 2019, Oakey et al., 2016 , Zhong et al., 2009 ; cassava (*Manihot esculenta*) Elias et al., 2018a, Elias et al., 2018b ; cauliflower (*Brassica olearacea spp*) Thorwarth, Yousef, and Schmid, 2018 ; cotton (*Gossiypium spp.* Gapare et al., 2018 ; maze (*Zea mays*) Moeinizade et al., 2019 , Allier et al., 2019 , Brauner et al., 2018, Schrag et al., 2018, Schopp et al., 2017b, Sousa et al., 2017, Schopp et al., 2017a, Kadam et al., 2016, Bustos-Korts et al., 2016a, Montesinos-López et al., 2015, Owens et al., 2014, Lehermeier et al., 2014, Technow et al., 2014, Peiffer et al., 2014 , Riedelsheimer et al., 2013, Guo et al., 2013, Technow, Bürger, and Melchinger, 2013, Windhausen et al., 2012 , Rincent et al., 2012 ; potato (*Solanum tuberosum*), Enciso-Rodriguez et al., 2018, Endelman et al., 2018 ; rape seed (*Brassica naps*) . Würschum, Abel, and Zhao, 2014, Jan et al., 2016, Luo et al., 2017, Werner et al., 2018, Snowdon and Iniguez Luy, 2012, Qian, Qian, and Snowdon, 2014 ; rice (*Oryza sativa*) Momen et al., 2019, Hassen et al., 2018, Xu, 2013, **Grenier2015** ; rye (*Secale cerale*) Auinger et al., 2016, Bernal-Vasquez et al., 2014, Wang et al., 2014, Bernal-Vasquez et al., 2017, Marulanda et al., 2016; soybean (*Glycine max*) Stewart-Brown et al., 2019 , Jarquin, Specht, and Lorenz, 2016, Xavier, Muir, and Rainey, 2016 ; switchgrass (*Panicum virgatum*) Poudel et al., 2019 ,Ramstein and Casler, 2019 , Ramstein et al., 2016 ; wheat (*Triticum aestivum*) Cuevas et al., 2019a , Howard et al., 2019 ,Krause et al., 2019, Rincent et al., 2018, Norman et al., 2018, Belamkar et al., 2018, Ovenden et al., 2018, Sukumaran et al., 2016, Bustos-Korts et al., 2016b, Gianola et al., 2016, Crossa et al., 2016, Thavamanikumar, Dolferus, and Thumma, 2015, Lopez-Cruz et al., 2015, and various tree species Almeida Filho et al., 2019, Rincent et al., 2018, Kainer et al., 2018, Ratcliffe et al., 2017, El-Dien et al., 2016, Kumar et al., 2015, Jaramillo-Correa et al., 2014, Zapata-Valenzuela et al., 2013, Holliday, Wang, and Aitken, 2012, Resende et al., 2012

4.1.4 Genomic selection using artificial neural networks

Genomic selection (GS) has been successfully applied in animal Gianola and Rosa, 2015, Hayes and Goddard, 2010 and plant breeding Crossa et al., 2010, Desta and Ortiz, 2014, Heffner et al., 2010, Crossa et al., 2017a as well as in medical applications, since it was first reported Hayes and Goddard, 2001a. 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., 2017b. The most commonly applied methods include GULP and a set of related algorithms known as the bayesian alphabet Gianola et al., 2009. Genomic prediction in general has repeatedly been shown 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 some 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 usefull in some cases the when the whole genome regression is performed by neural nets 1 While every new method is a valuable addition to the tool-kits for genomic selection, some fundamental problems remain unsolved, of which the n»p problematic stands out. Usually in genomic selection settings the size of the training population (TRN) with n phenotypes is substantially smaller than the number of markers (p) Fan, Han, and Liu, 2014. Making the number of features immensely large, even when SNP-SNP interactions are not considered. Furthermore each marker is treated as an independent observation neglecting collinearity and linkage disequilibrium (LD). Further difficulties arise through non-additive, epistatic and dominance marker effects. The main problem with epistasis issue quantitative genetics is the almost infinite amount of different marker combinations, that cannot be represented within the size of TRN in the thousands, the same problems arises for example in GWA studies Korte and Farlow, 2013. With already large p the number of possible additive SNP-SNP interactions potentiates to $p^{(p-1)}$. Methods that attempt to overcome those issues are EG-BLUP, using 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. The machine learning mimics the architecture of neural networks and are therefore commonly referred to as artificial neural networks (ANN). Those algorithms have widely been applied in many biological fields Min, Lee, and Yoon, 2017 , Lan et al., 2018, Mamoshina et al., 2016, Angermueller et al., 2016 , Webb, 2018, Rampasek and Goldenberg, 2016.

A variety of studies assessed the usability of ML in genomic prediction González-Camacho et al., 2018, González-Camacho et al., 2016, Ogutu, Piepho, and Schulz-Streeck, 2011, Montesinos-López et al., 2019a, Grinberg, Orhobor, and King, 2018 , Cuevas et al., 2019b, Montesinos-López et al., 2019b, Ma et al., 2017, Qiu et al., 2016, González-Camacho et al., 2012 Li et al., 2018. 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 generally aptitude of ML for genomic selection has been repeatedly shown, how no evidence exists that neural networks can outperform or in many cases perform on that same level as mixed-model approaches as GBLUP Hayes and Goddard, 2001b. While 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 of 1 - 3 hidden layers. With 1 layer networks often being the most successful among those. Contradicting to the idea behind machine learning in genomic selection 1 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 show 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. While in CL weights are shared between neurons in LCLs each neuron as 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 a common problem in machine learning. To evaluate the results of Pook et al. 2019 accomplished with simulated data we used the data sets generated in the scope of the 1001 genome project of *Arabidopsis thaliana* Alonso-Blanco et al., 2016

4.2 Proof of concept on ANN-based genomic selection

TABLE 4.1: The effects of treatments X and Y on the four groups studied.

	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

TABLE 4.2: The effects of treatments X and Y on the four groups studied.

	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

4.3 Material

4.3.1 DH populations derived from MAZE landraces

4.3.2 A. thaliana

4.4 Methods

4.4.1 ANN

4.4.2 GBLUP

4.5 Results

4.6 Discussion

5 GWAS

5.1 Reevalulation of 463 phenotypes from the AraPheno database

5.1.1 Introduction

5.1.2 Material and Methods

5.1.3 Results

5.1.4 Results

5.1.5 Disucssion

5.2 GWAS in DH landrace populatios of maze across and within environments

5.2.1 Introduction

5.2.2 Material and Methods

5.2.3 Results

5.2.4 Results

5.2.5 Disucssion

A Source code GWAS-Flow

A.1 gwas.py

```python
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":
        Y_file = sys.argv[i+1]
    elif sys.argv[i] == "-k" or sys.argv[i] == "--kinship":
        K_file = sys.argv[i+1]
    elif sys.argv[i] == "-m":
        m = sys.argv[i+1]
    elif sys.argv[i] == "-a" or sys.argv[i] == "--mac_min":
        mac_min = int(sys.argv[i+1])
    elif sys.argv[i] == "-bs" or sys.argv[i] == "--batch-size":
        batch_size = int(sys.argv[i+1])
    elif sys.argv[i] == "-p" or sys.argv[i] == "--perm":
        perm  = int(sys.argv[i+1])
    elif sys.argv[i] == "-o" or sys.argv[i] == "--out":
        out_file = sys.argv[i+1]
    elif sys.argv[i] == "-h" or sys.argv[i] == "--help":
        print("-x , --genotype :file containing marker information in csv or hdf5 format of size")
        print("-y , --phenotype: file container phenotype information in csv format"  )
        print("-k , --kinship : file containing kinship matrix of size k X k in csv or hdf5 format")
        print("-m : name of columnn containing the phenotype : default m = phenotype_value")
        print("-a , --mac_min : integer specifying the minimum minor allele count necessary for a marker to be included. Default a = 1" )
        print("-bs, --batch-size : integer specifying the number of markers processed at once. Default -bs 500000" )
        print("-p , --perm : single integer specifying the number of permutations. Default 1 == no perm ")
        print("-o , --out : name of output file. Default -o results.csv  ")
        print("-h , --help : prints help and command line options")
        quit()
    else:
        print(’unknown option ’ + str(sys.argv[i]))
        quit()



print("parsed commandline args")
```

```

start = time.time()

X,K,Y_,markers = main.load_and_prepare_data(X_file,Y_file,K_file,m)


## MAF filterin
markers_used , X , macs = main.mac_filter(mac_min,X,markers)

## prepare
print("Begin performing GWAS on ", Y_file)

if perm == 1:
    output = main.gwas(X,K,Y_,batch_size)
    if( X_file.split(".")[-1] == 'csv'):
        chr_pos = np.array(list(map(lambda x : x.split("- "), markers_used)))
    else:
        chr_reg = h5py.File(X_file,'r')['positions'].attrs['chr_regions']
        mk_index= np.array(range(len(markers)),dtype=int)[macs >= mac_min]
        chr_pos = np.array([list(map(lambda x: sum(x > chr_reg[:,1]) + 1, mk_index)), markers_used]).T
        my_time = np.repeat((time.time()-start),len(chr_pos))
    pd.DataFrame({
        'chr' : chr_pos[:,0] ,
        'pos' : chr_pos[:,1] ,
        'pval': output[:,0] ,
        'mac' : np.array(macs[macs >= mac_min],dtype=np.int) ,
        'eff_size': output[:,1] ,
        'SE' : output[:,2]}).to_csv(out_file,index=False)
elif perm > 1:
    min_pval = []
    perm_seeds = []
    my_time = []
    for i in range(perm):
        start_perm = time.time()
        print("Running permutation ", i+1, " of ",perm)
        my_seed  = np.asscalar(np.random.randint(9999,size=1))
        perm_seeds.append(my_seed)
        np.random.seed(my_seed)
        Y_perm = np.random.permutation(Y_)
        output = main.gwas(X,K,Y_perm,batch_size)
        min_pval.append(np.min(output[:,0]))
        print("Elapsed time for permuatation",i+1 ," with p_min",
   min_pval[i]," is",": ", round(time.time() - start_perm,2))
        my_time.append(time.time()-start_perm)
    pd.DataFrame({
        ’time’: my_time ,
        ’seed’: perm_seeds ,
        ’min_p’: min_pval }).to_csv(out_file,index=False)

print("done")

end = time.time()
eltime = np.round(end -start,2)

if eltime <= 59:
    print("Total time elapsed",  eltime, "seconds")
elif eltime > 59 and eltime <= 3600:
    print("Total time elapsed",  np.round(eltime / 60,2) , "minutes")
elif eltime > 3600 :
    print("Total time elapsed",  np.round(eltime / 60 / 60,2), "hours"
   )


```

A.2 main.py

```
    import pandas as pd
    import numpy as np
    from scipy.stats import f
    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]

idx_acc = list(map(lambda x: x in acc_isec, acc_X))
idy_acc = list(map(lambda x: x in acc_isec, acc_Y))
```

```
    idk_acc = list(map(lambda x: x in acc_isec, acc_K))

    Y_ = np.asarray(Y.drop('accession_id',1),dtype=np.float32)[idy_acc,:]

    if type_X == 'hdf5' or type_X == 'h5py' :
        X = np.asarray(SNP['snps'][0:(len(SNP['snps'])+1),],dtype=np.float32)[:,idx_acc].T
        X = X[np.argsort(acc_X[idx_acc]),:]
        k1 = np.asarray(k['kinship'][:])[idk_acc,:]
        K  = k1[:,idk_acc]
        K = K[np.argsort(acc_X[idx_acc]),:]
        K = K[:,np.argsort(acc_X[idx_acc])]
    else:
        X  = X[idx_acc,:]
        k1 = k[idk_acc,:]
        K  = k1[:,idk_acc]



    print("data has been imported")
    return X,K,Y_,markers


def mac_filter(mac_min, X, markers):
    ac1 = np.sum(X,axis=0)
    ac0 = X.shape[0] - ac1
    macs = np.minimum(ac1,ac0)
    markers_used  = markers[macs >= mac_min]
    X = X[:,macs >= mac_min]
    return markers_used, X, macs

def gwas(X,K,Y,batch_size):
    n_marker = X.shape[1]
    n = len(Y)
    ## REML
    K_stand = (n-1)/np.sum((np.identity(n) - np.ones((n,n))/n) * K) * K
```

```python
    vg, delta, ve  = herit.estimate(Y,"normal",K_stand,verbose = False)
    print(" Pseudo-heritability is " , vg / (ve + vg + delta))
    print(" Performing GWAS on ", n , " phenotypes and ", n_marker ," markers")
    ## Transform kinship-matrix, phenotypes and estimate intercpt
    Xo = np.ones(K.shape[0]).flatten()
    M = np.transpose(np.linalg.inv(np.linalg.cholesky(vg * K_stand + ve  * np.identity(n)))).astype(np.float32)
    Y_t = np.sum(np.multiply(np.transpose(M),Y),axis=1).astype(np.float32)
    int_t = np.sum(np.multiply(np.transpose(M),np.ones(n)),axis=1).astype(np.float32)
    ## EMMAX Scan
    RSS_env = (np.linalg.lstsq(np.reshape(int_t,(n,-1)) , np.reshape(Y_t,(n,-1)))[1]).astype(np.float32)
    ## calculate betas and se of betas
    def stderr(a,M,Y_t2d,int_t):
        x = tf.stack((int_t,tf.squeeze(tf.matmul(M.T,tf.reshape(a,(n,-1))))),axis=1)
        coeff = tf.matmul(tf.matmul(tf.linalg.inv(tf.matmul(tf.transpose(x),x)),tf.transpose(x)),Y_t2d)
        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])))))
        SE = tf.math.sqrt(SSE/(471-(1+2)))
        StdERR = tf.sqrt(tf.linalg.diag_part(tf.math.multiply(SE , tf.linalg.inv(tf.matmul(tf.transpose(x),x)))))[1]
        return tf.stack((coeff[1,0],StdERR))
    ## calculate residual sum squares
    def rss(a,M,y,int_t):
        x_t = tf.reduce_sum(tf.math.multiply(M.T,a),axis=1)
        lm_res = tf.linalg.lstsq(tf.transpose(tf.stack((int_t,x_t),axis=0)),Y_t2d)
        lm_x = tf.concat((tf.squeeze(lm_res),x_t),axis=0)
        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)))))
```

```
 ## loop over the batches
 for i in range(int(np.ceil(n_marker/batch_size))):
     tf.reset_default_graph()
     if n_marker < batch_size:
         X_sub = X
     else:
         lower_limit = batch_size * i
         upper_limit = batch_size * i + batch_size
         if upper_limit <= n_marker :
             X_sub = X[:,lower_limit:upper_limit]
             print("Working on markers ", lower_limit , " to ", upper_limit, " of ", n_marker )
         else:
             X_sub = X[:,lower_limit:]
             print("Working on markers ", lower_limit , " to ", n_marker, " of ", n_marker )
     config = tf.ConfigProto()
     n_cores = mlt.cpu_count()
     config.intra_op_parallelism_threads = n_cores
     config.inter_op_parallelism_threads = n_cores
     sess = tf.Session(config=config)
     Y_t2d = tf.cast(tf.reshape(Y_t,(n,-1)),dtype=tf.float32)
     y_tensor =  tf.convert_to_tensor(Y_t,dtype = tf.float32)
     StdERR = tf.map_fn(lambda a : stderr(a,M,Y_t2d,int_t), X_sub.T)
     R1_full = tf.map_fn(lambda a: rss(a,M,Y_t2d,int_t), X_sub.T)
     F_1 = tf.divide(tf.subtract(RSS_env, R1_full),tf.divide(R1_full,(n-3)))
     if i == 0 :
         output = sess.run(tf.concat([tf.reshape(F_1,(X_sub.shape[1],-1)),StdERR],axis=1))
     else :
         tmp = sess.run(tf.concat([tf.reshape(F_1,(X_sub.shape[1],-1)),StdERR],axis=1))
         output = np.append(output,tmp,axis=0)
     sess.close()
     F_dist = output[:,0]
 pval  = 1 - f.cdf(F_dist,1,n-3)
```

```
    output[:,0] = pval
    return output



```

A.3 herit.py

```

def estimate(y, lik, K, M=None, verbose=True):
    from numpy_sugar.linalg import economic_qs
    from numpy import pi, var, diag
    from glimix_core.glmm import GLMMExpFam
    from glimix_core.lmm import LMM
    from limix._data._assert import assert_likelihood
    from limix._data import normalize_likelihood, conform_dataset
    from limix.qtl._assert import assert_finite
    from limix._display import session_block, session_line
    lik = normalize_likelihood(lik)
    lik_name = lik[0]
    with session_block("Heritability analysis", disable=not verbose):
        with session_line("Normalising input...", disable=not verbose):
            data = conform_dataset(y, M=M, K=K)
        y = data["y"]
        M = data["M"]
        K = data["K"]
        assert_finite(y, M, K)
        if K is not None:
           # K = K / diag(K).mean()
            QS = economic_qs(K)
        else:
            QS = None
        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
```

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 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.

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

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

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.

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

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, 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.

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

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.

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

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 Genetics* 6.11, pp. 3733–3747. DOI: 10.1534/g3.116.035410. URL: https://doi.org/10.1534.

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.

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

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

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 Genetics* 6.7, pp. 1819–1834. DOI: 10.1534/g3.116.029637. URL: https://doi.org/10.1534.

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

Crossa, José et al. (2017a). "Genomic selection in plant breeding: methods, models, and perspectives". In: *Trends in plant science* 22.11, pp. 961–975.

Cuevas, Jaime et al. (2019a). "Deep Kernel for Genomic and Near Infrared Predictions in Multi-environment Breeding Trials". In: *G3 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.

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.

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.

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.

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 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.

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.

Freudenthal, Jan A. et al. (2019). “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.

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.

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

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 Genetics* 6.10, pp. 3241–3256. DOI: 10.1534/g3.116.034256. URL: https://doi.org/10.1534.

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.

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.

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 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.

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 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. (2001a). "Prediction of total genetic value using genome-wide dense marker maps". In: *Genetics* 157.4, pp. 1819–1829.

— (2001b). "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.

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.

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 Genetics* 2.9, pp. 1085–1093. DOI: 10.1534/g3.112.002733. URL: https://doi.org/10.1534.

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 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.

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 Genetics* 6.8, pp. 2329–2341. DOI: 10.1534/g3.116.031443. URL: https://doi.org/10.1534.

Jiang, Yong and Jochen C Reif (2015). “Modeling epistasis in genomic selection”. In: *Genetics* 201.2, pp. 759–768.

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 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.

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 Genetics*, g3.200856.2018. DOI: 10.1534/g3.118.200856. URL: https://doi.org/10.1534.

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

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.

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.

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, 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, 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.

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.

Lopez-Cruz, Marco et al. (2015). "Increased Prediction Accuracy in Wheat Breeding Trials Using a Marker x Environment Interaction Genomic Selection Model". In: *G3 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.

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*.

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.

Min, Seonwoo, Byunghan Lee, and Sungroh Yoon (2017). "Deep learning in bioinformatics". In: *Briefings in bioinformatics* 18.5, pp. 851–869.

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 Genetics* 9.10, pp. 3369–3380. DOI: 10.1534/g3.119.400346. URL: https://doi.org/10.1534/g3.119.400346.

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.

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 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.

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 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.

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

Poudel, Hari P. et al. (2019). "Genomic Prediction for Winter Survival of Lowland Switchgrass in the Northern USA". In: *G3 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 (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 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 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 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.

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 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.

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

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.

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.

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

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.

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.

Stewart-Brown, Benjamin B. et al. (2019). “Genomic Selection for Yield and Seed Composition Traits Within an Applied Soybean Breeding Program”. In: *G3 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.

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 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.

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 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.

Van Rossum, Guido and Fred L Drake Jr (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Walsh, Bruce and Michael Lynch (2018). *Evolution and selection of quantitative traits*. Oxford University Press.

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.

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.

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.

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.

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.

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.

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

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.

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*.