Combining traditional methods with novel machine learning techniques to understand the translation of genetic code into biological function

vorgelegt von
M.Sc.
Bettina Mieth
ORCID: 0000-0001-6055-6869

an der *Fakultät IV – Elektrotechnik und Informatik* der *Technischen Universität Berlin*
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte
Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Manfred Opper
Gutachter:	Prof. Dr. Klaus-Robert Müller
	Prof. Dr. Arcadi Navarro
	Prof. Dr. Peter Martus

Tag der wissenschaftlichen Aussprache: Berlin, 8. Juni 2021

Berlin 2021

"*The whole is greater than the sum of its parts.*"

Aristotle, 4th century BC

Abstract

One of the great challenges in modern biology is understanding the genome and its translation into biological structures and function. In this context, the aim of this dissertation is to show that combinatorial approaches of traditional methods and novel machine learning ideas can be developed and successfully applied to analyze large-scale biological datasets and provide novel insights into genetic and transcriptomic variation. This proposed thesis is validated in two fields of biological research: genome-wide association studies (GWAS) and single-cell RNA sequencing (scRNA-Seq). For the analysis of such data, we propose three novel methods, each consisting of traditional methods on the one hand and state-of-the-art machine learning algorithms on the other. It is shown that these combinatorial approaches outperform both their individual methodological components and existing techniques on suitable corresponding datasets in terms of statistical power and accuracy.

The standard approach to the evaluation of GWAS is based on testing each position in the genome individually for statistical significance of its association with the phenotype under investigation. To improve the analysis, we propose a combination of machine learning and statistical testing that takes *correlation structures* within the set of single-nucleotide polymorphisms (SNP) under investigation in a mathematically well-controlled manner into account. The general idea is to train an appropriate state-of-the-art classification algorithm, selecting a subset of candidate locations that are most relevant for the classifier's decisions and examining only those for significant associations via multiple statistical hypothesis testing. This dissertation's first methodological contribution, the two-step algorithm, COMBI, first trains a support vector machine to determine a subset of candidate SNPs and then performs hypothesis tests for these SNPs together with an adequate threshold correction. Applying COMBI to generated datasets as well as data from a WTCCC study (2007), we show that the novel method outperforms ordinary raw *p*-value thresholding and other state-of-the-art methods. COMBI presents higher power and precision than the examined alternatives while yielding fewer *false* (*i.e.* non-replicated) and more *true* (*i.e.* replicated) discoveries when its results are validated on later GWAS.

Deep learning has become one of the leading methodologies in data science, which oftentimes greatly improves prediction performances in comparison to conventional approaches. Recently, explainable artificial intelligence has emerged as a novel area of research that goes beyond pure prediction improvement by extracting knowledge from deep learning methodologies through the interpretation of their results. Following these developments, we present the second methodological contribution of this dissertation, DeepCOMBI - an improved, deep learning- and explanation-based extension of the previously proposed method COMBI. The three-step algorithm of DeepCOMBI first trains a neural network to classify subjects into their respective phenotypes. Second, it explains the classifier's decisions by applying layer-wise relevance propagation as one

example from the pool of explanation techniques. The resulting importance scores are eventually used to determine a subset of the most relevant locations for multiple hypothesis testing in the third step, which remains unchanged as in the original COMBI method. DeepCOMBI is shown to outperform COMBI, raw *p*-value thresholding and other baseline methods on generated datasets and the 2007 WTCCC study.

Beyond improving the identification of associations between phenotypes and genotypes, in this dissertation, we contribute to understanding how genetic information is translated into physical structures and biological function. When exploring the flow of sequential information from DNA to mRNA to proteins, we interpret the genome in the context of cell types and aim to identify the genes that are active in certain cells. Within this frame of reference, the goal of scRNA-Seq experiments is to define and catalog cell types from the transcriptional output of individual cells, which refers to an unsupervised clustering problem. To improve the clustering of small disease- or tissue-specific datasets, for which the identification of rare cell types is often problematic, we propose to combine conventional clustering algorithms with the machine learning concept of transfer learning to utilize large and well-annotated reference datasets. This dissertation's third methodological contribution modifies the target dataset while incorporating key information from the reference dataset via non-negative matrix factorization before providing the modified dataset to a traditional downstream clustering algorithm. We empirically evaluate the benefits of the novel approach on simulated scRNA-Seq data as well as on publicly available datasets. Finally, we present results for analyzing a recently published small dataset and find improved clustering when transferring knowledge from a large independent reference dataset.

To summarize, this dissertation contributes to a better understanding of the genome and the processes around its translation into biological structures and function. By proposing three approaches for the analysis of large-scale biological datasets combining traditional methods and state-of-the-art machine learning algorithms, it is shown that, in this regard, too, "*the whole is greater than the sum of its parts*" (indirect quote derived from Aristotle, 4th century BC).

Zusammenfassung

Eine der größten Herausforderungen der modernen Biologie besteht darin, das Genom und seine Umwandlung in biologische Strukturen und Funktionen zu verstehen. In diesem Zusammenhang wird in dieser Dissertation gezeigt, dass kombinatorische Ansätze traditioneller Methoden und neuartiger Ideen des maschinellen Lernens entwickelt und erfolgreich angewendet werden können, um große biologische Datensätze zu analysieren und neue Einblicke in genetische und transkriptomische Variationen zu erhalten. Diese für diese Arbeit aufgestellte These wird in zwei Bereichen der biologischen Forschung validiert: genomweite Assoziationsstudien (GWAS) und Einzelzell-RNA-Sequenzierung (scRNA-Seq). Es werden insgesamt drei neue Methoden vorgeschlagen, die jeweils aus traditionellen Methoden auf der einen Seite und modernen maschinellen Lernalgorithmen auf der anderen Seite bestehen. Es wird gezeigt, dass diese kombinatorischen Ansätze sowohl ihre einzelnen methodischen Komponenten als auch andere bereits existierende Konkurrenzmethoden bei der Anwendung auf entsprechenden Datensätzen hinsichtlich statistischer Power und Accuracy übertreffen.

Der Standardansatz für die Auswertung von GWAS basiert darauf, jede Position im Genom einzeln auf statistische Signifikanz ihrer Assoziation mit dem untersuchten Phänotyp zu testen. Um die Analyse zu verbessern, schlagen wir eine Kombination aus maschinellem Lernen und statistischem Testen vor, bei der Korrelationsstrukturen zwischen den untersuchten Einzelnukleotid-Polymorphismen (SNP) mathematisch kontrolliert berücksichtigt werden. Die zugrundeliegende Idee besteht darin, zunächst einen geeigneten Klassifizierungsalgorithmus zu trainieren, danach die Teilmenge aller SNPs auszuwählen, die für die Entscheidungen des Klassifizierers am relevantesten sind und letztendlich diese mit multiplen statistischen Hypothesentests auf signifikante Assoziationen zu untersuchen. Der erste im Rahmen dieser Dissertation entwickelte, zweistufige Algorithmus COMBI trainiert zunächst eine Support Vector Machine, um die Teilmenge der bedeutendsten Kandidaten-SNPs zu bestimmen und führt dann Hypothesentests mit einer entsprechenden Anpassung des Signifikanzlevels für diese SNPs durch. Mit der Anwendung von COMBI auf generierten Datensätzen sowie auf Daten aus einer WTCCC-Studie (2007) wird gezeigt, dass die neue Methode bessere Ergebnisse liefert als gewöhnliches multiples Testen sowie andere Konkurrenzmethoden. COMBI ermöglicht höhere statistische Power und Präzision als die untersuchten Alternativen und liefert weniger falsche (d.h. nicht replizierte) und mehr wahre (d.h. replizierte) Entdeckungen, wenn die jeweiligen Ergebnisse mit unabhängigen GWAS validiert werden.

In den letzten Jahren wurde tiefes Lernen zu einer der führenden Methoden der Datenwissenschaften, die die Vorhersageleistungen im Vergleich zu herkömmlichen Ansätzen häufig erheblich verbessert. In jüngster Zeit hat sich zudem erklärbare künstliche Intelligenz (Explainable AI) zu einem neuartigen Forschungsgebiet entwickelt, das über die reine Vorhersageverbesserung hinausgeht und Wissen aus

Deep-Learning-Methoden extrahiert, indem ihre Ergebnisse interpretiert und erklärt werden. Im Rahmen dieser Fortschritte entwickeln wir eine Erweiterung von COMBI, die auf tiefem Lernen und erklärbarer künstlicher Intelligenz basiert. Dieser zweite im Rahmen der Dissertation entwickelte, dreistufige Algorithmus DeepCOMBI trainiert zunächst ein neuronales Netzwerk für die Klassifizierung von Probanden in ihre jeweiligen Phänotypen. Anschließend werden die Entscheidungen der Klassifizierung mit Layerwise Relevance Propagation erklärt und die Ergebnisse verwendet, um die relevantesten SNPs zu identifizieren. Wie bei der ursprünglichen COMBI-Methode werden diese SNPs im dritten Schritt auf statistische Assoziation getestet. Auf generierten Datensätze und der bereits genannten WTCCC Studie von 2007 wird gezeigt, dass DeepCOMBI bessere Vorhersageleistungen erbringt als COMBI, gewöhnliches multiples Testen und andere Konkurrenzmethoden.

Über die Verbesserung der Identifizierung von Assoziationen zwischen Phänotypen und Genotypen hinausgehend, tragen wir in dieser Dissertation dazu bei, besser zu verstehen, wie genetische Informationen in phänotypische Strukturen und biologische Funktionen übersetzt werden. Bei der Untersuchung der Umwandlung genetischer Informationen von DNA über mRNA zu Proteinen wird das Genom häufig im Kontext von Zelltypen interpretiert, indem untersucht wird, welche Gene in bestimmten Zellen aktiv sind. In diesem Kontext ist das Ziel von scRNA-Seq-Experimenten die Definition und Katalogisierung von Zelltypen basierend auf dem Transkriptom einzelner Zellen, was auf ein unüberwachtes Clustering-Problem hinausläuft. Beim Clustern von kleinen krankheits- oder gewebespezifischen Datensätzen ist die Identifizierung seltener Zelltypen häufig problematisch. Deshalb schlagen wir vor, herkömmliche Clustering-Algorithmen mit dem Konzept des Transfer Learnings zu kombinieren, um große und gut untersuchte Referenzdatensätze verwenden zu können. Der dritte im Rahmen der Dissertation vorgeschlagene, kombinatorische Ansatz modifiziert daher den Zieldatensatz, indem Informationen aus dem Referenzdatensatz über eine nichtnegative Matrixfaktorisierung einbezogen werden, bevor der modifizierte Datensatz mit einem Clustering-Algorithmus analysiert wird. Die Leistung der vorgeschlagenen Methode wird auf simulierten scRNA-Seq-Daten sowie auf öffentlich verfügbaren Datensätzen empirisch evaluiert. Schließlich präsentieren wir die Ergebnisse der Analyse eines kürzlich veröffentlichten kleinen Datensatzes und finden ein verbessertes Clustering beim Transfer von Informationen aus einem großen Referenzdatensatz.

Zusammenfassend trägt diese Dissertation zu einem besseren Verständnis des Genoms und der Prozesse rund um seine Übersetzung in biologische Strukturen und Funktionen bei. Mit der Entwicklung dreier kombinatorischer Ansätze für die Analyse biologischer Datensätze aus traditionellen Methoden einerseits und modernen Algorithmen des maschinellen Lernens andererseits, wird gezeigt, dass auch hier "*das Ganze mehr ist als die Summe seiner Teile*" (sinngemäß Aristoteles, 4. Jh. v. Chr.).

Acknowledgements

I am very grateful for all the support, guidance and inspiration I have been blessed with over the last couple of years. Academic progress and scientific advances, as well as personal development, are more often than not the results of people coming together and supporting each other. As Aristotle said more than 23 centuries ago, "*the whole is greater than the sum of its parts*" (Aristotle, 4th century BC). This dissertation could not have been possible without the many advisors, colleagues, family and friends who contributed to my success.

First of all, I would like to thank my Ph.D. supervisor Prof. Dr. Klaus-Robert Müller, who introduced me - as a complete newbie - to the field of machine learning in 2012. He placed me on a fascinating initial project, where I got to work with him and other incredibly inspiring minds. He always had my back throughout the years and provided invaluable scientific feedback and advice whenever I asked for it. When I announced I was pregnant in the middle of my Ph.D. (twice!), he fully supported me and made the balancing act between family life and academic work possible.

I thank Prof. Dr. Arcadi Navarro, Prof. Dr. Peter Martus and Prof. Dr. Manfred Opper, who sacrificed some of their valuable time for being part of my doctoral committee.

I want to thank my initial advisor, Prof. Dr. Marius Kloft, who always took so much time out of his busy days when I needed him. His knowledge seemed boundless (who knows him, knows) and I never left his office with any unanswered questions. I know it is rare and I fully appreciate the huge amount of effort he put into supervising my work as closely as he did in the beginning phase of my Ph.D.

Prof. Dr. Daniel Schunk, Prof. Dr. Arcadi Navarro, Prof. Dr. Gilles Blanchard, Prof. Dr. Ernst Fehr, Dr. Daniel Ziemek and Dr. Alex Gutteridge also guided me through my Ph.D. and directed significant parts of my scientific work. It was always a pleasure to follow their inspiring thought processes and I was honored to be the one executing their ideas. I learned from them how scientific research is done.

I thank all of my co-authors for working with me in such a fruitful way and for allowing me to use parts of our publications in this dissertation.

The two most important people in my day-to-day Ph.D. life were my colleagues, Dr. Marina M.-C. Höhne and Dr. Nico Görnitz. They were always there when computers crashed, bugs were found, papers got rejected (or accepted!), when it was time for lunch or a coffee break, when frustration levels were rising or personal life had to be put first. They encouraged me to keep going when things were bad and celebrated with me when things suddenly fell into place. Beyond that, they always shared their incredible knowledge with me and provided countless important pieces of advice.

It was a great pleasure to work with Dr. Juan Antonio Rodríguez and Dr. James R.F. Hockley on our projects. I could always count on them and was often in awe of the work they did. I also thank them for proofreading parts of this work and providing such constructive feedback on the general storyline of my dissertation.

Moreover, I wish to thank Alexandre Rozier and Robin Vobruba for letting me supervise them. I always looked forward to both their questions and answers.

I am very thankful for all of my colleagues at the Machine Learning Lab at TU Berlin, who created such an inspiring and friendly work atmosphere. Honorable mentions go to Irene, Anne, Miriam, Franziska, Ann-Kathrin, Alex, Shin, Daniel, Sergej and Johannes.

Furthermore, I send my gratitude to Andrea Gerdes, Cecilia Bonetti, Andrea Scherz, Dominik Kühne, Imke Weitkamp and Kerstin Rudolph for their patient and comprehensive organizational and technical support.

I also appreciate the opportunities I had by being part of the Max Planck School of Cognition and meeting numerous inspiring people there.

I would like to thank my best friend Lisa and my former fellow students Aileen, Maik and Oli for being there throughout my academic and personal career. They accompanied me from being a lost bachelor student to becoming a confident mother and finally finishing my Ph.D. I am looking forward to sharing everything else that is ahead.

My mom, my sister and my dad are my emotional safety net. I would not be where I am without them and could not be more grateful for their unconditional love and support.

I thank my partner, Pat, from the bottom of my heart. His encouragement and support in all aspects of life give me strength and confidence. Not even working full-time without childcare while writing a dissertation during a pandemic and nation-wide lockdown can bring us down. I thank him for everything he does.

Finally, I dedicate this work to my children, Malik and Eleni, who are everything to me, always.

Contents

1 Overview..... 1

1.1 The author's thesis..... 1
1.2 General storyline of this dissertation..... 1
1.3 Organization of this dissertation..... 6
1.4 Previously published work..... 9
1.5 Abbreviations..... 10

2 Fundamentals..... 13

2.0 Notation of chapter 2..... 14

2.1 Biological background..... 16
2.1.1 The human genome..... 16
2.1.2 From DNA to protein - the central dogma of molecular biology..... 19
2.1.3 Study types under investigation in this dissertation..... 21
Genome-wide association studies..... 21
Single-cell RNA sequencing studies..... 22

2.2 Traditional analysis methods..... 23
2.2.1 Multiple statistical hypothesis testing for GWAS..... 23
2.2.2 Clustering methods for scRNA-Seq studies..... 24

2.3 Machine learning concepts..... 27
2.3.1 Machine learning basics..... 27
2.3.2 Support vector machines..... 28
2.3.3 Neural networks..... 31
2.3.4 Explainable AI and layer-wise relevance propagation..... 37
2.3.5 Non-negative matrix factorization..... 40
2.3.6 Transfer learning..... 41

3 Combining machine learning with multiple statistical hypothesis testing for genome-wide association studies..... 45

3.0 Notation of chapter 3..... 46

3.1 Introduction..... 48

3.2 Methods..... 53
3.2.1 Problem setting..... 53
3.2.2 Proposed workflow..... 54
The COMBI method..... 55
The DeepCOMBI method..... 60
3.2.3 Datasets and corresponding validation strategies..... 64
3.2.4 Preprocessing and parameter selection..... 67
3.2.5 Baseline methods..... 71
3.2.6 Performance metrics..... 74

3.3 Results……75
3.3.1 Results on generated datasets……75
3.3.2 Results on WTCCC data……85

3.4 Summary and discussion……100

4 Combining transfer learning with clustering methods for single-cell RNA sequencing studies……109

4.0 Notation of chapter 4……110

4.1 Introduction……111

4.2 Methods……115
4.2.1 Problem setting……115
4.2.2 Proposed workflow……115
4.2.3 Datasets and corresponding validation strategies……117
4.2.4 Preprocessing and parameter selection……122
4.2.5 Baseline methods……126
4.2.6 Performance metrics……127

4.3 Results……128
4.3.1 Results on generated datasets……128
4.3.2 Results on subsampled source and target datasets……132
4.3.3 Results on independent source and target datasets……134

4.4 Summary and discussion……139

5 Conclusion……145

Appendix……151

I. Parameter investigation on the conservativeness of the COMBI method on generated datasets……151

II. Stability analysis of the COMBI method on WTCCC data……156

III. Extended clustering analysis on independent source and target datasets using various source data labels as a priori knowledge……158

Author's publications and contributions……163

Bibliography……165

1 Overview

1.1 The author's thesis

This dissertation regards the validation of the following thesis:

"*Combinatorial approaches of traditional methods and novel machine learning ideas for the analysis of large-scale biological datasets can be developed and successfully applied to better understand the translation of genetic code into phenotypes and biological function increasing the statistical power and accuracy of existing techniques.*"

1.2 General storyline of this dissertation

This dissertation focuses on developing and applying artificial intelligence methods for the interrogation of large-scale biological data to improve our understanding of genetic variation and its translation into biological function in health and disease. In particular, the thesis to be validated in this work is that machine learning and traditional methods can be combined and successfully applied to biological datasets to provide greater insight into cellular and organismal phenotype, function and processes. It is to be shown that such combinatorial approaches can outperform appropriate competitor techniques on suitable biological datasets in terms of statistical power and accuracy. The dissertation's overall storyline is described in the following sections explaining, in brief, biological background, general ideas and methods, empirical results and concluding findings.

The entirety of an organism's genetic material is called a genome and is present in each of its individual (somatic) cells[1]. In most organisms, it is composed of multiple deoxyribonucleic acid (DNA) biopolymer chains built from four basic chemical units (including adenine (A), cytosine (C), guanine (G) and thymine (T)) called nucleotides. The genome sequences contain both coding regions called genes, which encode protein sequences, and noncoding regions, which serve other important functions such as gene regulation[1]. For information transfer, genes are transcribed into intermediate chain molecules of messenger ribonucleic acid (mRNA), which are then translated into sequences of amino acids. Eventually, these polypeptides may post-translationally be folded, combined and modified further to generate proteins, which are considered to be amongst the most essential functional molecules of life. They form structures, catalyze chemical reactions and hence determine phenotypes and functionalities of cells and, ultimately, the organism[1].

The process of determining the order of nucleotides in the genome is called DNA sequencing and has enabled great advances in the field of biological and medical

research[2]. The first time an entire human genome was sequenced and all genes were mapped to specific positions on the sequence was the result of over ten years of research and cost almost three billion dollars[3,4]. Once the Human Genome Project - a large team of research groups and scientists - succeeded in 2003, the entire human genome sequence of three billion letters was published[3,4]. One might have concluded that the mystery of the human genome and its hereditary function was solved. However, the published sequence was only an exemplary chimeric genome sequence: No two individuals ever have the same DNA sequence. In addition to large structural variations, they always differ at a large number of specific locations in the genome, which are called genetic variants or single-nucleotide polymorphisms (SNPs)[1]. At this point, the interdisciplinary scientific field of genomics was only at its beginning and more and more individuals' DNA was sequenced to identify those SNPs.

What followed in the years after 2003 was an impressive genome sequencing revolution[2]. Nowadays, sequencing a single human genome no longer takes ten years and three billion dollars but can be done in less than 24 hours (*e.g.* with the Illumina NovaSeq 6000 platform) for less than 1,000 dollars[5]. Millions of different genomes have been sequenced and large-scale datasets are available for analysis[6]. The methodological challenge has shifted away from one of physical sequencing to one of data storage, handling, analysis and interpretation. To gain biological insight, the aim is now to convert the sheer abundance of data (*i.e.* millions of sequences of three billion letters each) into an improved understanding of health and disease. In this dissertation, the aim is to contribute to discovering what can be learned from all of this data by introducing state-of-the-art machine learning methods that, in combination with more traditional analysis approaches, can improve our understanding of the genome and, ultimately, provide novel biological insights.

One way to identify and understand the meaning of genetic sequences is by conducting genome-wide association studies (GWAS), where the genomes of a group of cases (with a disease or trait) and a group of (healthy) controls (without the trait) are sequenced and compared. SNPs that are statistically associated with the disease or trait under investigation are discovered by creating tables of genotype counts and calculating the corresponding *p*-values. One of the most widely used repositories for the findings of such studies is the GWAS Catalog, which is a valuable resource that - as of April 2021 - contains the results of almost 5,000 published GWAS identifying over 250,000 associated SNPs[7]. Surprisingly, however, these SNPs explain only a small fraction of individual traits and most of the heritability remains unexplained. This phenomenon is referred to as the „*mystery of missing heritability*"[8,9] and might be caused by the fact that all SNPs are separately tested for association. An individual *p*-value of a SNP only depends on the data in that SNP ignoring any interactions between SNPs and possible correlations with the rest of the genome. To overcome this drawback of the conventional multiple testing methods to analyze GWAS, we propose to employ the potential of artificial intelligence. Machine learning approaches aimed at predicting a phenotype are

not based only on the information at a specific SNP but take the entire dataset, *i.e.* all SNPs and correlation structures, into account. If we identify the specific positions in the genome that are most important for the decision of a classifier, we can use this information as an indicator for the relevance of each specific SNP in addition to the raw *p*-values of statistical testing. This dissertation's contribution lies in developing combinatorial approaches that use a machine learning-based algorithm and statistical testing to identify disease-associated SNPs.

The proposed method, called COMBI[10], first trains a support vector machine (SVM)[11–13] to predict a phenotype based on genotypic data. Subsequently, in the second step, the resulting SVM weight vector is interpreted as an importance score to select an appropriate subset of candidate SNPs based on their relevance for phenotype prediction. The final third step consists of statistically testing only those preselected SNPs for association with the phenotype under investigation. As a result of the screening step, COMBI elegantly filters out any irrelevant noise SNPs and enables a clear identification of the most important associated SNPs via multiple testing. It is shown on both generated semi-real datasets as well as on real 2007 data of seven major diseases that COMBI outperforms all relevant baseline methods in terms of both power and precision.

Following recent developments in machine learning and the rise of deep neural networks (DNN) as the most successful prediction tools[14], we develop an improved version of COMBI by introducing DeepCOMBI[15] as a deep learning-based extension of the proposed method. Here, a DNN is trained instead of an SVM and layer-wise relevance propagation (LRP)[16–18] is used as an explanation method for identifying relevant SNPs. The final step of statistical testing remains unchanged. High-performing deep learning techniques and state-of-the-art explanation methods significantly improve the statistical power of the combinatorial approach and are shown to perform even better than the original COMBI method. Both methods help to increase the heritability, which can be accounted for in the GWAS Catalog.

With the proposal of two novel methods for the analysis of GWAS, a substantial contribution is made for identifying important positions in the genome when considering genome trait associations. Beyond that, when trying to understand how genetic information is translated into phenotypic structures and function, the goal is to focus on actual causation rather than raw association. The scientific question often asked is whether a genetic variant is biologically meaningful. Why do we find a specific SNP to be associated with a disease or trait? What happens with the associated gene during the natural process of a cell's life to actually cause disease? If we stop our investigation after identifying significant SNP disease associations, it is, metaphorically speaking, like having discovered the ingredients list of a recipe while knowing the resulting dish but missing any kind of cooking instructions. To figure out the entire recipe, we have to explore the flow of genetic information and investigate how the

genome is translated into function. To this end, it is essential to interpret the genome in the context of cell types. There is an enormous variety of cell types and tissues in the human body[1]. Some are easily distinguishable, *e.g.* skin cells from brain cells or blood cells from muscle cells; others look morphologically identical but still have entirely different functions or even change their function over time[1]. All of them contain the exact same genome, indicating that an identical genetic sequence is being implemented into function differently. Cells do not constantly express the whole genome but the specific pieces of the genetic sequence that are activated in different combinations vary from cell to cell and can additionally change over time. Hence, instead of investigating genetic function only at the DNA level, it is crucial to examine which genes are active in certain cells by following the flow of genetic information from DNA to mRNA to proteins. Because specific genes are activated and repressed in different cell types at different points in time and the goal is to explain how genetic information causes biological function, we need to explore which and how many mRNA molecules can be found in certain cells. Going back to the initial challenge of explaining the results of a GWAS and determining why specific significant GWAS SNPs increase the risk of developing a disease, we need to identify where these SNPs are transcribed, in what types of cells the corresponding genes are active and hence where the corresponding mRNA molecules can be found.

A novel technology in this area of biology, called single-cell RNA sequencing (scRNA-Seq), was developed in 2009[19] and has given rise to a very fast-moving research field in the following years[20]. It allows us to analyze the entirety of mRNA molecules - called the transcriptome - of individual cells when it was previously only possible to look at pooled transcriptomes. All cells of a sample taken from the tissue under investigation are separated and the number of the different mRNA molecules present in each cell is determined in a complicated process of extraction, amplification, sequencing and library alignment. One of the most urgent research questions about the resulting datasets is clustering the individual cells into groups based on their transcriptomes through unsupervised clustering[21]. The challenge that often remains is that high experimental barriers in scRNA-Seq (*e.g.* relative cost and inaccessibility of rare tissues) cause many datasets to be small but high-dimensional, where rare subtypes of cells are poorly represented[21]. To overcome this, the idea presented in this dissertation is to use the machine learning concept of transfer learning for clustering scRNA-Seq data. We propose to utilize prior knowledge from large, well-annotated reference datasets to modify small novel target datasets and, as a consequence, improve the clustering of traditional downstream clustering algorithms[22]. The general methodological approach of the proposed method is to use non-negative matrix factorization (NMF)[23–25] of the source dataset to reconstruct a modified version of the target dataset, which is of improved quality having been adjusted to the clustering information available in the source dataset. The reconstructed target dataset is eventually clustered with a widely used single-cell clustering algorithm. In this

dissertation, we show that clustering the modified dataset performs better than clustering the original dataset or the concatenation of source and target dataset. Once again, a combinatorial approach of existing analysis methods and novel machine learning ideas can help analyze biological datasets.

To summarize, this dissertation validates the thesis proposed by the author and contributes to one of the most important challenges in biology of understanding the genome by proposing combinatorial approaches of traditional methods and machine learning. This is in agreement with Aristotle, who once claimed "*the whole is greater than the sum of its parts*"[26] (indirect quote derived from Aristotle, 4th century BC) while trying to explain the ambivalent facets of the definition and terms of objects and parts. The first two proposed methods can identify positions in the genome that influence the risk of developing a disease or trait and the third method improves the clustering of cells into groups based on the active genes in that cell. All methods help us to better understand the translation of genetic code into biological function, increasing the statistical power and accuracy of existing techniques. In combination, the three proposed methods could be used to first identify SNPs that are significantly associated with a disease or trait and second determine in what types of cells the corresponding gene is active. As an outlook, it can be envisioned that the results of the proposed methods will find their way into practical applications. For example, the outcome of such studies could be used for data-driven diagnosis, predictive personal prognosis, the identification of potential drug targets or the design of optimal treatment plans.

1.3 Organization of this dissertation

Chapter 1 provides an overview of this dissertation and begins with stating the general thesis to be investigated in the course of this work. A short and easy-to-read summary for a general audience follows, containing, in brief, the overall storyline of this dissertation. A guide through the chapters of the dissertation is given, along with a list of the author's relevant publications. A list of abbreviations is presented at the end of this chapter.

Important notation is always defined and collected in a list at the beginning of each chapter. Please note that the respective notation is only valid in the corresponding chapter.

Chapter 2 introduces essential background information that is necessary to address the author's proposed thesis of superior combinatorial approaches that improve traditional methods with the application of machine learning for research questions concerning the translation of genetic information into biological structures and function. After covering the biological topics this dissertation focuses on, the specific challenges of genome-wide association studies and single-cell RNA sequencing datasets are presented. Finally, the corresponding traditional approaches to solving such problems and the relevant machine learning concepts are introduced to set up their combined application in subsequent chapters.

In the following main part of this dissertation, three completely novel methods are introduced and it is shown that combinatorial approaches for the presented datasets can be successfully implemented outperforming its individual components as well as other appropriate baseline methods.

In **Chapter 3,** traditional methods for the analysis of genome-wide association studies are combined with machine learning approaches to increase the statistical power of such studies. We present two novel methods, called COMBI and DeepCOMBI, which are based on a combination of multiple hypothesis testing and a support vector machine or a deep neural network, respectively. It is shown that both methods outperform relevant competitor approaches on generated datasets in a controlled environment as well as on a real 2007 GWAS dataset of seven major diseases.

Chapter 4 introduces a novel method, called TransferCluster, to combine the machine learning concepts of transfer learning and non-negative matrix factorization with a traditional clustering method to analyze single-cell RNA sequencing datasets. In an empirical study of three different settings - generated, subsampled and independent source and target datasets - the performance of the proposed method is investigated and found to be preferable compared to all investigated baseline methods.

The dissertation concludes on the validity of the author's thesis in **Chapter 5** and shows that successful combinatorial approaches of machine learning and traditional methods were developed to better understand the translation of genetic code into biological function. The main findings of this dissertation are summarized, open problems are discussed and an outlook on future research directions is presented.

The **Appendix** includes a number of additional experiments. At the end of the dissertation, there is a list of all of the author's publications, along with statements of contributions. Please note that references in the **Bibliography** are numbered in consecutive order as they appear in the text and superscript Arabic numerals are used to cite.

1.4 Previously published work

Parts of this dissertation have previously been published as journal articles, which are listed below.

A. **Bettina Mieth**, Marius Kloft, Juan Antonio Rodríguez, Sören Sonnenburg, Robin Vobruba, Carlos Morcillo-Suárez, Xavier Farré, Urko M. Marigorta, Ernst Fehr, Thorsten Dickhaus, Gilles Blanchard, Daniel Schunk, Arcadi Navarro & Klaus-Robert Müller. Combining Multiple Hypothesis Testing with Machine Learning Increases the Statistical Power of Genome-wide Association Studies. *Scientific Reports*, volume 6, article number: 36671 (2016)[10].

B. **Bettina Mieth**, James R.F. Hockley, Nico Görnitz, Marina M.-C. Höhne, Klaus-Robert Müller, Alex Gutteridge & Daniel Ziemek. Using transfer learning from prior reference knowledge to improve the clustering of single-cell RNA-Seq data. *Scientific Reports*, volume 9, article number 920353 (2019)[22].

C. **Bettina Mieth**, Alexandre Rozier, Juan Antonio Rodriguez, Marina M.-C. Höhne, Nico Görnitz & Klaus-Robert Müller. DeepCOMBI: Explainable artificial intelligence for the analysis and discovery in genome-wide association studies. Currently under review at *NAR Genomics and Bioinformatics*. bioRxiv. doi.org/10.1101/2020.11.06.371542 (2020)[15].

The content of this dissertation is related to the above publications in the following way: **Chapter 2** contains parts of **A-C**. **Chapter 3** is based on **A** and **C**; **Chapter 4** is based on **B**; **Chapter 5** contains material from **A-C**.

I thank all of my co-authors for allowing me to include parts of our publications in this dissertation.

1.5 Abbreviations

Abbreviation	Definition (page it is introduced on)
A	Adenine (16)
AI	Artificial intelligence (27)
ARI	Adjusted Rand index (127)
AUC	Area under the curve (74)
BD	Bipolar disorder (65)
C	Cytosine (16)
CAD	Coronary artery disease (65)
CD	Crohn's disease (65)
cDNA	Complementary deoxyribonucleic acid (22)
CNN	Convolutional neural network (35)
CPM	Counts per million (22)
DNA	Deoxyribonucleic acid (16)
DNN	Deep neural network (32)
DRG	Dorsal root ganglia (120)
ENFR	Expected number of false rejections (23)
eQTL	Expression quantitative trait locus (89)
FN	False negative (74)
FP	False positive (74)
FPR	False-positive rate (74)
FWER	Family-wise error rate (23)
G	Guanine (16)
gFWER	Generalized family-wise error rate (23)
GWAS	Genome-wide association studies (21)
HT	Hypertension (65)
KTA	Kernel target alignment (117)
LD	Linkage disequilibrium (18)
LMM	Linear mixed model (73)
LRP	Layer-wise relevance propagation (37)
ML	Machine learning (27)
mRNA	Messenger ribonucleic acid (19)
NMF	Non-negative matrix factorization (40)
NN	Neural network (31)
PCA	Principal component analysis (25)
PFI	Permutation feature importance (49)
PMID	PubMed identification number (88)

PR	Precision-recall (74)
RA	Rheumatoid arthritis (65)
ReLU	Rectified linear unit (activation function) (33)
RNA	Ribonucleic acid (19)
ROC	Receiver operating characteristic (74)
RPM	Reads per million (22)
RPKM	Reads per kilobase per million mapped reads (22)
RPVT	Raw *p*-value thresholding (23)
scRNA-Seq	Single-cell RNA sequencing (22)
SNP	Single-nucleotide polymorphism (17)
SVM	Support vector machine (28)
T	Thymine (16)
TN	True negative (74)
TP	True positive (74)
TPM	Transcripts per million (22)
TPR	True-positive rate (74)
tRNA	Transfer ribonucleic acid (20)
t-SNE	t-Distributed stochastic neighbor embedding (26)
T1D	Type 1 diabetes (65)
T2D	Type 2 diabetes (65)
U	Uracil (19)
WTCCC	Wellcome Trust Case Control Consortium (21)
XAI	Explainable artificial intelligence (37)

2 Fundamentals

This chapter lays the foundation for the methodological innovations presented later on in this dissertation. Since the content of this work heavily relies on concepts from across several scientific disciplines, necessary background information is given here for all of those areas. After presenting detailed information on the fundamental biology of the genome to provide sufficient applicative context, we introduce the corresponding analysis methods that are traditionally used to explore genome-wide association studies and single-cell RNA sequencing datasets, which are under investigation in this dissertation. To validate the thesis proposed by the author in this dissertation in **Chapter 1.1**, these conventional methods are combined with sophisticated artificial intelligence techniques in the course of this dissertation. Hence, this chapter's final section presents the related machine learning concepts, including classification, explanation and information transfer. This chapter contains parts of articles **A-C**[10,15,22] from **Chapter 1.4** on previously published work.

2.0 Notation of chapter 2

Symbol	Definition (page it is introduced on)
α_i	Optimization variables in the dual problem of an SVM (30)
α_{LRP}	Parameter of the $\alpha\beta$ -LRP rule for positive contributions (39)
α_{NMF}	Penalty multiplier of the elastic net in NMF (41)
β_{LRP}	Parameter of the $\alpha\beta$ -LRP rule for negative contributions (39)
b	Bias term in linear predictive functions or in propagation functions of an NN (28)
C	SVM regularization parameter (29)
χ^2	Chi-square test statistic (23)
d	Number of dimensions in a dataset for SVM training (28)
$d_{1,\ldots,D}$	Number of selected eigenvectors in SC3 clustering (26)
e	Index of an outputclass in an NN (34)
E	Number of output nodes in an NN (34)
ϵ	Parameter of the ϵ -LRP rule (39)
eps	Number of epochs for NN Training (36)
ξ_i	Slack variables in an SVM (29)
$f(.)$	Predictive function in ML (27)
g	Number of dimensions in a source dataset for NMF (40)
$g(.)$	Propagation or pre-activation function in an NN (33)
γ	Parameter of the LRP-γ rule (39)
$h(.)$	Activation function in an NN (33)
h_t^q	Output of neuron t at layer q in an NN (34)
H	Dictionary in NMF (40)
H^*	Initial starting point of H in NMF (41)
i	Index of a datapoint in ML (28)
k	Number of clusters to find with a clustering algorithm (24)
$k(.,.)$	Kernel function (31)
λ_{NMF}	Parameter of the elastic net in NMF controlling L1 and L2 regularization (41)
n	Number of datapoints in a dataset for SVM training (39) or NMF (40)
nn	Number of neurons per dense hidden layer (36)
η	Learning rate of an NN (36)
p	Index of a predecessor layer of layer q in an NN (37)
q	Index of successor layer of layer p in an NN (33)
Q	Output layer of an NN (34)
ϱ	Parameter of *gFWER* (23)

$\varphi(\,.\,)$	Shape of the activation function in an NN (33)
φ	Dropout rate in an NN (35)
$R_s^{(p,i)}$	Relevance score of neuron s in layer p for sample i (37)
s	Index of predecessor neuron at layer p of neuron t at layer q in an NN (33)
$sgn(\,.\,)$	Sign function (30)
t	Index of successor neuron at layer q of neuron s at layer p in an NN (33)
τ	L1 norm regularization parameter in a loss function of an NN (36)
$vec(\,.\,)$	Vectorization of a given matrix (41)
υ	L2 norm regularization parameter in a loss function of an NN (36)
w	Weight vector in linear predictive functions (28)
w_{st}^q	Weight of the connection from neuron s at layer p to neuron t at layer q (33)
W	Reconstruction data matrix in NMF (40)
W^*	Initial starting point of W in NMF (41)
$x,\ x_i$	Observed input data in ML (27)
x_{new}	Unseen input data in ML (27)
$y,\ y_i$	Observed label to be predicted in ML (27)
$\widehat{y}_i$	Prediction score of a datapoint x_i in an NN (36)
y_{new}	Label of unseen data in ML (27)
$\|\|\,.\,\|\|_1$	L1 Manhattan Norm (36)
$\|\,.\,\|_2$	L2 Euclidean norm (29)
$\|\|\,.\,\|\|_{Fro}$	Frobenius norm (41)
$\langle\,.\,,\,.\,\rangle$	Dot product (31)
$(\,.\,)^+$	Positive value selector (39)
$(\,.\,)^-$	Negative value selector (39)

2.1 Biological background

This section introduces the fundamental biological concepts that play an important role in this dissertation's applications. Background information on the human genome and the central processes surrounding its translation into biological function is provided and important terms are defined.

2.1.1 The human genome

The human **genome** is a long **deoxyribonucleic acid** (**DNA**) chain created by stringing together thousands of its four unique basic units called **nucleotides**.[1] Each nucleotide consists of a five-carbon sugar (deoxyribose in the case of DNA), a phosphate group and a nitrogenous base. The latter determines the type of a nucleotide and can be either adenine, cytosine, guanine or thymine, which are usually abbreviated with the letters A, C, G and T in sequential genomic datasets. Genetic information is captured in the linear code of long sequences of single DNA strands. Together with a second strand, where complementary nucleotides bind together in **base pairs** (**bp**) via hydrogen bonds (A binds with T and C binds with G), DNA chains are stored in double helix structures and only unwound and opened when access to the retained information is necessary. See **Figure 1** for a graphical representation of DNA, its double helix structure and the chemical binding of base pairs.

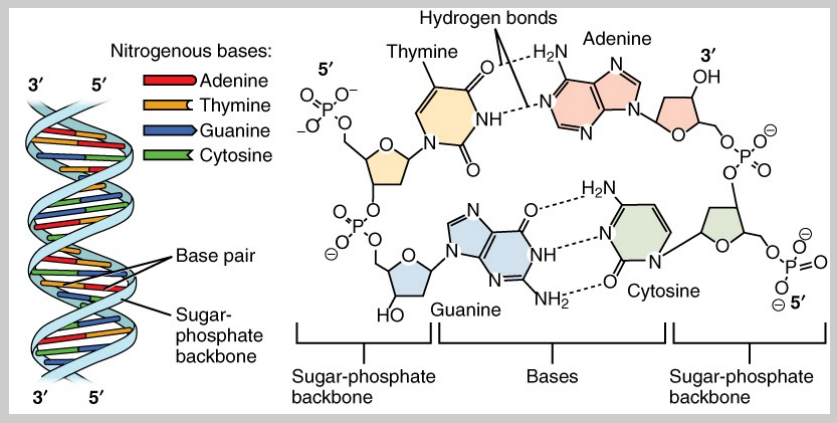

Figure 1: DNA double helix structure, nucleotides and base pairing. A DNA chain is a strand of thousands of nucleotides (A, C, G, T), consisting of a five-carbon sugar, a phosphate group and a nitrogenous base. Together with a second strand, where complementary nucleotides bind together in base pairs (bp) via hydrogen bonds (two between A and T, three between C and G), DNA chains are stored in double helix structures.

Image "DNA Nucleotides" by OpenStax College licensed under the Creative Commons Attribution 3.0 Unported license. Source: https://commons.wikimedia.org/wiki/File:0322_DNA_Nucleotides.jpg

The human genome comprises approximately 3.2 billion bp and is split up into 23 strands of DNA, called **chromosomes**.[1] For notation, the autosomes have been numbered according to their lengths ranging from the longest sequence of around 120 million bp on chromosome 1 and around 25 million bp on chromosome 22. The 23rd chromosome is the allosome containing genetic information to determine the sex of the

individual.[1] All human cells (except for reproductive cells) contain two copies of each chromosome, where one originates from the biological mother and one from the biological father. The genetic information on these chromosome pairs determines an individual's **genotype**, which is described at each position as a pair of two letters (A, C, G and T) - one from each chromosomal copy.

The human genome consists of several thousand **genes**, which are specific subsequences of DNA, encode protein sequences and represent the basic physical and functional unit of heredity.[1] Besides these coding regions, the genome contains noncoding regions, which perform other crucial functions such as gene regulation.

DNA sequencing refers to the technical procedure of determining the order of nucleotides in a genetic sequence and has allowed extraordinary progress in the biological and medical sciences[2]. In 2003 the Human Genome Project - a large consortium of research groups and scientists - published their result of over ten years of research, which had cost almost three billion dollars: They had sequenced the entire human genome for the first time and mapped all of its genes to specific positions on the sequence[3,4]. However, the mystery of the genome and its biological function remained somewhat unsolved since the published sequence was only an exemplary chimeric genome sequence composed of three human genomes of Asian, European and African ancestry. The DNA sequence differs significantly between any two individuals. More and more genomes had to be sequenced in order to identify how much and why genomes differ from each other. Nowadays, sequencing a single human genome can be done in less than 24 hours (*e.g.* with the Illumina NovaSeq 6000 platform) for less than 1,000 dollars[2,5]. Consequently, large-scale genetic datasets of millions of DNA sequences have been collected and are available for analysis to gain biological insights[2,6]. Challenges no longer lie in the sequencing process itself but in appropriately storing, handling, analyzing and interpreting these big datasets. It was found that a typical genome differs from the reference human genome at four to five million locations[27] (*i.e.* loci) widely distributed in the genome. Large structural variations, chromosomal duplications or chromosomal rearrangements can lead to differences between the genome of two individuals. When the genetic information of an individual differs from the consensus sequence at a specific position, a **mutation** has occurred at some point in the individuals' ancestry, which is typically caused by errors during the natural process of copying DNA. A specific substitution of a single nucleotide at a specific locus, as shown in **Figure 2**, is called a genetic variant or **single-nucleotide polymorphism** (**SNP**) if it appears in at least 1% of the human population.[1] When determining an individual's genetic information at these specific positions, three possible genotypes can occur. In the first case, both chromosomal copies are identical and match the consensus sequence, *i.e.* carry the so-called major allele: The individual is said to be **homozygous** of the wild-type allele for this locus in the genome. The second option is that the subject is **heterozygous** and carries one wild-type (major) and one mutant-type (minor) allele, which appears only in the minority of the population.

Finally, the subject could even have two copies of the minor allele and is hence homozygous for that. The described variations enable diversity in the population because the corresponding genes are translated into proteins, which amongst other influences determine key characteristics of the cell and, ultimately, the organism[1].

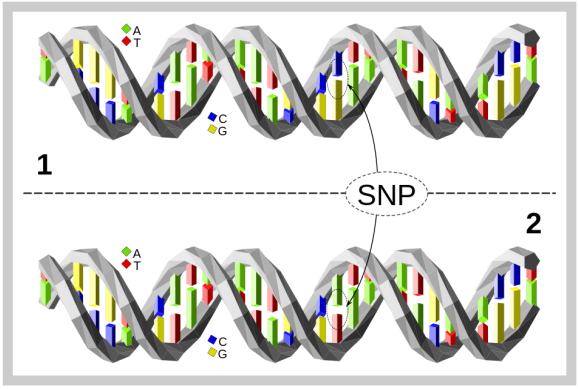

Figure 2: Single-nucleotide polymorphism. A single-nucleotide polymorphism (SNP) is a specific substitution of a single nucleotide at a specific locus in a DNA strand. In this example, individual 1 has the wild-type allele C at the highlighted position and individual 2 has the mutant-type allele A.

A SNP is called informative and statistically associated when linked to the development of a trait under study. These observable characteristic traits of an organism (or cell) are called **phenotypes** and are mostly not only determined by the corresponding genotypes but also influenced by environmental factors[28]. When investigating susceptibility to diseases, a subject's phenotype is usually either determined to be healthy or ill. In this context, it is important to note that the **penetrance** of a genetic variation, *i.e.* the proportion of people who carry a specific genetic mutation (*i.e.* the risk allele) and also carry the corresponding phenotypic trait, can vary significantly from phenotype to phenotype. For example, it is very high for genetic disorders like cystic fibrosis[29] and can be rather low for chronic diseases like rheumatoid arthritis[30]. Since genetic factors often play an important role in the risk of developing a disease, many studies focus on hereditary aspects of certain diseases. In this context, it is important to note that genes and SNPs are not passed down to successors independently of each other and the tendency of two genes to be inherited together is called **genetic linkage**[31]. It has been shown that the closer two loci are physically together on a chromosome, the more likely they are to be inherited together simply because of the physical link between the two[32]. Beyond that, there is an effect called **linkage disequilibrium** (**LD**), indicating an increased statistical association between allelic variants that are not necessarily physically linked[33]. LD can be caused by selective pressures on these loci in a genomic region (*i.e.* when certain combinations of alleles reduce reproductive success in any way) and in the case of long-range LD by epistatic interactions (when a mutation at one position changes the local environment of another position either by directly contacting it or by causing changes in the corresponding protein structure[34]).

2.1.2 From DNA to protein - the central dogma of molecular biology

As mentioned in the previous section, genes are the basic physical and functional unit of heredity and are translated into proteins, which determine important characteristics and functionalities of the organism. Here, we describe the natural processes around the transfer and conversion of genetic information.

The **central dogma of molecular biology** was first introduced by Francis Crick in 1958[35] and describes the flow of genetic information within a biological system. There are three biological molecules that can contain sequential information: DNA, as it was described above, **ribonucleic acid** (**RNA**) - a polymeric molecule almost identical to DNA but with a ribose instead of deoxy sugar in the backbone of the chain and the nucleobase uracil (U) instead of thymine (T) - and **protein** - which is also a biopolymer consisting of a sequence of **amino acids**.

Crick grouped the different directions that genetic information could be transferred from one of these molecules to another into three classes according to their likeliness to appear in natural or artificial circumstances. The most common pathways in living cells are from DNA (the structure the information is stored in) to DNA as part of the replicative cycle, from DNA to RNA (the intermediate transport molecule) and from RNA to protein (the final product and functional unit of the pathway). Information transfer from protein to protein or back to nucleic acids is stated to be impossible[35]. The third group of pathways (from RNA to RNA or DNA and from DNA directly to protein) are extremely rare - so-called special - transfers that can occur but only under specific conditions in the case of some viruses or in a laboratory[36]. In this dissertation, however, we focus on the general pathways that happen most frequently in natural cells and describe them in more detail in the following sections.

When a cell is duplicated, the DNA it contains needs to be copied to provide a complete genome for both offspring cells. This process is called **replication** and consists of several substeps that are performed by specialized functional proteins[37]. At first, the DNA helix structure is unwound and the hydrogen bonds between complementary nucleotides are opened up by an enzyme called helicase. Subsequently, a primer molecule is bound to both DNA strands to start the replication process at the correct starting positions. Complementary nucleotides are now added to both of the original DNA strands by the enzyme DNA polymerase III, creating two new identical DNA double strands, which are eventually separated and brought back into helix structure to be stored in the nuclei of the two newly emerging offspring cells.[1]

When a gene is activated during the natural life cycle of a cell, the sequential information of its DNA is copied into **messenger RNA** (**mRNA**), which is subsequently used as a template to build a sequence of amino acids that eventually form a protein

(shown in **Figure 3**). The first step of transforming DNA into mRNA is called transcription and the second step of transforming mRNA into protein is called translation.

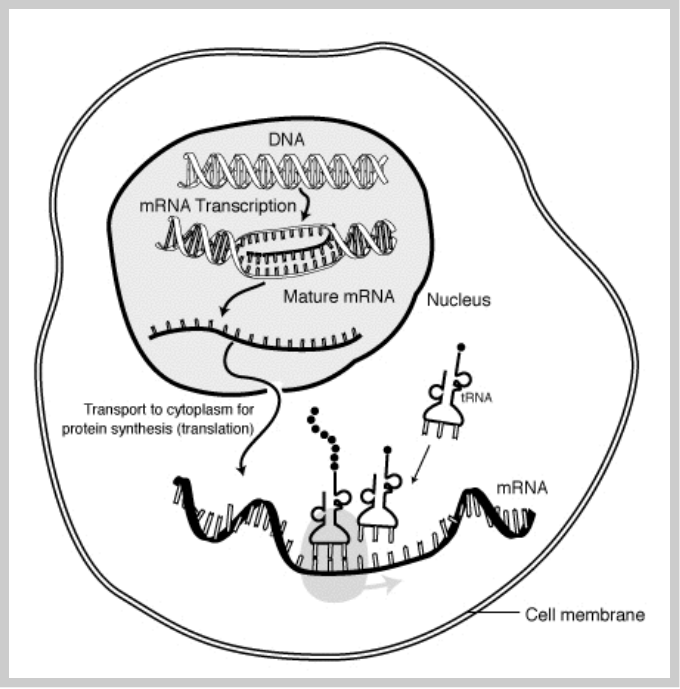

Figure 3: The flow of genetic information from DNA to mRNA to protein. When a gene is activated, its DNA sequence is first converted into pre-mRNA during transcription, which is further transformed into mature mRNA. During translation and after transport out of the nucleus, the mRNA chain is converted into a sequence of amino acids to eventually create a protein.

During **transcription**, proteins called transcription factors identify specific promoter regions on the genome and indicate the starting position for the transcription process.[1] The DNA double helix is unwound and an enzyme called RNA polymerase progresses along the template strand to synthesize a complementary RNA strand. When it reaches a termination sequence, the precursor mRNA molecule (pre-mRNA) is detached and further processed to add a specific cap and tail to both ends of the molecule. In an additional step called alternative splicing, different subsequences are removed or combined in eukaryotic cells to increase the number of proteins a single mRNA sequence can produce. The created mature mRNA is transported out of the cell nucleus - where DNA is usually stored - into the cytoplasm - where translation takes place.

The **translation** process begins with the mRNA binding to a ribosome, which moves along the mRNA sequence and synthesizes a specific amino acid for each set of three consecutive nucleotides called triplet codons.[1] The process of attaching new nucleotides to a growing polypeptide chain begins at a defined start codon (AUG) and ends at one of three stop codons (UAA, UGA, or UAG). All other codons are matched to an anticodon of a **transfer RNA** (**tRNA**) molecule, which carries the corresponding amino

acid. The chain of amino acids immediately folds into the correct conformation, but further post-translational modifications are often necessary to build a functional protein.

Many mechanisms exist that control the production of mRNA and protein in specific cells and at specific points in time[1]. For example, noncoding regions in the DNA serve important regulatory functions. Other mechanisms by which the same genotype can lead to various phenotypes include changes in transcription rate, mRNA turnover, chromosome accessibility, promoter activity, enhancer access and antisense mRNAs[1].

2.1.3 Study types under investigation in this dissertation

In the course of this dissertation, novel methodologies for two different types of biological studies, *i.e.* genome-wide association studies and single-cell RNA sequencing experiments, are introduced. An overview of the general ideas of these two study types is presented here. More detailed information is given in the problem setting sections of the corresponding chapters (**Chapter 3.2.1** and **Chapter 4.2.1**).

Genome-wide association studies

Genome-wide Association Studies (**GWAS**) are observational studies that explore the meaning of genetic sequences by investigating the phenotypic effects of SNPs. The goal is to examine the relationship between SNPs and individual traits, which are usually complex major diseases, behavioral characteristics or anthropometric traits. The genotypes of a large set of SNPs are sequenced for a group of patients with a disease (or trait) and compared to the genotypes of a group of healthy controls (without a trait). To identify locations in the sequence that are associated with the disease (or trait), tables of genotype counts are created and the corresponding *p*-values are compared against a multiple testing significance threshold, which usually lies between 1×10^{-8} for rigorous studies and 1×10^{-5} to report weaker associations as well[38]. The first GWAS was published in 2002[39–41] and several years later, a landmark study - the largest GWAS ever conducted at the time of its publication in 2007 - was presented by the **Wellcome Trust Case Control Consortium** (**WTCCC**)[38], sequencing the genotypic data of over 500,000 SNPs and including 14,000 cases of seven common diseases and 3,000 shared controls. The corresponding dataset is used in **Chapter 3** of this dissertation to evaluate the performance of the proposed methods. Since then, sample sizes, rates of discovery and numbers of studied traits have been rising continuously[42]. A repository for collecting the results of such studies is the **GWAS catalog**[7] which includes almost 5,000 studies and more than 250,000 SNP-phenotype associations with *p*-values below 1×10^{-5} (accessed on April 3. 2021). Especially for common human diseases such as diabetes, autoimmune disorders or psychiatric illnesses, GWAS have provided valuable insight into the corresponding genetic inheritance processes[43,44]. A few studies have included over 1 million subjects enabling the identification of SNPs with lower risks and frequencies[45,46].

Single-cell RNA sequencing studies

Single-cell RNA sequencing (**scRNA-Seq**) was introduced in 2009[19] and enables the analysis of the **transcriptome** (*i.e.* the entirety of mRNA molecules) of singular cells at a certain point in time. Previously it was only possible to look at pooled transcriptomes for multiple cells in microarray or bulk RNA sequencing experiments, where differences between individual cells were lost. During most common scRNA-seq protocols, the cells of a sample taken from the tissue under investigation are isolated. To determine the number of different mRNA molecules in each cell, one of multiple available protocols is executed[47]. Most of them include variations of the following substeps: mRNA extraction, reverse transcription for conversion of mRNA to complementary DNA (cDNA), DNA amplification, sequencing and library generation. Eventually, the sequenced fragments are aligned to reference genomes. As a result, a count table is obtained, where each transcript is assigned a number referring to one of various expression units such as raw read counts, reads per million (RPM), counts per million (CPM), transcripts per million (TPM) or reads per kilobase per million mapped reads (RPKM). Numerous scRNA-Seq protocols are available[19,48], which, for example, implement alternative ways of reverse transcription, cDNA synthesis or amplification.

With improved technical possibilities and increased sample sizes, scRNA-Seq has been applied successfully in many research areas[20,49–52] and the progress of ScRNA-Seq in the field of embryo development was named the "*2018 Breakthrough of the Year*" by Science[53].

The most relevant research questions about scRNA-Seq datasets concern either the identification of gene expression patterns through gene clustering analyses[54] or the determination of cell types through cell clustering into groups based on their transcriptomes[21]. In this dissertation, we focus on the latter and employ unsupervised clustering approaches to interpret the genome in the context of cell types. Even though all (somatic) cells contain the same genetic material, there is a huge variety of cell types, functions and tissues in the human body[1]. The genome is implemented into function differently, expressing only specific pieces of the genetic sequence in different combinations. Via numerous types of regulation, the transcriptomes of cells differ significantly not only between cells but also for one cell at different points in time[1]. When investigating the transcriptome of individual cells in scRNA-Seq studies, we can cluster cells according to which genes are (currently) activated in those cells. In contrast to bulk sequencing, scRNA-Seq allows identifying and examining rare cell types, *e.g.* highly specialized lung cells[55,56]. The granularity of assessing individual cellular transcriptomics has highlighted the vast heterogeneity in cell types previously believed to be relatively homogeneous[47]. When adding tissue context, diversity increases dramatically. Due to the relatively high cost of scRNA-Seq studies and the difficulty of accessing rare tissues, many of these datasets are small but high-dimensional and only include poor representations of rare subtypes of cells[21].

2.2 Traditional analysis methods

In the following section, we describe the traditional methods used to analyze the two types of studies that are relevant in this dissertation and are described in **Chapter 2.1.3**. Multiple hypothesis testing for GWAS and different clustering approaches for scRNA-Seq data are introduced and important competitor approaches are discussed.

2.2.1 Multiple statistical hypothesis testing for GWAS

Traditional approaches for the analysis of GWAS can be categorized into two main categories: While some methods focus on phenotypic risk prediction based on the given genetic information[57–61], others try to explain these risk effects by highlighting which SNPs are having an effect on a given trait[62–66]. Generally, a GWAS investigates the observed genotypes of a set of SNPs (represented by the number of minor alleles) and a group of subjects labeled with the corresponding phenotypes separating controls from cases. The classic approach - which we refer to as **raw *p*-value thresholding** (**RPVT**) - consists of carrying out a statistical association test to assign a *p*-value to each individual SNP. The null hypothesis of this single-locus test is that there is no difference between the trait means of any genotype group, which indicates that the genotype at a specific SNP is independent of the phenotype under investigation[67]. Via a χ^2 test, RPVT calculates a *p*-value for each SNP and declares it significantly associated with the phenotype if it is smaller than a predefined threshold[43,68,69]. This threshold has to be chosen carefully as the significance level in the case of a single test. Since generally, a large number of statistical tests are performed in parallel, the threshold has to be adjusted for multiple testing to bound, for example, the **expected number of false rejections** (**ENFR**), the **family-wise error rate** (**FWER**), *i.e.* the probability of at least one false-positive test result, or the **generalized FWER** (**gFWER**), i.e. the probability of at least $\varrho \geq 1$ false-positive test results. Some standard methods for choosing the threshold for the purpose of controlling multiple type I error rates are reviewed in **Chapter 3.2.1**. Bonferroni correction is the most straightforward way to take multiplicity into account by dividing the significance level by the number of conducted tests[70].

When the *p*-values of all SNPs have been calculated and the corresponding threshold was defined, a so-called **Manhattan plot** can be created[71]. An exemplary Manhattan plot is shown in **Figure 4**. For each genomic position of the corresponding SNP on the *x*-axis, it displays the negative logarithmic *p*-value on the *y*-axis. SNPs that are strongly associated with the disease or trait appear as high values in the plot and are commonly referred to as towers. If they exceed the predefined significance threshold - which is usually represented by a horizontal line in those plots - they are considered to be statistically associated with the trait under investigation.

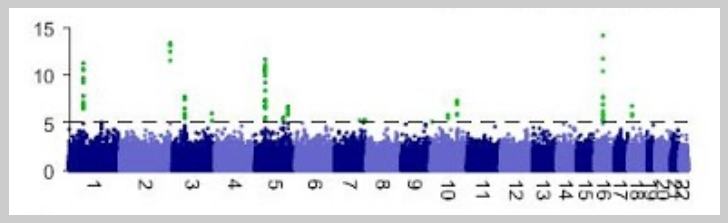

Figure 4: Exemplary RPVT Manhattan plot. The negative logarithmic χ^2 test *p*-values of the Crohn's Disease WTCCC dataset are plotted against position on each chromosome. The threshold indicating statistical significance is represented by the dashed horizontal line and significant *p*-values are highlighted in green.

The problem setting for RPVT and the analysis of GWAS, in general, is described in full detail, including formula and notation in **Chapter 3.2.1** and **Chapter 3.2.5**.

2.2.2 Clustering methods for scRNA-Seq studies

Whilst the analysis of scRNA-Seq data has many challenges, including normalization[72,73], dealing with noise[74], zero inflation and missing values[75,76], dimensionality reduction[75,77] and visualization[78,79], one of the key analytical techniques to address questions of cell type identification is that of unsupervised clustering. **Unsupervised algorithms** learn from using only input data (in this case, representing RNA counts) without the knowledge of any outcome variables (in this case, referring to the underlying true cell categories). **Clustering** of cells into discrete groupings according to their transcriptional state is the fundamental analysis required in many scRNA-Seq experiments. A range of approaches has been taken to address the problem of clustering scRNA-Seq data, including hierarchical and iterative clustering[80,81], principal component analysis based approaches[82,83], ensemble clustering[81,84] and graph-based approaches[85–89]. As the number of cells in scRNA-Seq datasets increases, the development of machine learning-based[90,91] and **specifically deep learning-based**[92–95] clustering approaches has expanded.

One traditional and simple unsupervised approach to clustering cells in scRNA-Seq datasets is to apply ***k*-means clustering**[96,97] and group similar datapoints (*i.e.* individual cells) together in a fixed number of *k* clusters (*i.e.* cell types). These clusters are defined by a point in the center of each cluster, called centroid, which is calculated by averaging all datapoints belonging to that cluster. The aim of *k*-means clustering is now to assign each datapoint to the closest centroid and minimize the variance within a cluster (*i.e.* the within-cluster sums of squares). According to the law of total variance, this is equivalent to minimizing the pairwise squared distances of points in one cluster, which is again equivalent to maximizing the sum of squared distances between points in different clusters (*i.e.* the between-cluster sum of squares)[98].

The most common method that implements *k*-means clustering - called ***k*-means algorithm** or **Lloyd's algorithm** after his creator Stuart P. Lloyd - was developed in the 1950s and follows an iterative approach[97]. At first, the *k*-means algorithm identifies an initial group of *k* centroids either by selecting a random set of *k* datapoints or by randomly assigning each datapoint to a cluster and calculating the initial centroids by averaging all datapoints belonging to those initial clusters. Now, each datapoint is assigned to the nearest initial cluster centroid in terms of squared distance and the new *k* centroids referring to those updated allocations are calculated. This process of alternating between the expectation step of assigning every datapoint to the cluster with the nearest centroid and the maximization step of updating the centroids of each cluster is repeated until convergence (*i.e.* the cluster assignments and centroids do not change anymore) or the limit of iterations is reached. The *k*-means algorithm is not guaranteed to find the global optimum and might instead converge to a local optimum[99] since the result of the algorithm depends heavily on the initial clusters and running the algorithm multiple times results in different solutions. Adjusted *k*-means algorithms exist that escape local optima[99].

The *k*-means algorithm can be adjusted to different problems through the application of different **distance functions**. The most commonly used measures are the Euclidean distance (*i.e.* the classical length of a line between two datapoints)[100], the Pearson correlation (*i.e.* the covariance of two datapoints divided by the product of their standard deviations)[101] and the Spearman correlation (*i.e.* the Pearson correlation between the rank of two datapoints)[102]. Another way to adapt to different problems is to apply data transformation techniques to the distance matrices before running the clustering algorithm. For example, with a **principal component analysis** (**PCA**)[103,104] - which is often used for visualization of high-dimensional datasets - a linear change of basis can be performed to reduce the number of features while preserving most of the variance in the data. A **Laplacian graph Eigen decomposition**[105] is a similar dimensionality reduction algorithm that - in contrast to PCA - allows for the data to lie on a nonlinear manifold.

To combine the different solutions into a consensus clustering and harness the full potential of all *k*-means variations for scRNA-Seq data, a combinatorial approach called **SC3** was developed in 2017[84]. SC3 is a popular method - commonly used to solve unsupervised scRNA-Seq clustering problems - that is based on the combination of multiple *k*-means clustering solutions based on different distance metrics and transformations to determine a consensus clustering. The basic steps of the SC3 algorithm are shown in **Figure 5**. Given an expression matrix, SC3 first applies a gene filter and log-transforms the data. Then, three different cell distance matrices are calculated using Euclidean, Pearson and Spearman metrics, respectively. The three distance matrices are then transformed by applying both PCA and Laplacian graph Eigen decomposition. Subsequently, *k*-means clustering is performed *D* times on the

first $d_{1,\ldots,D}$ eigenvectors of the resulting six matrices, where $d_{1,\ldots,D}$ comes from a predefined range of values. The clustering results are now combined by applying the Cluster-based Similarity Partitioning Algorithm[106] to compute a consensus matrix. Hierarchical clustering[107] is finally used to cluster the resulting matrix into k clusters. SC3 is not deterministic and produces different results when solving the same clustering problem multiple times.

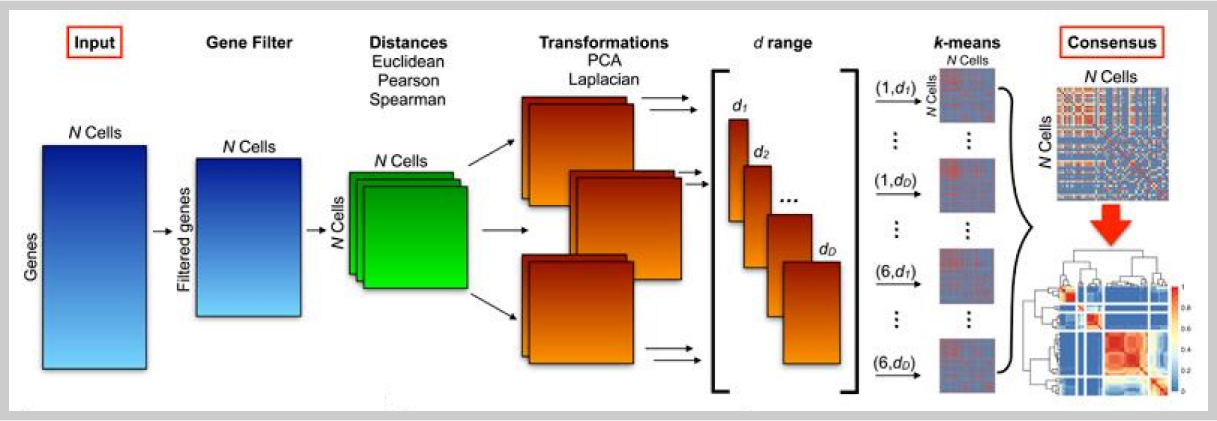

Figure 5: The SC3 framework by Kiselev *et al.* 2017[84]. SC3 is a consensus clustering approach for scRNA-Seq data that combines a number of different distance measures, transformation techniques and clustering methods to eventually determine a consensus cluster assignment for the cells in a given dataset.

In this dissertation, we use SC3[84] as an exemplary clustering algorithm whose performance can be improved through the application of transfer learning (see **Chapter 4**).

For visualizing the clustering results of scRNA-Seq data, we can, for example, employ PCA[103,104] or **t-distributed stochastic neighbor embedding** (**t-SNE**)[79] as a nonlinear dimensionality reduction technique, which projects the points from higher-dimensional space to lower-dimensional space while at the same time trying to preserve the local neighborhoods of each point.

2.3 Machine learning concepts

In the following sections, we introduce the basic machine learning concepts that are of importance in the methodological proposals of this dissertation. We start with a section about the general ideas of machine learning and go on to present the specific learning models of support vector machines and deep neural networks to which we refer to in **Chapter 3**. Afterward, we introduce the field of explainable artificial intelligence and layer-wise relevance propagation as an exemplary explanation method, which is also utilized in **Chapter 3**. Finally, we present the methodology of non-negative matrix factorization and the concept of transfer learning, which is employed in **Chapter 4**.

2.3.1 Machine learning basics

Artificial intelligence (**AI**) is a scientific research area that concerns various forms of intelligence demonstrated by machines. As one of its branches, **machine learning** (**ML**) is a generic term first coined in 1959 by Arthur Samuel for the artificial generation of knowledge based on experience and observation[108]. The goal in ML is to determine, based on a sample called training data and without being explicitly programmed to do so, a predictive statistical model, which does not only represent the examples at hand but rather detects patterns and laws that enable future prediction for unseen data. ML techniques have seen a huge success in the field of data analysis and have been applied with flying colors in countless different data science problems, *e.g.* automatic translation[109], speech and text recognition[110], physical property prediction[111,112], automated patient diagnosis[113], credit card fraud[114], email spam detection[115], stock market analysis[116] and autonomous driving[117].

In ML, a **predictive function** $f(x)$ is learned that predicts the class label y (*e.g.* a phenotype) based on the observation of the corresponding data x (*e.g.* the genotype). It is crucial to require such a function to not only capture the sample at hand but to also **generalize**, as well as possible, to new and unseen measurements[118], *i.e.* to ensure that the sign of $f(x)$ is a good predictor for y for previously unseen patterns x_{new} and labels y_{new}. A model is said to be **overfitting** when it fails to generalize well to unseen data and only represents the sample at hand[118].

ML algorithms can be divided into two groups depending on the availability of true class labels in the training dataset. **Unsupervised learning** refers to techniques that describe an input dataset by discovering categories and patterns without any previous knowledge on the corresponding class labels[119]. Clustering methods like *k*-means, as described in **Chapter 2.2.2**, assign each training datapoint to one of multiple groups that the machine considers to be a good partitioning of the data.

During **supervised learning**, the machine is presented with training data that consists of pairs of input data and expected output labels enabling the algorithm to learn the association between input and output variables[120].

Depending on the assumptions made about the mapping function *f*, we distinguish between linear and nonlinear learning models. **Linear methods** assume that the input variables affect the outcome variables only through a linear relationship and usually have the form $f_{w,b}(x) = w^T x + b$, where w is called a **weight vector** and b is a **bias term**. The prediction functions of **nonlinear methods** can be much more complex and are used when the data under investigation is not linearly separable.

Please refer to **Chapter 2.2.2**. for an exemplary unsupervised clustering algorithm (*k*-means), **Chapter 2.3.2** for an exemplary linear supervised learning algorithm (*i.e.* a support vector machine) and **Chapter 2.3.3** for an exemplary supervised nonlinear algorithm (*i.e.* a neural network).

2.3.2 Support vector machines

A popular approach to learning a supervised linear model as described above is training a **support vector machine** (**SVM**)[11–13]. Here, a linear model $f_{w,b}(x) = w^T x + b$ is learned that predicts the class labels y (*e.g.* the phenotypes) based on the observation of the corresponding data x (*e.g.* the genotypes).

In this dissertation, binary classification is considered, where a training set consists of n d-dimensional datapoints and their corresponding labels, *i.e.* $x = (x_i)$ and $y = (y_i)$ with $i = 1, ..., n$, $x_i \in \Re^d$, $d, n \in \aleph$ and $y_i \in \{+1, -1\}$. The idea behind an SVM is then to find a hyperplane in the feature space $\Re^d$ that linearly separates the two groups of datapoints from each other. Additionally, the goal of a reliable classifier is to maximize the smallest distance between any two points of opposite classes. This distance is called margin and can also be defined as the distance between two lines parallel to the hyperplane that pass through the two points closest to the hyperplane. See **Figure 6** for a visual representation in two-dimensional space.

The hyperplane is defined by b, the intercept term, the inner product of the data x and the weight vector w. Then, $w^T x + b$ corresponds to the aforementioned linear classification model $f_{w,b}(x)$. In order to solve this equation for the w and b of an optimal hyperplane, a canonical hyperplane is created by normalizing its parameters to the observed data with

$$y_i (w^T x_i + b) \geq 1 \; \forall \; i = 1, ..., n .$$

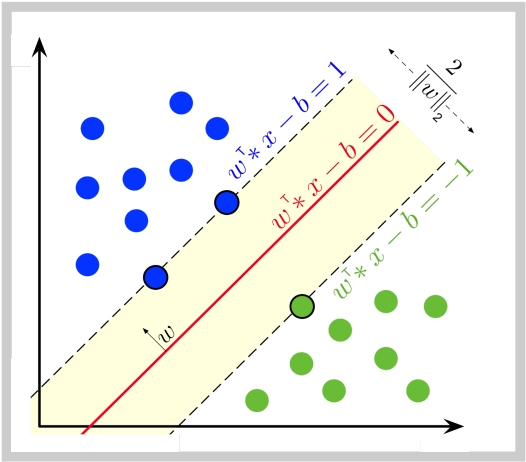

Figure 6: A support vector machine in two-dimensional space. A support vector machine (SVM) aims to identify a hyperplane that linearly separates the two groups of blue points with positive labels and green points with negative labels from each other. Hence, the smallest distance between any two points of opposite groups, *i.e.* the margin, is maximized. The hyperplane is highlighted in red and the corresponding maximum-hard-margin is shown in yellow. The datapoints on the dashed lines defining the margin are called support vectors.

In practical applications, where outliers or random noise occur, complete linear separation of classes - called **hard margin classification** - is often not possible. The concept of **soft margin classification**[13] introduces slack variables $\xi_i \geq 0 \; \forall \; i = 1, \; ..., \; n$ allowing some datapoints to lie within the margin as follows:

$$y_i \; (w^T x_i + b) \; \geq 1 - \xi_i \; \forall \; i = 1, \; ..., \; n \, .$$

Now, since the margin is equal to $\frac{2}{\|w\|_2}$, where $\|w\|_2$ is the euclidean norm of the weight vector w, maximizing the margin corresponds to minimizing $\|w\|_2$, which can be achieved by solving the following optimization problem for $C>0$:

$$min_{w,b,\xi} \left(\tfrac{1}{2} \; ||w||_2^2 \; + \; C \sum_{i=1}^{n} \xi_i \right)$$

$$s.t. \;\; y_i \; (w^T x_i + b) \; \geq 1 - \xi_i \; \forall \; i = 1, \; ..., \; n$$

$$\xi_i \geq 0 \; \forall \; i = 1, \; ..., \; n$$

This is known as the **primal problem** and can be rewritten as

$$min_w \left(\|w\|_2^2 \; + C \sum_{i=1}^{n} max \, (0, 1 - y_i w^T x_i) \right) .$$

It is similar to regression problems and can be interpreted as follows: we aim to minimize the trade-off between a vector w with a small norm (the term on the left-hand

side) and small errors on the data (the term on the right-hand side). The trade-off is controlled by the **regularization parameter** C, which has to be chosen carefully[11,121,122]. While high values of C cause high penalties for datapoints that fall on the wrong side of the hyperplane and may lead to overfitting, small values of C might prevent the algorithm from capturing the underlying pattern in the data. Once a classification function f has been determined by solving the above optimization problem, it can be used to predict the label y_{new} of any unseen datapoint x_{new} by putting

$$y_{new} = sgn\,(f\,(x_{new})) = sgn\,(w^T x_{new})\,.$$

A different version of the above optimization problem can be generated based on the fact that the weight vector w can be written as a linear combination of training examples as follows:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

The α_i can be interpreted as the contribution of the i-th datapoint to the final solution w. With that knowledge, the above primal problem can be reformulated as the following **dual problem**[123]:

$$argmax_{\alpha} \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$s.t. \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C \quad \forall\, i = 1,\ ...,\ n$$

After optimization, a new datapoint x_{new} can be classified with

$$y_{new} = sgn\,(f\,(x_{new})) = sgn\left(\sum_{i=1}^{n} \alpha_i y_i \langle x_i, x_{new} \rangle + b\right).$$

The dual is equivalent to the primal problem in such a way that solutions to the dual can be directly converted into solutions to the primal and vice versa. Since the primal is a quadratic program with $d+n+1$ optimization variables ($w \in \Re^d$, $\xi \in \Re^n$ and b) and the dual is a quadratic program with $n+1$ optimization variables ($\alpha \in \Re^n$ and b), it is always advantageous to solve the dual problem when $d \gg n$. This is the case for the data application presented in **Chapter 3**, where the number of SNPs is much higher than the number of subjects in a study. In the same manner, if d is large, it is a lot more expensive to classify an unseen datapoint x_{new} by explicitly computing the scalar product $w^T x_{new}$ from the primal problem than to calculate $\sum_{i=1}^{n} \alpha_i y_i \langle x_i, x_{new} \rangle + b$ from the dual problem.

There is another important scenario, where it is of great advantage to solve the dual instead of the primal problem. When the underlying classification problem is not linear and it is impossible to find a linear hyperplane in feature space that separates the classes well, the data is projected into a higher-order space, where linear separation is possible. The challenges for this approach lie in very high computational cost for the necessary data transformations and the usually highly complex presentation of the hyperplane back in lower-dimensional space. At this point, a technique commonly referred to as the **kernel trick**[11,124] is applied. Instead of explicitly performing the forward and reverse transformations to and from the higher-dimensional space, suitable kernel functions $k(x_i, x_j)$ are calculated for all pairs of datapoints ($\langle x_i, x_j \rangle$ in the case of linear kernels) that describe the hyperplane in high-dimensional space well and remain manageable in low-dimensional space. Since the prediction of a class label in the dual problem contains a scalar product only involving *data* vectors (*i.e.* $\sum_{i=1}^{n} \alpha_i y_i \langle x_i, x_{new} \rangle + b$), it is possible to apply the kernel trick in this setting.

2.3.3 Neural networks

Artificial neural networks (NN) are a powerful tool for learning nonlinear relationships between an input and an output variable by transferring information through "*a computing system made up of a number of simple, highly interconnected processing elements*"[125]. NNs have seen an unprecedented rise in data science[126] and created enormous progress in numerous fields, *e.g.* image classification[127,128], natural language processing[129], speech recognition[130] and quantum chemistry[111,112]. NNs mimic the behavior of natural biological neurons in a nervous system of an organism. However, instead of only recreating the biological neural networks, the aim of NNs in ML is to enable powerful modeling of information processing in areas where no direct modeling laws are given. Through the interaction of numerous simple parts of the same type, NNs can create extremely complex behavior and are therefore another example for the validity of the famous saying that "*the whole is greater than the sum of its parts*"[26] (derived from Aristotle, 4th century BC), which undeniably plays an important role in this dissertation. Although NNs are often referred to as novel modern ML techniques, historically, they have been around for more than 70 years. In 1943 Warren McCulloch and Walter Pitts were the first to describe the connection of elementary units as a type of network - similar to the network of neurons in the nervous system - that could practically learn any logical or arithmetic function[131]. In the fifties, the first computational machines and the perceptron[132] (the most simple NN consisting of a single artificial neuron) were created to implement such learning algorithms and the scientific field of AI was born. NNs found their way into practical application shortly after. However, following a number of publications revealing the boundaries of perceptrons (*e.g.* with linear separability and exclusive-or-operands[133]) and a lack of sufficient computing power, funding was limited and research in AI temporarily lost its

momentum in the late sixties. It took many years of slow progress and improving technology to increase computing power until AI finally came back to the forefront of data science. In 1985 some of the initially claimed boundaries concerning linear separability were refuted with the use of multi-layer perceptrons and the technique of backpropagating error[134]. In the 1990s and 2000s, numerous novel AI methods were developed[11,135,136]. When GPUs, distributed computing and the vast amount of available data in the 2010s allowed the use of large multi-layer networks, called **deep neural networks (DNN)**, the performance of the corresponding **deep learning** approaches increased significantly[137,138]. In recent years AI methods have taken over data science by storm, winning important international competitions[139] and providing outstanding performances on benchmark datasets[128]. An advantage of (D)NNs over traditional ML methods is that they can often be applied as end-to-end learning approaches from the sampled datasets to the desired results[140], avoiding any hand-crafted intermediary algorithms, such as feature engineering, which are required for most traditional ML methods.

In the following paragraphs, we describe the basic topology of NNs and their various components. See **Figure 7** for a graphical representation of the interconnected elementary subunits of an NN - called **artificial neurons -** and a simple NN, where - according to common practice - neurons are represented by nodes and interneuronal connections by edges.

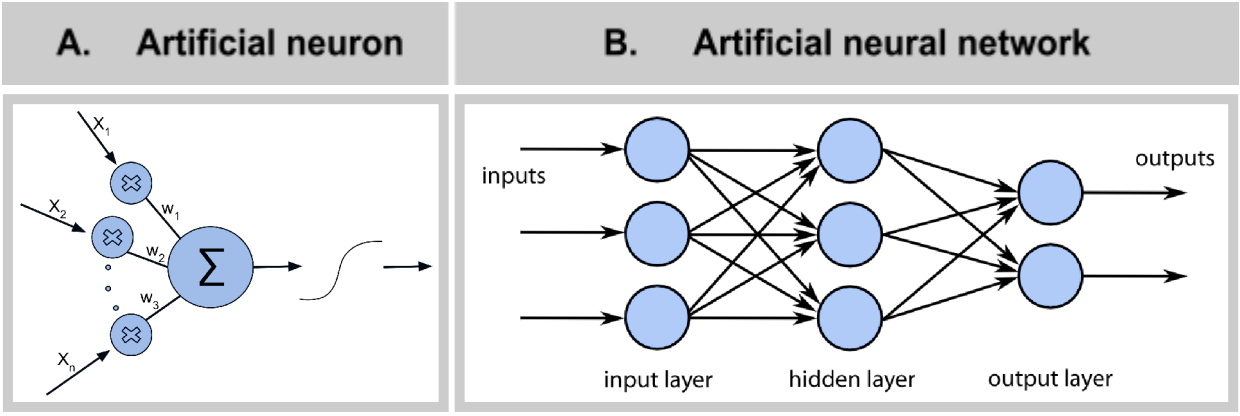

Figure 7: Graphical representation of an artificial neuron and a simple artificial neural network. A The artificial neuron weights and sums up input information it receives from predecessor neurons through a propagation function. The result, *i.e.* the pre-activation, is subsequently processed through a nonlinear activation function and the corresponding output is the input to any successor neurons. **B** Multiple artificial neurons are connected and form a simple dense artificial neural network (NN) with an input layer, one hidden layer and an output layer.

Artificial neurons are mathematical functions that act as computational units, which perform specified calculations based on the information they receive from other neurons they are connected to[132]. At neuron *t*, all dimensions *s* of its input *x* (either from an input

datapoint or a previous neuron) are firstly weighted and summed up by a **propagation function** *g(x)* to produce a pre-activation output

$$g(x) = \sum_s w_{st}^q x_s + b_t ,$$

where w_{st}^q is the weight of the connection between node *s* at layer *q-1* and node *t* at layer *q* and b_t is the bias term of node *t*. The pre-activation *g(x)* is subsequently processed through a predefined (often nonlinear) output **activation function** $h(x) = \varphi(g(x))$ to transform the output into a certain range.

There are various choices for the shape of φ (see **Figure 8**), which has a huge effect on the functionality of the network.

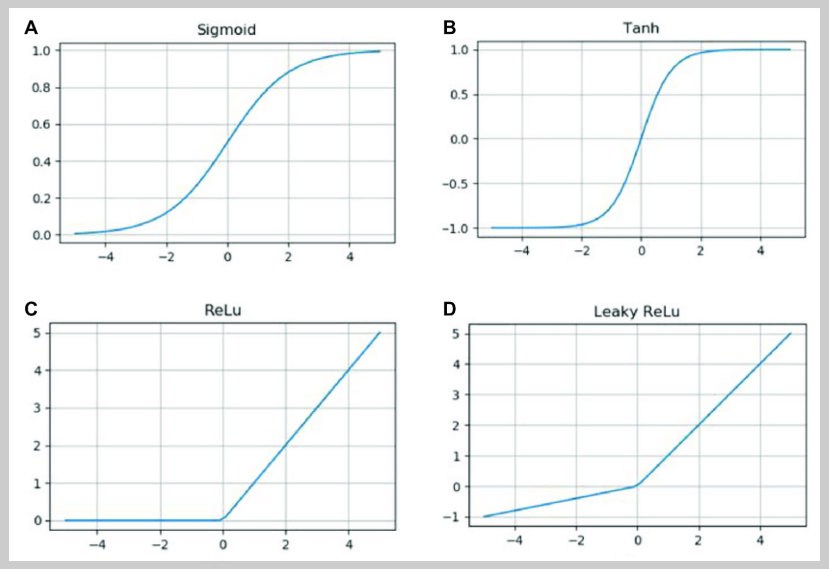

Figure 8: Various activation functions. Displayed are the sigmoid (**A),** the tanh **(B),** the ReLU (**C**) and the leaky ReLU (**D**) activation functions.

When nonlinear data is being modeled, nonlinear activation functions provide the means to successfully capture those structures in the data. A popular (and often the default) choice of φ is the **rectified linear unit** (**ReLU**) **function** $\varphi(x) = max(0, x)$, which sets all negative values in the input to 0. It is often preferable to complicated activation functions like the **sigmoid** or the **tanh** function because it is so simple and reduces running times[142]. In addition, the ReLU function produces sparse results, which increase the ability of a network to learn meaningful information and reduce overfitting. It also reduces the risk of creating a phenomenon called the "*vanishing gradient problem*", which can occur when applying activation functions that are only sensitive around the origin (*e.g.* the tanh or the sigmoid function)[143]. During training (which

includes the gradient descent approach) and because the derivatives of these functions are flat in the tails, the gradients keep decreasing with increasing number of layers and oftentimes, the initial layers are not able to learn properly. In contrast to this behavior, ReLU functions stay sensitive at all stages[142]. However, when the input gets very large, they might cause exploding gradients, which cause the model to be unstable[144]. Another problem arises when neurons are stuck within the negative values and fail to recover because of the hard "*below zero rule*" of the ReLU function. These issues can be resolved by employing lower learning rates (i.e smaller steps in the learning process) or a less stringent activation function called **leaky ReLU**[144] $\varphi(x) = max(0.01 * x, x)$.

The **topology** of an artificial NN describes its internal structure, which is mostly determined by the number of artificial neurons on a specified number of layers and the way they are connected to each other. The simplest NN is a **perceptron**, a single-layer network with one artificial neuron. The more neurons and layers are added, the higher the depth of the network. While the first layer of an NN represents the input data and is referred to as the **input layer**, the last layer is referred to as the **output layer** and represents the outcome variables of the problem under investigation[125]. All **intermediate layers** are usually not accessible to the user and are therefore called **hidden layers**. All NNs with more than one hidden layer are considered to be deep and are therefore called DNNs.

The most common type of an intermediate layer is a **fully connected** or **dense layer**, where neurons are connected to all neurons of the previous layer[125]. We define $h(x) = h_t^q$ as the output of neuron t at layer q, which is computed based on the outputs of all neurons s of the preceding layer q-1, the weights w_{st}^q connecting those neurons to the current neuron t and the bias term b_t in the following way:

$$h_t^q = \varphi\left(\sum_s w_{st}^q h_s^{q-1} + b_t\right).$$

The output layer Q at the end of an NN is usually a fully connected layer that consists of E output nodes, where E is the number of potential outcome classes. To perform the classification task, it takes the input values h_e^Q for each class e and converts them into a probability that the input data belongs to class e. This is achieved by applying the **softmax output activation function**

$$p(y = e|h_e^Q) = \frac{exp\left(h_e^Q\right)}{\sum_{c=1}^{E} exp\left(h_c^Q\right)}$$

which converts the vector of input numbers into a vector of probabilities that are proportional to the relative scale of the corresponding value in the vector. These class probabilities naturally sum up to 1.

A regularization strategy to avoid overfitting is to implement **dropout layers**, where a predefined proportion (called dropout rate φ) of randomly selected neurons are turned off and not passed on to the next layer[145].

Networks, where each layer is only connected to subsequent layers towards the output layer, are called **feedforward networks**. **Recurrent networks** allow connections (*i.e.* feedback loops) in both directions of the network[146]. In this dissertation, only feedforward networks are applied. **Convolutional neural networks** (**CNN**) have at least one convolutional layer, where a so-called kernel or filter moves across the image to check whether a specific feature is present[147]. CNNs provide superior performance, especially in image recognition, by extracting features from the input image while spatial relationships between pixels are preserved[128] but are not utilized in this dissertation.

After determining the architecture of an NN, defining all hyperparameters, initializing the weights randomly to small values and setting the biases to 0, the **training** process is started. The underlying data structure is now learned by assigning output values to specific input patterns and repeatedly improving the performance of the model. In **supervised learning**, this is achieved through adjusting the weights w_{st}^{q} from neuron s to neuron t at layer q in an iterative process of passing information from the given datapoints through the network towards the desired output variables and backpropagating information on the classification performance to update the weights of the network[148–151]. The following four-step training process is called an **iteration** of the NN training and is repeated for each datapoint until the whole dataset is passed through the network at least once.[148–151]

1. **Forward propagation:** A training datapoint is passed through the network, starting at the input layer and going through all intermediate layers, where all neurons process the incoming data with the activation function and pass it on to all successive neurons. When the information has reached the final layer, outcome probabilities are assigned to each class.

2. **Error calculation:** A predefined loss function is evaluated, comparing the current output probabilities with the desired class label of the datapoint under investigation.

3. **Backward propagation:** Starting from the output layer, the calculated error term is processed backwards through the network by calculating the gradients/derivatives of all parameters.

4. **Optimization:** With the activations (from step 1) and the gradients (from step 3), the weights and biases of the network are updated to minimize the error / loss function (from step 2) using stochastic gradient descent[152].

A suitable loss function has to be chosen before training. In this dissertation, classic **cross-entropy loss** is applied[153]. To enable good generalization to unseen samples and avoid overfitting despite the large number of model parameters, the binary cross-entropy loss is coupled with an **L1-L2 mixed regularization** term[154]:

$$loss = \sum_i (y_i * log(\widehat{y_i}) + (1 - y_i) * log(1 - \widehat{y_i})) + \tau * \sum_{q,t} \left|\left| w_{*t}^q \right|\right|_1 + \upsilon * \sum_{q,t} \left|\left| w_{*t}^q \right|\right|_2$$

with y_i being the ground-truth label of sample *i*, $\widehat{y_i}$ the corresponding predicted class, which depends on the learned parameters w_{st}^q of the NN and $\tau, \upsilon > 0$ the regularization parameters. This loss function contributes to prevent the network from overfitting by minimizing the trade-off between small errors on the data on the one hand and small L1 and L2 norms of the weights on the other hand[154].

Instead of passing a single datapoint through the network in each iteration, steps 1-3 can be repeated for a **batch** of points before going on to optimize the weights in step 4, accumulating errors over the entire set of points. Passing the training points of one batch through the network is then called an iteration and the number of points in one batch is referred to as the **batch size**. Randomly selecting batches of datapoints from the dataset can cause a faster and more stable descent to a local minimum but can also lead to memory problems.

The process of passing the entire training dataset through the NN exactly once (as a whole, for each datapoint separately or in batches) is called an **epoch**. If the number of epochs (*eps)* is increased, the NN is trained repeatedly, passing the same data through the network multiple times. The number of datapoints divided by the batch size gives the number of iterations required to finish one epoch.

The **learning rate** η is a hyperparameter of the model that represents the step size at each iteration when moving towards the minimum of the loss function in step 4 from above. In order to avoid undesired learning behavior inside the network, such as alternating connection weights and finding local minima, and to improve the rate of convergence, an adaptive learning rate can be used that increases or decreases η as appropriate. The easiest form entails a learning rate reduction with a specific factor after a predefined number of epochs of no improvement.

While the weights of an NN are called parameters and are learned through learning, **hyperparameters** are constant and fixed before training. Important hyperparameters that can have significant effects on the learning behavior of an NN are the number of hidden layers, the number of neurons in each layer (*nn*), the interconnectedness of the network, the regularization parameters τ and υ, the learning rate η of the network, the batch size, the number of epochs *eps* and the choice of activation and loss functions. These important properties and configurations of an NN, as well as the initial weights,

have to be chosen carefully, fitting the respective dataset under investigation. Oftentimes this can only be done in a trial and error approach, but sophisticated guesses can be made with the help of appropriate literature[14].

2.3.4 Explainable AI and layer-wise relevance propagation

Explainable AI (**XAI**) is a field of AI that has been gaining importance recently[155–158]. It refers to techniques, which open the so-called "*black box*" of ML methods and reveal the processes underlying their decisions so that the results can be better understood.

In XAI, we distinguish between **local** and **global explanations** depending on whether they provide sample-dependent explanations or generate explanations of the model as a whole. While some explanation methods depend on feature permutation[159], others provide local interpretable model-agnostic explanations[160], build generalized additive models[161] or compute class-specific gradient-based saliency maps[162]. A very popular method to generate local relevance scores from trained NNs is **layer-wise relevance propagation** (**LRP**)[16–18], which provides a way to compute feature importance scores, automatically taking correlation structures and possible interactions into account. The visual or computational investigation of heat maps generated based on LRP can reveal which features in the input datapoints are most relevant for the class prediction of the NN. LRP has been applied successfully in numerous data science problems to explain decisions of NNs[163,164].

The process of explaining the prediction of an NN through LRP consists of the following two steps:

1. After an NN f is trained on a prediction task, the prediction scores of a datapoint x_i are computed by $f(x_i) = \widehat{y_i}$, a forward pass through the network.
2. Following a specific backpropagation rule, a single output score, *i.e.* the highest output score, $\widehat{y_i}$ is backpropagated successively layer-by-layer through the network until reaching the input layer.

See **Figure 9** for a graphical representation of an NN trained on image data and subsequent explanation via LRP. When an NN is trained on a binary image classification task to separate cats from dogs , the relevance score $R_s^{(p,i)}$ of neuron s in layer p demonstrates to which extent this specific neuron affected the classification decision $f(x_i)$ into a cat or a dog[17]. Once the input layer is reached, the corresponding relevance scores represent the level of impact each input feature (*i.e.* pixel) had on the decision of the network for that specific image. Plotting the relevance scores as a heatmap on the corresponding pixels highlights the parts of the images that are most important for the classifier's decision. If x_i is indeed the image of a cat, LRP of a successful NN highlights the cat's ears and whiskers or other cat-specific features.

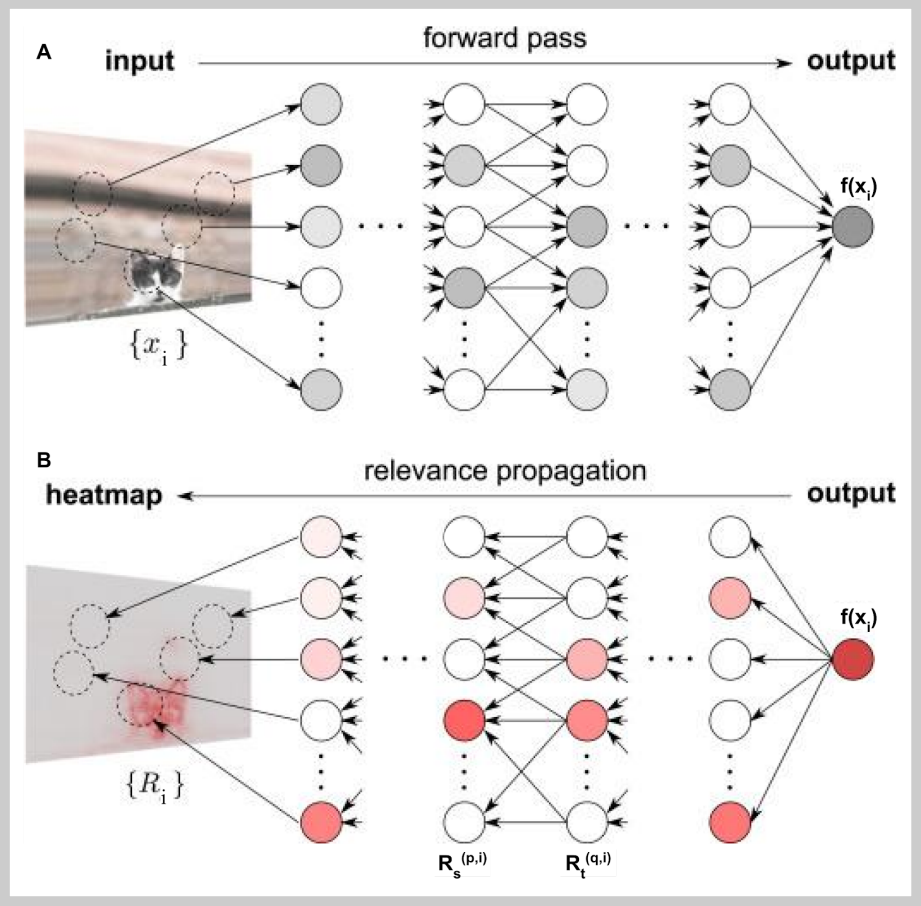

Figure 9: Image classification and subsequent pixel-wise explanation via layer-wise relevance propagation. A During explanation via layer-wise relevance propagation (LRP), the prediction score of an image is first calculated through a forward pass through the NN. **B** Following a backpropagation rule, the output score is then backpropagated through the network layer by layer until reaching the input layer, where a relevance heatmap is displayed. The image is classified as a cat and the LRP heatmap highlights cat-specific features like its ears and whiskers.

The backpropagation process begins with assuming that the relevance scores at the last layer equal the output of its activation functions. Thus, without loss of generality, it is assumed that the relevance distribution at layer q is known and the relevance distribution at the predecessor layer $q\text{-}1=p$ can be extracted. LRP aims to find the **relevance score** $R_s^{(p,i)}$ of neuron s at layer p based on the following equation, which indicates that no relevance is lost and the sum of relevances in all layers is equal to the output activations of the last layer:

$$f(x_i) = \ldots = \sum_{t \in (q)} R_t^{(q,i)} = \sum_{s \in (p)} R_s^{(p,i)} = \ldots = \sum_{r \in (1)} R_r^{(1,i)}$$

Additionally, all relevance scores are constrained to be positive (*i.e.,* $\forall i, p, s: \; R_s^{(p,i)} \geq 0$). Based on these constraints, the relevance of one neuron is distributed amongst its

predecessors following one of several **backpropagation rules**[18], some of which we present here. The **LRP-0 rule** simply redistributes relevance proportionally to the contribution of neuron *s* to its successors *t* in terms of weights and activation:

$$R_s^{(p,i)} = \sum_t \left(\frac{h_s^p w_{st}^q}{\sum_s h_s^p w_{st}^q}\right) \times R_t^{(q,i)} ,$$

where h_s^p denotes the activation of neuron *s* and w_{st}^q is the weight between the two neurons *s* and *t*.

When applying the **LRP-ϵ rule**, noise is removed by reducing weak or contradictory (*i.e.* alternating) activations:

$$R_s^{(p,i)} = \sum_t \left(\frac{h_s^p w_{st}^q}{\epsilon + \sum_s h_s^p w_{st}^q}\right) \times R_t^{(q,i)} .$$

The **LRP-Z rule** distributes the relevance of neurons only amongst predecessor neurons linked through positive weights, indicating that only features with a positive influence on the classification of a specific sample as the resulting outcome are highlighted. Features with the opposite effect of not supporting this particular choice of the network are ignored when applying this rule:

$$R_s^{(p,i)} = \sum_t \left(\frac{(h_s^p w_{st}^q)^+}{\sum_s (h_s^p w_{st}^q)^+}\right) \times R_t^{(q,i)} .$$

When applying the so-called **LRP-$\alpha\beta$ rule**, the relevance $R_s^{(p,i)}$ of neuron *s* in layer *p* depends on the relevance of all of its successors *t* in layer $p+1=q$ in the following way:

$$R_s^{(p,i)} = \sum_t \left(\alpha_{LRP} \frac{(h_s^p w_{st}^q)^+}{\sum_s (h_s^p w_{st}^q)^+} - \beta_{LRP} \frac{(h_s^p w_{st}^q)^-}{\sum_s (h_s^p w_{st}^q)^-}\right) \times R_t^{(q,i)} ,$$

This rule allows us to weigh the positive and negative contributions ($\left(h_s^p w_{st}^q\right)^+$ and $\left(h_s^p w_{st}^q\right)^-$, respectively) of neurons *s* to their predecessor *t* differently by the hyperparameters α_{LRP} and β_{LRP} .

The **LRP-γ rule** controls how much positive evidence is favored over negative evidence with its parameter γ. This is particularly helpful to smooth out relevances close to the input layer, where patterns are already captured.

$$R_s^{(p,i)} = \sum_t \left(\frac{h_s^p (w_{st}^q + \gamma w_{st}^{q^+})}{\sum_s h_s^p (w_{st}^q + \gamma w_{st}^{q^+})}\right) \times R_t^{(q,i)}$$

Depending on the network architecture and other data-specific factors different backpropagation rules are to be preferred.

2.3.5 Non-negative matrix factorization

In recent times technological possibilities are improving in many fields, causing very large datasets to be available for analysis. Hence, to enable further processing, it is often necessary to reduce the dimensions of such datasets without a loss of crucial information. When the corresponding data is limited to positive values (*e.g.* for physical reasons) and expected to represent linear combinations of recurring patterns with added noise, methods referred to **non-negative matrix factorization** (**NMF**)[23–25] can be applied. Under the constraint of non-negativity, NMF aims to identify a list of such basic recurring patterns and the corresponding set of coefficients to recreate the dataset under investigation. If both of these entities are unknown, the solution is non-convex and approximate solutions have to be determined numerically in an iterative process. The result provides a lower-dimensional representation of the original dataset as a linear combination of basic patterns. The concept of NMF was first introduced in the 1990s[25] and became more popular in the early 2000s[23,24], when it started to be successfully employed in various fields such as text clustering[165], image imputation[166] or information filtering[167]. Applications that are relevant for this dissertation lay in bioinformatics, where NMF was able to improve performances, for example, in protein alignment[168], microarray analysis[169], various omics tasks[170] or clustering gene expression data[171].
During NMF of a **dataset** $X \in \Re^{g \times n}$ of n datapoints in g-dimensional space, the set of recurring patterns $H \in \Re^{g \times k}$ called **dictionary** and the corresponding coefficients to reconstruct the **data matrix** $W \in \Re^{k \times n}$ are learned so that $HW \approx X$. See **Figure 10** for a schematic representation of the underlying approach.

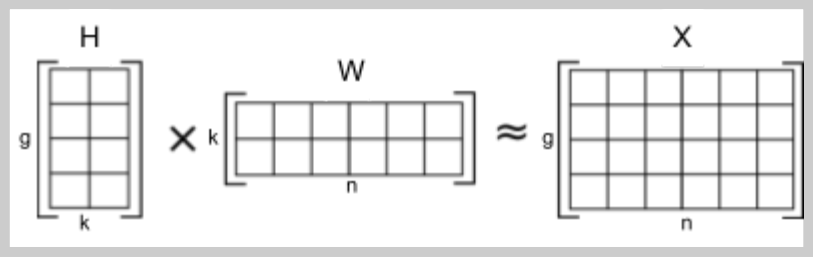

Figure 10: Non-negative matrix factorization. A data matrix $X \in \Re^{g \times n}$ of n datapoints in g-dimensional space is approximately factorized into a set of recurring patterns, *i.e.* the dictionary $H \in \Re^{g \times k}$ and the corresponding coefficients $W \in \Re^{k \times n}$ to reconstruct the data matrix.

A number of different optimization functions for measuring the error between the dataset and its factorization exist that can also include regularization terms for the two learned matrices. Here, we employ a function based on the squared error (or Frobenius norm) and regularize the denseness of the results with an elastic net[154]:

$$\widehat{H}, \widehat{W} = argmin_{W,H} \left(\frac{1}{2} ||X - HW||_{Fro}^2 + \alpha_{NMF}\lambda_{NMF} \left(||vec(H)||_1 + ||vec(W)||_1 \right) + \frac{\alpha_{NMF}}{2} (1 - \lambda_{NMF}) \left(||H||_{Fro}^2 + ||W||_{Fro}^2 \right) \right),$$

where λ_{NMF} is the elastic net mixing parameter controlling the combination of L1 and L2 (Frobenius) regularization and α_{NMF} is the corresponding penalty multiplier. Please note that without loss of generality, X is sometimes defined to have dimension $n \times g$ and NMF is used to learn $WH \approx X$, where the data matrix W and the dictionary H have reversed dimensions.

If any prior knowledge is available, an initial starting point W^* for W or H^* for H can be provided to the expectation-maximization algorithm that solves this equation by alternately optimizing H and W. A number of different solvers can be applied to the above equation, the multiplicative update rule[24] being one of the most popular options. In this dissertation, an optimization algorithm called coordinate descent[172] is used, which successively minimizes the function along each coordinate while keeping all other directions constant.

Naturally, NMF can be viewed as a form of clustering[23,24], where the cluster memberships are selected based on the column-wise maxima of W, *i.e.* $\widehat{y}_i = argmax_{l \in \{1, \ldots, k\}} (W)_{li}$.

There are a number of hyperparameters in NMF that need specification before learning, *e.g.* the number of patterns or clusters k in the dictionary, the elastic net parameters α_{NMF} and λ_{NMF} and the maximum number of iterations until convergence up to a specified level of relative error.

2.3.6 Transfer learning

In many different areas of scientific research, datasets are constrained by the scarcity, feasibility and expense of collecting samples. In such scenarios, it is often not possible to apply methodologies like deep learning, which require large and well-annotated datasets. To address this common problem, the ML concept of **transfer learning** can be employed to integrate *a priori* knowledge from large reference datasets into smaller datasets to generate additional insights[173,174]. To be able to use previously learned knowledge can not only help in situations of data scarcity, but it can also be an advantage in terms of efficiency, computing power and time when large datasets are analyzed. In general, transfer learning refers to applying knowledge gained from one context to another - distinct but related - context in a setting where the solution of one or multiple source tasks is applied to a related target task. The term and corresponding

mathematical ideas in relation to NNs were first introduced in 1976[175] and further developed in the following years. Thrun contributed to the emergence of the field in 1996 by asking if "*learning the n-th thing [is] any easier than learning the first?*"[176], which was motivated by findings in human psychology. One of the key insights was that humans build upon related concepts when learning new tasks, which Thrun coined *lifelong learning*. Another influential line, which popped up around the same time, introduced the term **multitask learning**[136]. Instead of learning a sequence of related tasks, multiple related tasks are learned in parallel using a shared representation. In combination, transfer learning is an umbrella term for problems[173] such as multitask learning, domain adaptation and covariate shift[177], which have been applied in many fields, including cognitive science[178], EEG data analysis[179], web search[180], spam detection[181] and speech and text recognition[182].

Transfer learning algorithms can be based on very different approaches but always have in common that they aim to help improve the learning of a target task by using knowledge previously gained in learning a source task. Some exemplary methods include the application of hierarchical bayesian models[183], Markov logic networks[184], kernel methods[185] or deep CNNs[186].

3 Combining machine learning with multiple statistical hypothesis testing for genome-wide association studies

In this chapter, we propose two novel methods that combine advanced machine learning algorithms (from **Chapter 2.3.1**, **2.3.2**, **2.3.3** and **2.3.4**) with traditional multiple statistical hypothesis testing (from **Chapter 2.2.1**) to identify genotype-phenotype associations in GWAS (as described in **Chapter 2.1.3**). Given a GWAS dataset, the proposed methods improve the identification of disease trait associations by training an appropriate state-of-the-art classification algorithm, selecting a subset of candidate locations, which are most relevant for the classifier's decisions and examining only those SNPs for significant associations via multiple statistical hypothesis testing. In support of the author's proposed thesis of this dissertation, these combinatorial approaches help to better understand the genetic architectures of different traits and diseases and therefore examine the translation of genetic code into biological function.
This chapter is based on and contains parts of articles **A**[10] and **C**[15] from **Chapter 1.4** on previously published work. It also builds upon work published in a master thesis written by Alexandre Rozier and supervised by the author of this dissertation[187].

3.0 Notation of chapter 3

Symbol	Definition (page it is introduced on)
α	Significance level to bound type 1 error rates (54)
α_{LRP}	Parameter of the $\alpha\beta$ -LRP rule for positive contributions (39 and 61)
b	Bias term in an SVM (28 and 55)
B	Number of Monte Carlo repetitions in the permutation test (58)
β_{LRP}	Parameter of the $\alpha\beta$ -LRP rule for negative contributions (39 and 61)
c	Case-control status of a subject (53)
C	SVM regularization parameter (29 and 56)
χ^2	Chi-square test statistic (54)
d	Number of SNPs under investigation in a GWAS (53)
eps	Number of epochs for NN Training (36 and 68)
$E_{c,g}$	Expected frequencies of genotype g in combination with case-control status c (72)
$f(\,.\,)$	Predictive function (27 and 55)
g	Genotype group of a subject (53)
γ	Effect size parameter for generation of synthetic GWAS data (64)
H_j	Null hypothesis of equal p_1 and p_2 at SNP j (54)
i	Index of a subject in a GWAS (53)
j	Index of a SNP in a GWAS (53)
ϑ	Parameter vector of the statistical model of a GWAS with $\vartheta = \left(\vartheta_j\right)_{j=1,\ldots,d}$ (54)
ϑ_j	Pair of probability vectors $\vartheta_j = (p_1^{(j)}, p_2^{(j)})$ of SNP j (54)
k	Number of SNPs to select in the COMBI and the DeepCOMBI method (55)
K_j	Two-sided alternative of unequal p_1 and p_2 at SNP j (54)
l_{filter}	Filter length or window size of a p-th-order moving average filter (56)
n	Number of subjects in a GWAS (53)
nn	Number of neurons per dense hidden layer (36 and 68)
n_{cg}	Numbers of individuals in the group of case-control status c and genotype g (53)
η	Learning rate of an NN (36 and 68)
$O_{c,g}$	Observed frequencies of genotype g in combination with case-control status c (72)
p_{filter}	Norm parameter of a p-th-order moving average filter (56)
p_1 , p_2	Probability vectors of cases $p_1 = (p_{11}, p_{12}, p_{13})^T$ and controls $p_2 = (p_{21}, p_{22}, p_{23})^T$ at a specific SNP (54)
p_{cg}	Probability of an individual having case-control status c and genotype g at a specific SNP (54)
p_j	p-value of SNP j corresponding to testing H_j (54)

P_0	Probability distribution under the global null hypothesis of no informative SNPs (58)
p_{min}	The smallest of the k p-values of the selected positions (58)
P_ϑ	Unknown true data-generating distribution (58)
q	Index of a successor layer of layer p in an NN (33 and 61)
ϱ	Parameter of *gFWER* (23 and 59)
φ	Dropout rate in an NN (35 and 61)
r	Global relevance scores in the input layer with $r = \left(r_j\right)$, where the r_j are averaged over all subjects (60)
$R^{(0,i)}$	Relevance scores of the input layer for sample i (61)
R^2	Squared correlation coefficients of SNPs in LD (66)
ρ_{ij}	Relevance score of subject i and SNP j with $\rho_{ij} = \left(\sum_u R_u^{(0,i)}\right)/3$, where u are the indices of the three features that one-hot-encode the corresponding genotype (62)
s	Index of predecessor neuron of neuron t in an NN (33 and 61)
$sgn(\,.\,)$	Sign function (56)
t	Index of successor neuron of neuron s in an NN (33 and 61)
t^*	Multiple testing significance threshold (48)
τ	L1 norm regularization parameter in the loss function of an NN (36 and 68)
υ	L2 norm regularization parameter in the loss function of an NN (36 and 68)
w	Weight vector of SVM (28 and 55)
$w^*(k)$	k-th largest value in w in absolute value (57)
w_{st}^q	Weight of the connection from neuron s to neuron t at layer q in an NN (33 and 61)
x	Observed genotypes of a set of SNPs and a set of subjects with $x = \left(x_{ij}\right)$ (53)
x_{ij}	Genetic information of subject i in SNP j with $x_{ij} \in \{(1,0,0),(0,1,0),(0,0,1)\}$ (53)
x_{new}	Genotype of an unseen subject (56)
κ	p-value threshold to preselect SNPs for NN training (61)
y	Observed phenotypes of a set of subjects with $y = (y_i)$ (53)
y_i	Label of subject i with $y_i \in \{+1, -1\}$ or $y_i \in \{0,1\}$ (53)
$\widehat{y}_i$	Prediction scores of a datapoint x_i (61)
y_{new}	Phenotype of an unseen subject (56)
$y_{\pi(1)}, \ldots, y_{\pi(n)}$	Random permutation of the phenotypes y (58)
$\|\,.\,\|_2$	L2 Euclidean norm (56)

3.1 Introduction

The goal of GWAS (as described in **Chapter 2.1.3**) is to examine the relationship between small genetic variations called SNPs and individual traits, which are usually complex diseases or behavioral characteristics. The approaches to the analysis of GWAS either investigate phenotypic risk prediction[57–61] or the explanation of the corresponding risk effects by determining the set of SNPs are associated with the trait[62–66]. This dissertation aims at both of these goals but puts focus on the latter by using ML-based prediction methods in combination with statistical testing to identify SNPs associated with the phenotype under investigation.

Please refer to **Chapters 2.1.3** and **2.2.1** for an introduction to GWAS and the corresponding traditional analysis methods.

In order to analyze GWAS datasets, a huge number of statistical tests are performed in parallel, each SNP being *individually* tested for association[43,68,69]. The traditional approach - referred to as RPVT and introduced in **Chapter 2.2.1** - consists of carrying out individual statistical association tests for all SNPs and comparing the resulting p-values to a predefined significance threshold t^*. Per definition, precisely those SNPs with p-values smaller than t^* are declared to be associated with the trait[43,44]. We review some standard methods for choosing t^* for the purpose of controlling multiple type I error rates (in particular, *FWER* and the *ENFR*) in **Chapter 3.2.1**. Following developments in biotechnology, GWAS have developed into a powerful tool that provides valuable insights into the genetic causes of several important phenotypes. Especially for common diseases, GWAS have improved our understanding of the underlying genetic inheritance processes[43,44]. However, variants reported by GWAS tend to explain only small fractions of individual traits and the genetic architectures and variances of most traits and complex diseases remain largely unexplained. This phenomenon - often referred to as "*the mystery of missing heritability*" - is assumed to be - at least partially - caused by the way GWAS datasets are traditionally analyzed[8,9]. RPVT is based on testing SNPs individually and in parallel, which intrinsically ignores any potential epistatic interactions[9,188] between or correlation structures among the set of SNPs under investigation[68,69]. Studies fail to identify multi-locus effects by using the traditional RPVT approaches and a large amount of potentially available information is lost[189]. Only very few diseases rely on single genetic defects with large effects, while most complex diseases are caused by epistatic interactions of multiple genetic factors with small effects. Further influence originates from correlation structures due to both population genetics (*i.e.* LD) and biological relations (*i.e.* functional relationships between genes)[42]. The latter issue by itself is likely to introduce confounding factors and artifacts, implying a loss in statistical power[189] and a lack of reliable insights about genotype-phenotype associations. Brute force multivariate approaches to identify the aforementioned dependencies are oftentimes computationally too expensive for large

GWAS datasets and are limited by low statistical power due to excessive multiple testing. A few attempts have been made to identify genetic interactions, but most of them are not able to find strong, statistically significant associations[188,190–192].

To overcome these limitations of traditional approaches and following the rise of ML in data science and the increase in the amount of available large-scale GWAS datasets, a number of methods have been proposed to introduce ML tools for the analysis of such studies. Linear approaches such as multivariate logistic regression and sparse penalized methods, including Lasso, have been applied to GWAS datasets. In general, penalized models achieve better performances than non-penalized methods[60,193–195]. Nonlinear models, such as random forests, gradient boosted trees and bayesian models[60,193,196,197] investigate interactions and correlations in the genetic architecture of traits but are mostly found to be outperformed by linear penalized methods[60,190,193,196].

To harness even more sophisticated nonlinear ML methods for the analysis of GWAS, attention has recently been drawn to NNs, which we have introduced in **Chapter 2.3.3**. NNs have been applied to the analysis of GWAS datasets[198,199], but most of the corresponding publications focus on risk prediction[193,200–202] and only very few methods have been proposed for identifying SNP disease associations[193,203].

Romagnoni *et al.*[193] present a thorough comparison of conventional statistical approaches, traditional ML-based techniques and state-of-the-art deep learning-based methods in terms of both prediction rates and the identification of SNP associations on a Crohn's Disease immunochip dataset. Classification performances of numerous methods (Lasso as a reference, penalized logistic regression, gradient boosted trees, NNs) are compared and found to be similar for most methods (linear and nonlinear) implicating potentially "*limited epistatic effects in the genetic architecture*"[193]. However, when investigating the associated genetic regions identified by the different methods, ML and deep learning-based methods are indeed found to provide new insights into the genetic architecture of the trait. Romagnoni *et al.* achieve this by applying the concept of XAI, which we have introduced in **Chapter 2.3.4** of this dissertation as a fast-moving field of AI that has emerged recently[155–158]. The explanation method used by Romagnoni *et al.* - permutation feature importance (PFI)[159] - is a generalized, model-agnostic approach and more sophisticated methods specifically tailored to NNs are available.

In this dissertation, we aim to validate the proposed thesis that it is preferable to develop methods that combine the advantages of traditional and novel technologies and approaches. With this in view, we propose two combinatorial approaches to couple multiple hypothesis testing with ML-based methods for the analysis of GWAS. Some models have previously been introduced for GWAS that combine statistical testing and ML for identifying SNP disease associations[62,63]. However, most of these methods do not provide validation on real data comparing to the GWAS database and very few

provide a full evaluation of identified genetic variants in terms of comparison to previously published GWAS.

The core idea of this work is to develop principled, reliable and replicable methods for identifying significant SNP-phenotype associations that can be described as two-step algorithms consisting of

1. an ML and SNP selection step that drastically reduces the number of candidate SNPs by selecting only a small subset of the most predictive SNPs; and
2. a statistical testing step, where only the SNPs selected in step 1 are tested for association.

In this dissertation, we propose two novel methods - called COMBI and DeepCOMBI - that implement this general idea in different ways. **Figure 11** shows graphical representations of both approaches.

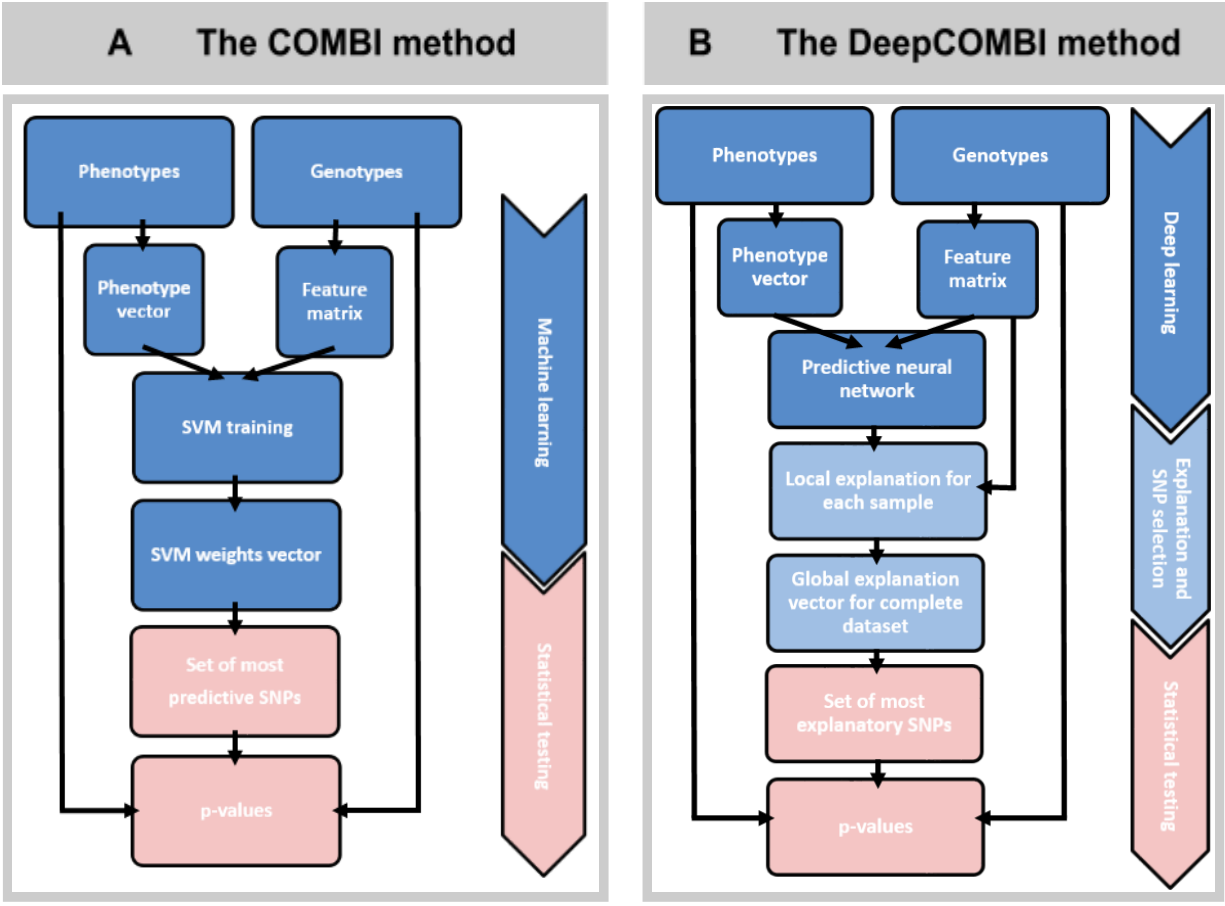

Figure 11: Graphical representations of the COMBI and the DeepCOMBI method: A The COMBI method. Receiving genotypes and corresponding phenotypes of a GWAS as input, the COMBI method first trains an SVM to select a set of most predictive candidate SNPs and then calculates *p*-values and corresponding significance thresholds in a statistical testing step. **B The DeepCOMBI method.** Receiving genotypes and phenotypes of a GWAS as input, the DeepCOMBI method first applies a deep learning step to train a DNN to classify subjects. Afterwards, in the explanation step, it selects the most relevant SNPs by applying LRP to calculate relevance scores for each SNP. Finally, for this set of most relevant SNPs, DeepCOMBI calculates *p*-values and corresponding significance thresholds in a statistical testing step.

The COMBI and the DeepCOMBI method share the same types of input and output variables but differ in the way they identify candidate locations by training an SVM and using the weights as importance scores on the one hand (**Figure 11 A** - COMBI) or learning a DNN and applying LRP to compute relevance scores on the other (**Figure 11 B** - DeepCOMBI). In identical final steps, both methods apply multiple hypothesis testing and appropriate thresholding to calculate *p*-values and obtain a list of significant phenotype-genotype associations.

The first novel methodology of this dissertation, the COMBI method (**Figure 11 A**), employs the ML technique of an SVM[11–13] in the first step of the aforementioned general idea. Crucially, SVMs are tailored to predict the target output (here, the phenotype) from high-dimensional data with a possibly complex, unknown correlation structure. In our application, COMBI trains an SVM based on a sample of observed genotypes and corresponding phenotypes and interprets the absolute values of the parameter vector as a measure of importance of each SNP for the prediction function. After post-processing the weights through a *p*-th-order moving average filter, the SNPs corresponding to the largest weights are selected for multiple hypothesis testing while all other SNPs are discarded. Since the SVM is trained using the complete SNP data of one chromosome, the first step acts as a filter, selecting only SNPs that are relevant for phenotype classification with either high individual effects or effects in combination with the rest of the SNPs of that chromosome, while discarding artifacts due to correlation structures. The second step uses multiple statistical hypothesis testing for a quantitative assessment of the individual relevance of the selected SNPs. The significance threshold is calibrated using a permutation-based method over the *whole* procedure**.** All in all, the two steps extract complementary types of information, which are combined in the final output. Importantly, the calibration of the method is such that a global statistical error criterion is controlled for the entire procedure consisting of steps 1 and 2.

The second contribution of this dissertation - the DeepCOMBI method (**Figure 11 B**) - employs artificial NNs and XAI in the first step of the general algorithm from above for identifying SNP-phenotype associations. The method is based on a deep learning step that trains a DNN to classify GWAS subjects into their respective phenotype class. Using an explanation method, the contributions of all SNPs to each individual classification result are assessed. The obtained relevance scores are used to select the subset of most relevant SNPs for the final multiple testing step, where only the SNPs selected in step 2 are tested for statistically significant association with the trait under investigation. The same permutation-based procedure used in COMBI is applied here to calibrate the significance threshold. DeepCOMBI can be viewed as an extension of the COMBI method replacing the prediction step of the SVM with a more sophisticated deep learning method and using the concept of explainability to extract SNP relevance scores via LRP[16–18]. To the best of our knowledge, deep Taylor-based explanation techniques[17] have not yet been applied in the field of GWAS for the analysis of such data. LRP provides a direct way to compute feature importance scores and has been

applied very successfully in numerous data science problems to explain decisions of NNs[163,204]. Instead of basing the importance score of a SNP on the data of that SNP alone, correlation structures and possible interactions are automatically taken into account. The main motivation behind DeepCOMBI is to harness the immense potential of sophisticated, state-of-the-art AI methods to examine complex and potentially nonlinear structures in high-dimensional data by applying the concept of DNNs to GWAS in the first step of the algorithm. Subsequently, in step 2, DeepCOMBI identifies a set of SNPs that have strong effects on the classification result of the DNN either individually or in combination with other SNPs and not due to correlation structures by calculating an explanation score for each SNP, which reflects its contribution to the final classification decision.

In the following **Chapter 3.2**, we first describe the methodologies behind COMBI and DeepCOMBI and then provide details on our approach to validating their superiority over other methods on both controlled generated datasets as well as on a 2007 GWAS dataset of seven common diseases[14]. The performances of the COMBI and DeepCOMBI methods are reported and compared in **Chapter 3.3**, where we also include and discuss the highly favorable comparisons with the algorithms that could potentially compete with both methods. Note that COMBI and DeepCOMBI yield better predictions with fewer *false* (*i.e.* non-replicated) and more *true* (*i.e.* replicated) discoveries when its results are validated on later, larger GWAS. DeepCOMBI compares favorably to the COMBI method in terms of both classification accuracy as well as SNP association prediction when validated with all associations reported within the GWAS catalog accessed in 2020. A thorough discussion of the results and all related ML work, along with a concluding summary, is given in **Chapter 3.4**.

An implementation of both methods in Python is available on GitHub at https://github.com/AlexandreRozier/DeepCombi. Further Implementations of the COMBI method are available in R, MATLAB and JAVA, as a part of the GWASpi toolbox 2.0 (https://bitbucket.org/gwas_combi/gwaspi/, login user name: gwas_combi_guest, password: combi123).

3.2 Methods

In the following sections, we formally introduce the statistical problem under investigation in a GWAS, present the detailed methodology behind the two proposed methods - COMBI and DeepCOMBI - and present the corresponding evaluation procedures on generated synthetic data and a real-world application on a known GWAS dataset.

3.2.1 Problem setting

A GWAS, as described in **Chapter 2.1.3**, investigates the observed genotypes x of a set of SNPs and a set of subjects labeled with the corresponding phenotypes y. Let n denote the number of subjects in the study and d the number of SNPs under investigation. Given a sample of observed genotypes $x = \left(x_{ij}\right)_{1\leq i\leq n, 1\leq j\leq d} \in \Re^{n\times 3d}$ and corresponding phenotypes $y = (y_1, \ldots, y_n) \in \Re^n$, each x_{i*} and each x_{*j} corresponds to a subject and a SNP, respectively. Both the genotypic information in SNP j and the phenotypes of subject i are encoded in a binary way. The number of minor alleles in SNP j of subject i is represented by $x_{ij} \in \{(1,0,0),(0,1,0),(0,0,1)\}$ and $y_i \in \{+1,-1\}$ or $y_i \in \{0,1\}$ for all $i = 1, \ldots, n$ is the binary label separating cases from controls. The data of one SNP can be summarized in a contingency table as presented in **Table 1**.

Table 1: Tabular representation of single SNP data. Single SNP data is summarized in categories according to phenotypes (cases, Y = +1 and controls, Y = -1) and genotypes (A_1A_1, A_1A_2 and A_2A_2). The numbers n_{cg} denote the numbers of individuals within the corresponding group. The total number of subjects in the study is n.

	A_1A_1	A_1A_2	A_2A_2	$\sum$
$Y = +1$	n_{11}	n_{12}	n_{13}	$n_{1.}$
$Y = -1$	n_{21}	n_{22}	n_{23}	$n_{2.}$
$\sum$	$n_{.1}$	$n_{.2}$	$n_{.3}$	n

The numbers n_{cg} denote the number of cases ($c = 1$) and controls ($c = 2$), respectively, which exhibit the genotype corresponding to column g. Notice that the row sums $n_{1.}$ and $n_{2.}$ are fixed and non-random by the experimental design (case-control study). Hence, the two random vectors $(n_{11}, n_{12}, n_{13})^T$ and $(n_{21}, n_{22}, n_{23})^T$ follow a multinomial distribution with three categories each, sample sizes $n_{1.}$ and $n_{2.}$ and unknown vectors of probabilities $p_1 = (p_{11}, p_{12}, p_{13})^T$ and $p_2 = (p_{21}, p_{22}, p_{23})^T$, respectively.

The parameter $\vartheta = (\vartheta_j)_{j=1,\ldots,d}$ of the statistical model for the whole study thus consists of all such pairs $\vartheta_j = (p_1^{(j)}, p_2^{(j)})$ of multinomial probability vectors, one for each of the d SNPs under investigation. For every SNP j, we are interested in testing the null hypothesis $H_j : p_1^{(j)} = p_2^{(j)}$, where we introduce the superscript j to indicate the SNP. This hypothesis is equivalent to the null hypothesis that the genotype at locus j is independent of the binary trait of interest. In general, the null hypothesis of a conventional single-locus test is that there is no difference between the trait means of the genotype groups, which indicates that the genotype at SNP j is independent of the phenotype under investigation[67]. Standard asymptotic tests for H_j versus its two-sided alternative K_j (genotype j is associated with the trait) are the χ^2 test for association and the Cochran-Armitage trend test[205]. Both tests employ test statistics that are asymptotically (as $\min(n_{1.}, n_{2.})$ tends to infinity) χ^2 distributed under H_j. The number of degrees of freedom equals two for the χ^2 test for association and 1 for the Cochran-Armitage trend test. Thus, p-values $(p_j : 1 \leq j \leq d)$ corresponding to these tests can be calculated by applying the upper-tail distribution function of the χ^2 distribution with the corresponding degrees of freedom to the observed values of these statistics, and this for every SNP. Observe that the test statistics obtained for different SNPs are highly correlated if these SNPs are in strong LD to each other; consequently, the corresponding p-values also exhibit strong dependencies[206,207]. RPVT, as described in **Chapter 2.2.1**, calculates a p-value p_j for each SNP j via a χ^2 test and declares it significantly associated with the phenotype if $p_j \leq t^*$. If there was a single test to perform (*i.e.* $d = 1$), then t^* would be taken as a predefined significance level α, as in the classical approach to statistical hypothesis testing. In multiple testing, however, the threshold t^* is modified to take the multiplicity of the problem (the fact that $d > 1$) into account. The simplest method to take multiplicity into account is the so-called Bonferroni correction, which sets $t^* = \frac{\alpha}{d}$.[70] This choice guarantees that the *FWER* (that is, the probability of one or more erroneously reported associations) of the multiple tests is bounded by α. A variety of other RPVT methods are explained, for instance, in the monograph by Dickhaus[208].

3.2.2 Proposed workflow

The individual RPVT p-value for an association of the j-th SNP only depends on x_{*j} and thus ignores any possible correlations and interactions with other SNPs – which could yield additional information. In contrast, ML approaches aimed at prediction take the information of the whole genotype into account at once and thus implicitly consider all possible correlations to strive for an optimal prediction of the phenotype. Based on this observation, we propose to calculate p-values only for the SNPs that are of importance in the decision process of such machines. We combine the advantages of both worlds - *i.e.* statistical hypothesis testing as the traditional way to compute

associations scores and ML, which takes multi-locus effects into account - by developing algorithms that consist of the following two basic steps:

- the ML step, where an appropriate subset of candidate SNPs is selected, based on their relevance for the prediction of the phenotype;
- the statistical testing step, where a hypothesis test is performed for the selected set of candidate SNPs together with an appropriate threshold calibration.

The ML step can be implemented in various ways and in this dissertation, we investigate two possible options: SVMs, where the learned weight vector is interpreted as an importance score, and DNNs, where LRP scores are computed as relevance indicators. The two resulting methods - COMBI and DeepCOMBI - are discussed in detail in the following sections.

The COMBI method

Combining the concepts of SVMs (as described in **Chapter 2.3.2**) and statistical testing (as described in **Chapter 2.2.1**), we propose a novel algorithm consisting of the following two steps (See **Figure 11A**):

1. **SVM training and SNP-selection:** Given the genotypes $x = \left(x_{ij}\right)$ and the corresponding phenotypes $y = (y_i)$ of a GWAS - an SVM is trained for phenotype prediction. The SVM returns a linear function $f_{w,b}(x) = w^T x + b$ and the sign of $f_{w,b}(x_{new})$ is a prediction of the unknown phenotype of a previously unseen genotype x_{new}. For each SNP *j*, the absolute value $|w_j|$ of the corresponding component of the parameter vector *w* is interpreted as a measure of importance for the prediction function. The SNPs corresponding to the *k* largest values of the scores $|w_j|$ are selected; all other SNPs are discarded.
2. **Statistical testing:** A hypothesis test is performed for each of the selected SNPs. Those SNPs with a *p*-value less than a significance threshold t^* are returned. The threshold t^* is calibrated using a permutation-based method over the *whole* procedure.

The above steps are presented in detail in the following sections.

The first step of COMBI - SVM training and SNP selection

The goal in ML is to determine, based on the sample, a function $f(x)$ that predicts the phenotype y based on the observation of genotype x. COMBI trains an SVM based on the sample of observed genotypes $x = \left(x_{ij}\right)$ with $x_{ij} \in \{(1,0,0),(0,1,0),(0,0,1)\}$ and corresponding phenotypes $y = (y_i)$ with $y_i \in \{+1, -1\}$. The SVM determines the

parameter w of the linear model $f_{w,b}(x_{i*}) = w^T x_{i*} + b$ by solving, for $C>0$, the following optimization problem:

$$w = argmin_w \left(\|w\|_2^2 + C \sum_{i=1}^{n} max(0, 1 - y_i w^T x_{i*}) \right).$$

Here, we aim to minimize the trade-off (controlled by C) between a vector w with a small norm (on the left) and small errors on the data (on the right). In a preprocessing step, the data is centered and scaled.

The learned prediction function f can be used to predict the phenotype of any genotype by calculating

$$y_{new} = sgn(f(x_{new})) = sgn(w^T x_{new}).$$

The above equation shows that the largest components (in absolute value) of the vector w (called SVM *parameter* or *weight* vector) also have the most influence on the predicted phenotype. Note that the weight vector contains three values for each position due to the feature embedding, which encodes each SNP with three binary variables. To convert the vector back to the original length, we simply take the average over the three weights. We also include an offset by including a constant feature that is all one.

Considering that the use of SVM weights as importance measures is a standard approach[209], for each j, the absolute value $|w_j|$ can be interpreted as a measure for the importance of the j-th SNP for the phenotype prediction task. The main idea is to select only a small number k of candidate SNPs before statistical testing, namely those SNPs having the largest scores.

Before selecting the most relevant SNPs, the parameter vector w is processed through a p-th-order moving average filter, that is:

$$|w_j^{new}| := \sqrt[p]{\sum_{h=max(1,j-(l_{filter}-1)/2)}^{min(d,j+(l_{filter}-1)/2)} |w_h|^{p_{filter}}},$$

where $l_{filter} \in 1, \ldots, d$ denotes a fixed filter length or window size (required to be an odd number). The value $p_{filter} \in]0, \infty[$ is a free norm parameter; in the case $p_{filter} = 1$, a standard moving average filter is obtained.

Finally, the SNPs corresponding to the k largest values of the scores $|w_j^{new}|$ are selected; all other SNPs are discarded.

The second step of COMBI - Statistical testing

In the statistical testing step, a hypothesis test (carried out as a χ^2 test) is performed for each of the k selected SNPs exactly as described above for RPVT, with the only modification that p-values for SNPs not ranked among the top k in terms of their filtered SVM weights are set to 1, without calculating a test statistic. A SNP j is said to be statistically associated with the trait if its p-value p_j is smaller than a threshold t^*, which is carefully chosen to bound the *FWER* by α.

In summary, the proposed methodology of the COMBI method, including the first and the second step from above, is formally stated as **Algorithm 1**.

Algorithm 1

THE COMBI METHOD.

Require: genotypes $x = (x_{ij})$ and phenotypes $y = (y_i)$, a reasonable upper bound $k \in \{1, \ldots, d\}$ for the number of informative SNPs, and an *FWER* level α

1: train an SVM using genotypes x and phenotypes y, resulting in scores $w_1, \ldots, w_d$
2: filter the weights $w_1, \ldots, w_d$ through a p-th-order moving average filter
3: let $w^*(k)$ denote the k-th largest of the w_j's in absolute value and re-number the corresponding positions from $1, \ldots, k$
4: **for all** $j = 1, \ldots, k$ **do**
5: compute the p-value of the j-th SNP $p_j(x_{*j}, y)$
6: **end for**
7: decide that SNP j is associated with the trait if $w_j \geq w^*(k)$ and $p_j < t^*$, where $t^* \equiv t^*(k, \alpha)$ is chosen as the α-quantile of the permutation distribution of the smallest of the k p-values (see **Algorithm 2** for details)

Return predicted set of informative SNPs

The methodological challenge now consists of finding a threshold t^* for the remaining k p-values such that the *FWER* is controlled for the multiple tests defined by the entire workflow, including SVM training, filtering of weights, p-value calculation and thresholding. The Bonferroni correction can only attain the prescribed *FWER* upper bound and therefore have maximal power if the p-values $(p_j : 1 \leq j \leq d)$ do not exhibit strong (positive) dependencies. This assumption is violated in GWAS due to strong LD in blocks of SNPs and also in the proposed COMBI method, where the SVM weight of a SNP j depends heavily on the information in other SNPs. To this end, we investigated previously proposed approaches[210,211] in order to split the sample, meaning that the selection of k SNPs is made on one (randomly chosen) sub-sample of individuals, while the p-value calculation and thresholding for the selected SNPs are performed on another. In this scheme and regardless of the SNP selection method used on the first sub-sample, a Bonferroni-type threshold $t^* = \frac{\alpha}{k}$ guarantees *FWER* control at level α for

the *p*-values computed on the second sub-sample. Since $k \ll d$, this correction is much less conservative than the original Bonferroni correction using all SNPs. However, this is severely mitigated by the loss of power in the *p*-values due to the sample splitting. In fact, computer simulations indicate very low power for detecting true associations with such a method because of the reduced sample size for calculation of test statistics and *p*-values (see **Figure 19** in **Chapter 3.3.1**). An alternative way to calibrate the threshold t^* for *FWER* control, taking the dependencies into account, is the Westfall-Young permutation procedure[212], which controls the *FWER* under an assumption termed "*subset pivotality*" (see Westfall and Young[212] as well as Dickhaus and Stange[207]). Furthermore, Meinshausen *et al.* [213] prove that this permutation procedure is asymptotically optimal in the class of RPVT procedures, provided that the subset pivotality condition is fulfilled. A thorough discussion and derivation of the extension of the Westfall and Young procedure, its assumptions and validity, in general, can be found here[207,213].

Based on these findings, our suggestion is to resample the entire workflow of **Figure 11A**, following a Westfall and Young type procedure, and to choose t^* based on the permutation distribution of the resampled *p*-values. *FWER* control at level α of the multiple tests defined by **Algorithm 1** can be proven under a relaxed form of the subset pivotality condition, the validity of which is checked empirically in **Appendix Chapter I**. To describe this condition formally, let P_0 denote any probability measure under the global null hypothesis of no informative SNPs in $\{1, \ldots, d\}$ at all. We assume that the following condition holds true: Let p_{min} denote the smallest of the k *p*-values corresponding to the positions selected by the SVM method for which the null hypothesis of no association between SNP and trait is true. Regarding p_{min} as a random variable, assume that its distribution under the true data-generating distribution P_ϑ (which is unknown) is stochastically not smaller than under P_0. The distribution under P_0 of the k *p*-values corresponding to positions chosen by applying the SVM method is now estimated by the resampling procedure given below as **Algorithm 2**. We estimate the distribution of *p*-values under the global null hypothesis of no informative SNPs by repeatedly assigning a random permutation of the phenotypes $y_{\pi(1)}, \ldots, y_{\pi(n)}$ to the observed genotypes $x_{1*}, \ldots, x_{n*}$ and applying the complete workflow of the COMBI method to save the resulting *p*-values of the B Monte Carlo repetitions. The empirical lower α-quantile of the smallest of these k *p*-values is then a valid choice for t^* in the sense that the *FWER* for the entire procedure defined by **Algorithm 1** is bounded by α. In contrast to the Bonferroni calibration, this procedure takes all dependencies in GWAS datasets caused by strong LD into account.

Note that the choice $k = d$ leads to skipping the SVM step and arriving at the popular *MinP procedure*, originally proposed in Westfall and Young[212]. Following the argumentation in Dudoit and van der Laan[214], it is also possible to control the *gFWER* with the aforementioned resampling scheme as well as the *ENFR*. For *gFWER* control with parameter ϱ, one has to consider the $(\varrho + 1)$th-smallest of the resampled *p*-values instead of $p_{min}^{(b)}$ in **Algorithm 2**. For *ENFR* control, one has to store all $B * k$ computed *p*-values and determine the *p*-value threshold that leads to an average number of rejections (over the *B* Monte Carlo repetitions), which matches the desired *ENFR* level.

Algorithm 2

RESAMPLING-BASED THRESHOLD DETERMINATION.

Require: genotypes $x = \left(x_{ij}\right)$ and phenotypes $y = (y_i)$, the number $k \in \{1, \ldots, d\}$ as in **Algorithm 1**, an *FWER* level α and a number *B* of Monte Carlo repetitions

1: **for** $b = 1, \ldots, B$ **do**
2: pick a random permutation π and set $y^{(b)} = \left(y_{\pi(1)}, \ldots, y_{\pi(n)}\right)$
3: carry out steps 1-6 in **Algorithm 1** with taking $y^{(b)}$ as phenotypes, resulting in corresponding *p*-values $p_j\left(x_{*j}, y^{(b)}\right)$
4: store the smallest of the k computed *p*-values as $p_{min}^{(b)}$.
5: **end for**
6: Order the $p_{min}^{(b)} : 1 \leq b \leq B$ increasingly as $\left(p_{ordered}^{(b)} : 1 \leq b \leq B\right)$.

Return the value $p_{ordered}^{(\alpha * B)}$

Implementation details of the COMBI method

The COMBI method is implemented in Matlab/Octave, R and Java as a part of the GWASpi toolbox 2.0 (https://bitbucket.org/gwas_combi/gwaspi/, login user name: gwas_combi_guest, password: combi123.). The implementation for Matlab/Octave is cluster-oriented and uses libLinear[215]. The Java implementation is desktop computer oriented and makes use of the following packages: libLinear[215], libSVM[216] and apache commons math[217]. Finally, the R implementation requires LiblineaR[218], qqman[219], data.table[220], gtools[221] and snpStats[222]. The COMBI method is also available in Python as part of the DeepCOMBI implementation, which can be found at https://github.com/AlexandreRozier/DeepCombi.

The DeepCOMBI method

Combining the concepts of DNNs (as described in **Chapter 2.3.3**), explanation methods (as described in **Chapter 2.3.4**) and statistical testing (as described in **Chapter 2.2.1**), we propose another novel algorithm consisting of the following three steps (see **Figure 11B**):

1. **Deep learning:** Given the genotypes $x = \left(x_{ij}\right)$ and the corresponding phenotypes $y = (y_i)$ of a GWAS, a DNN is trained for phenotype prediction.
2. **Explanation and SNP selection:** A subset of SNPs is selected by applying LRP[16–18] as an explanation method for each individual prediction and averaging the absolute values of the resulting explanations to compute global prediction relevance scores $r = \left(r_j\right)$ for all d SNPs j. The relevance scores are processed through a moving average filter with a window size l_{filter}. Given a predefined upper bound $k \in \{1, \ldots, d\}$ for the number of informative SNPs, we select the k most relevant SNPs based on r.
3. **Statistical testing:** A hypothesis test is performed for all SNPs selected in the previous step to compute the p-values of those SNPs, while the p-values of all other SNPs are set to one. Via a permutation-based threshold calibration and given a FWER level α, we decide that SNP j is associated with the trait if $p_j < t^*$, where $t^* \equiv t^*(k, \alpha)$ is chosen as the α-quantile of the permutation distribution of the k smallest p-values.

The proposed algorithm can be viewed as an extension of the COMBI method replacing the SVM with a state-of-the-art deep learning method in combination with an explanation technique. The above steps are presented in detail in the following sections.

The first step of DeepCOMBI - Deep learning

The first step of the proposed method consists of constructing and training a well-performing DNN for the prediction of the phenotypes $y = (y_i)$ (here $y_i \in \{0, 1\}$) of a GWAS given the corresponding genotypes $x = \left(x_{ij}\right)$. Selecting a DNN architecture is often critical for achieving good performance for a specific - in this case, SNP-based - classification task. Montaez *et al.*[201] developed a 2-class DNN for the classification of polygenic obesity and showed its performance to be superior to numerous competitor methods. Romagnoni *et al.*[193] compared the performance of similar architectures and presented a detailed review of the best design choices for an NN on a Crohn's Disease dataset. Taking inspiration from the conclusions of both of these works and investigating performances on synthetic datasets, we use an architecture of two fully connected layers with 64 neurons and ReLU activations and a dense softmax output

layer with two output nodes. To improve validation accuracy by reducing overfitting, each hidden layer is followed by a dropout layer with a dropout probability of φ. We employ a classic cross-entropy loss function coupled with an L1-L2 mixed regularization term (defined as in **Chapter 2.3.3**) to avoid overfitting by minimizing the trade-off between small errors on the data and small L1 and L2 norms of the weights. Adam[55] is used as an adaptive learning rate optimizer to minimize the given loss function. To overcome limitations due to imbalanced datasets, class weights are calculated according to the class frequencies and used to direct the DNN to balance the impact of controls and cases. In a preprocessing step, the data is centered and scaled by subtracting the global mean and dividing by the global standard deviation. To minimize computational effort and limit the number of model parameters in the DNN, a *p*-value threshold κ can be applied in order to only select SNPs with *p*-values smaller than κ to be used for training.

Once the parameters w_{st}^{q} of the DNN have been trained by optimizing the corresponding learning problem (See **Chapter 2.3.3** for details), the network is able to predict the phenotype of any unseen genotype x_{new}. Regarding this binary classification problem, the output node with the highest score represents the predicted phenotype.

The second step of DeepCOMBI - Explanation and SNP selection

To harness the potential of DNNs for identifying SNP disease associations in GWAS, we now apply the concept of XAI. Once the DNN is fully trained, the aim is to define an importance measure that determines which loci play an important role in the determination of a phenotype. Generating relevance scores from trained DNNs can be achieved by utilizing LRP[16–18], which we have introduced and described in detail in **Chapter 2.3.4**. LRP firstly computes the prediction scores of a datapoint x_i by $f(x_i) = \widehat{y_i}$, a forward pass through the network after a DNN f is trained on a prediction task. Secondly, following a specific propagation rule, a single output score $\widehat{y_i}$, *i.e.* the highest output score, is backpropagated successively layer-by-layer through the network until reaching the input layer. Here, the **LRP-$\alpha\beta$ rule** (also see **Chapter 2.3.4** for a definition) is used, which allows us to weigh the positive and negative contributions of each neuron to their predecessor differently with the hyperparameters α_{LRP} and β_{LRP}. Once the input layer $R^{(0,i)} \in \Re^{3*d}$ is reached, the relevance score ρ_{ij} of SNP j in subject i is attributed to each dimension of x_i with $\rho_{ij} = \left(\sum_u R_u^{(0,i)}\right)/3$. Since the original relevance vector $R^{(0,i)}$ contains three values for each one-hot-encoded location, it is converted back to size d by averaging over the three nodes $u \in \{(j \times 3) - 2,\ (j \times 3) - 1,\ (j * 3)\}$ corresponding to SNP j in the input layer. Please note that all relevance scores ρ_{ji} are positive since a softmax output layer with two output nodes for the binary classification problem is used and only the highest of the

two output activations is backpropagated. ρ_{ij} now demonstrates to which extent the dimension j of x_i plays a role in the classification decision $f(x_i)$ and can be used to uncover the most relevant SNPs for prediction. Note, however, that LRP is applied individually to each datapoint i. By averaging the values of all individual LRP explanations ρ_{ij} of SNP j, we propose to generate a global explanation, that is:

$$r_j = \left(\sum_{i=1}^{n} \rho_{ij} \right) / n$$

which is independent of datapoints. The relevance scores of one sample sum up to the activation value of the output prediction, which means that datapoints classified with low certainty also have a small impact on the global explanation. Intuitively, the global LRP score r_j of each SNP j can now be interpreted as a measure of relevance regarding the prediction: The higher r_j, the greater the influence of locus j on the decision process of the DNN. To achieve better performance, the SNP relevance scores are now filtered - like the SVM weights of the COMBI method - before using them to select the highest scoring locations. The vector r is post-processed through a p-th-order moving average filter, that is:

$$r_j^{new} := \sqrt[p]{\sum_{h=max(1,j-(l_{filter}-1)/2)}^{min(d,j+(l_{filter}-1)/2)} (r_h)^{p_{filter}}}$$

where $l_{filter} \in 1, \ldots, d$ denotes the window size and $p_{filter} \in]0, \infty[$ is a norm parameter. We have now generated relevance scores showing which SNPs played an important role in the classification decision and can use them for the selection of promising locations. For the next step of DeepCOMBI, we choose to test all SNPs with the k largest values of the scores r_j^{new} and eliminate all SNPs with lower relevance.

The third step of DeepCOMBI - Statistical testing

The statistical testing step of the DeepCOMBI method is identical to the second step of the COMBI method. A χ^2 hypothesis test is performed for each of the k SNPs selected in the LRP explanation step and the p-values for all other SNPs are set to one.

In summary, the proposed methodology, including all steps, is formally stated as **Algorithm 3**.

Algorithm 3

THE DEEPCOMBI METHOD.

Require: genotypes $x = \left(x_{ij}\right)$ and phenotypes $y = (y_i)$, a reasonable upper bound $k \in \{1, \ldots, d\}$ for the number of informative SNPs and an *FWER* level α

1: train a DNN using genotypes x and phenotypes y
2: calculate global relevances $r_j, \ldots, r_d$ via averaging local LRP scores
3: filter the weights $r_j, \ldots, r_d$ through a p-th-order moving average filter
4: let $r^*(k)$ denote the k-th largest of the r_j's in absolute value and re-number the corresponding positions from $1, \ldots, k$
5: **for all** $j = 1, \ldots, k$ **do**
6: compute the p-value of the j-th SNP $p_j\left(x_{*j}, y\right)$
7: **end for**
8: decide that SNP j is associated with the trait if $r_j \geq r^*(k)$ and $p_j < t^*$, where $t^* \equiv t^*(k, \alpha)$ is chosen as the α-quantile of the permutation distribution of the smallest of the k p-values (see **Algorithm 2** for details)

Return predicted set of informative SNPs

To identify statistically significant associations, the p-value threshold t^* is calibrated to control the *FWER* for multiplicity by applying the same permutation procedure as proposed above for COMBI in **Algorithm 2**, replacing "*Algorithm 1*" with "*Algorithm 3*" in the requirements and in line 3.

Implementation details of the DeepCOMBI method

The DeepCOMBI method is implemented in Python and the source code is available at https://github.com/AlexandreRozier/DeepCombi. The implementation uses the DNN development library Keras[223] in combination with the LRP library iNNvestigate[158]. The code for simulating GWAS datasets is also available at https://github.com/AlexandreRozier/DeepCombi.

3.2.3 Datasets and corresponding validation strategies

To evaluate the performances of the proposed methods in comparison to their competitor methods, we analyze a number of different datasets. First, we generate semi-real GWAS datasets, where the underlying truth of associated SNPs is known and the methods can be investigated in a controlled environment. Afterwards, we examine seven real GWAS datasets from a 2007 study[38] and evaluate the findings of the methods in terms of replication in independent studies. The datasets and corresponding validation strategies are presented in the following paragraphs.

Validation on generated datasets

First, we aim to assess the performance of the two proposed methods in comparison to other methods in controlled simulation experiments. To create a realistic but controlled environment where the ground-truth labels of a dataset, *i.e.* the SNPs that are indeed linked to the disease, are known, we generate semi-real datasets with real genotypes and synthetic phenotypes. The basic concept is to take an ensemble of real genotypes and generate a synthetic phenotype for each subject according to a specific rule. With this method, the underlying architecture of the genome, including, for example, genetic LD and correlation structures, is kept intact while control over the phenotypic labels is gained at the same time. We use the WTCCC data[38] described in more detail below and randomly select 300 subjects of the Crohn's disease dataset. We draw a random block of 20 consecutive SNPs from chromosome 1 and a random block of 10,000 consecutive SNPs from chromosome 2 without breaking linkage. The former, smaller block represents the set of informative SNPs to be associated with the phenotype in this experiment, while the latter, larger block constitutes the set of uninformative SNPs. These *noise* SNPs are placed surrounding the 20 informative loci, which thus are to be found at the positions 5,001 to 5,020. Synthetic phenotypes are now generated based on only one of the 20 associated SNPs (at position 5,010), using a logistic regression model. The corresponding probability function describes the statistical distribution of phenotypes as

$$P\left(Y_i = +1 | X_{i,*} = x_{i,*}\right) = \left(1 + exp\left(-\gamma\left(x_{i,5010} - median\left(x_{*,5010}\right)\right)\right)\right)^{-1},$$

where, γ is an effect size parameter, $x_{i,*}$ is the allele sequence in nominal feature encoding (*i.e.* x_{ij} is the number of minor alleles in SNP j of subject i) and Y_i is the generated phenotype of subject *i*. Basing the label of a subject on the SNP at position 5,010 creates associations to all 20 informative SNPs because there are real covariance structures and LD within this set of SNPs. A typical tower-shaped *p*-value formation with realistic covariances appears in the resulting Manhattan plot. The tower structure is limited to those 20 informative positions because there are no correlations of those SNPs with the surrounding 10,000 noise SNPs coming from chromosome 2. The

random generation process also ensures that the datasets have associations of different strengths to the 20 informative SNPs, which increases similarity to real GWAS datasets. Three exemplary datasets are shown and investigated in **Figure 13** in **Chapter 3.3.1**. The complete process of drawing random genotypes and generating the corresponding phenotypes is repeated 1,000 times to generate 1,000 datasets. COMBI, DeepCOMBI and all baseline methods are applied to each dataset with an 80:20 class balanced split into training and test data. The results on the test data are evaluated with the known ground truth of only 20 informative SNPs at the positions 5,000 to 5,020 and the corresponding performance can be measured in terms of the number of true and false positives for each method.

An additive heritability model is assumed appropriate for this simulation study for a number of different reasons. Most importantly, it is the standard model in the field of SNP effect estimation, genomic risk score computation and other related problems[224,225]. This is especially true for the seven diseases that are studied in this dissertation. In the original WTCCC study, an additive test is used as the null model, spotting only a few cases where departure from this additivity are observed[38]. Additive, infinitesimal models have been shown to perform well in the research area of quantitative genetics and, indeed, most GWAS hits seem to behave additively[226–228]. It is also the most agnostic model, with few parameters and no assumptions on values of dominance or complex interactions between loci[224,225]. The investigation of other models for the genetic architecture of a disease could be the subject of future research projects.

The validation results using the generated semi-real datasets are presented in **Chapter 3.3.1**. In addition to assessing the effectiveness of the proposed algorithms, the generated datasets with access to the true set of informative SNPs are also used to identify optimal parameter settings for both COMBI and DeepCOMBI in a controlled environment, as described in **Chapter 3.2.4**.

Validation on WTCCC data

For evaluation on real-world genomic data, the performances of COMBI and DeepCOMBI are assessed on the WTCCC phase 1 dataset, released in 2007[38], featuring the genotypic information of 17,000 British subjects. With 3,000 shared controls and 2,000 case samples for seven major human diseases (Crohn's disease (CD), type 1 diabetes (T1D), type 2 diabetes (T2D), coronary artery disease (CAD), hypertension (HT), bipolar disorder (BD) and rheumatoid arthritis (RA)), it was a landmark study both in terms of sample size and dimensionality at the time of its publication.

Here and in contrast to the simulation experiments described above, the true underlying genetic architectures of the traits under investigation, *i.e.* the sets of informative SNPs for each disease, are largely unknown. Hence, for evaluation purposes, the concept of replicability in independent studies is used as a measure of performance. An approximation of the true set of informative SNPs is created by employing the GWAS

catalog[7] and examining the results of the 13 years of independent studies after the WTCCC dataset was published. In summary, we proceed as follows: the application of some method (for instance, COMBI, DeepCOMBI or RPVT) to the 2007 WTCCC data results in a list of SNPs that are potentially associated with the trait (this is illustrated on the left-hand side of **Figure 12**).

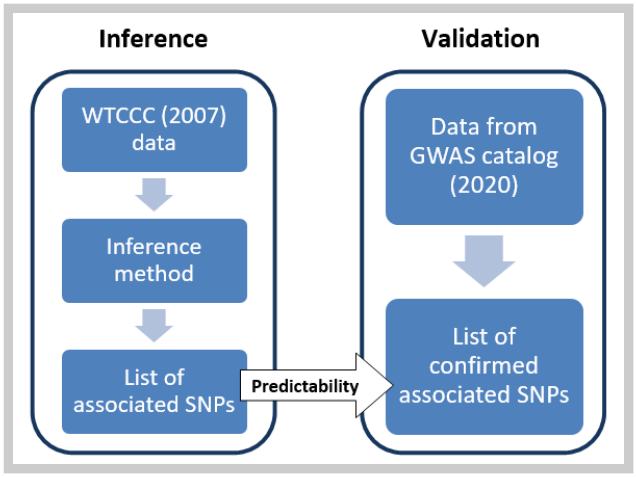

Figure 12: Illustration of the validation methodology on WTCCC data. After producing a list of associated SNPs via an appropriate inference method (*i.e.* COMBI, DeepCOMBI or RPVT), the GWAS catalog is used in an independent validation step to confirm or refute those candidate SNPs accessing the predictability of the used inference method.

This list of potentially associated SNPs is then evaluated considering replicability on independent data to obtain a "*List of confirmed associated SNPs*" (illustrated on the right-hand side of **Figure 12**). All studies for the WTCCC diseases included in the GWAS catalog constitute the set of studies examined for replicability. Most of these studies are performed either with larger sample sizes or using meta-analysis techniques and were published after the original WTCCC paper. In a sense, we thus examine how well any particular method, when applied to the WTCCC dataset, is able to make discoveries in that dataset that are actually confirmed by later research using RPVT in independent publications. To evaluate the reported finding of a method (COMBI, DeepCOMBI or competitor), the GWAS catalog is inquired for that SNP and all SNPs in LD (R^2>0.2) within a 200kb window around that SNP. LD calculations are performed with PLINK[229] based on the genomic sequences of the 85 CEU individuals from Phase 1 of the 1000 Genomes Project[27]. A hit indicates that a GWAS other than the original WTCCC study has since reported this SNP (or SNP in LD) to be associated with the disease. Note that the GWAS catalog only contains SNPs with *p*-values $<10^{-5}$, meaning that we miss some hits that are statistically weak, but that might be biologically relevant in the sense that they contribute to the classification of individuals according to phenotypes. With this procedure, methods can be compared by counting the respective number of replicated and unreplicated reported associations. If an association with the disease with *p*-value $<10^{-5}$ of the SNP itself or the SNPs in LD was reported by at least

one independent GWAS published after the WTCCC study, the reported SNP is counted as a true-positive finding. In contrast, all SNPs that were not replicated count as false negatives.

The seven 2007 WTCCC datasets were downloaded from the EGA website (European Genome-phenome Archive, ega.crg.eu) after being granted the corresponding access licenses. Since it is not publicly available, access can be requested from the owners at https://www.wtccc.org.uk/info/access_to_data_samples.html and https://www.sanger.ac.uk/legal/DAA/MasterController. The validation results using the seven WTCCC datasets are presented in **Chapter 3.3.2**.

3.2.4 Preprocessing and parameter selection

The application of COMBI and DeepCOMBI requires some preprocessing and the determination of a number of free parameters. In the following sections, we present the selected preprocessing steps and optimal parameter values for both methods. We describe the process of finding these values for the different datasets under investigation. For future applications, the choice of exact parameter values needs to be adapted for each particular phenotype or disease under study since they will have different genetic architectures and distribution of effect sizes[43,230]. For this manuscript and in order to provide a comprehensive and comparable set of results across many diseases, we employ a unique set of parameter values supported by the results of our simulation study and other findings in related literature.

Preprocessing and parameter selection for generated datasets

For the generation process of semi-real datasets, the effect size parameter is set to $\gamma = 6$. One-hot-encoded genotypic feature encoding is utilized, where all features are normalized such that the 6th centered moments equal one (this is similar to the common practice of scaling each feature to unit standard deviation).

The application of the COMBI method requires the selection of a number of free parameters (*e.g.* the SVM optimization parameter C, the window size l_{filter} of the moving average filter or the filter norm parameter p_{filter}). To this end, the generated semi-real datasets are used to determine performance changes induced by varying those free parameters. Most findings are in agreement with related literature and biologically sensible. For example, it is found that aggregating SNPs within the filtering step based on a filter window size of 35 is optimal, which is on the same magnitude as in Alexander and Lange[231], who find that grouping of SNPs into bins of size 40 helps the performance of their algorithm. The moving average filter of the COMBI method is designed to correct for non-independence of statistical tests within LD blocks. Given the SNP density in the arrays used by the original WTCCC study and LD patterns in the CEU population (1000 Genomes[27]), we estimate that the average LD block ($r^2 > 0.8$)

harbors no more than 20-30 SNPs[232], which supports our findings of setting the filter window size l_{filter} to 35 in the sense that we average-out blocks and conservatively add a bit of noise by potentially smoothing out signals across blocks. The norm parameter p_{filter} of the moving average filter is set to 2 and the SNP selection parameter to $k = 30$. Choosing an optimal SVM parameter using cross-validation-based model selection in each repetition of the Westfall-Young permutation procedure in order to maximize the (estimated) generalization ability of the function f does not result in a higher power of the COMBI method. Performance results for constant and cross-validated C are almost identical. Thus, time-consuming cross-validation is avoided and a fixed $C = 0.00001$ is used for all further applications of the COMBI method. A linear L2 regularized L1-loss dual classifier[215] is used to solve the SVM minimization problem. The simulation experiments show better performance when the χ^2 test for associations is applied instead of the Cochran-Armitage test**.**

When applying the DeepCOMBI method to the generated datasets, we study the effect of all hyperparameters on the performance of the DNN. An accuracy-based random grid search with a stratified split in 90% training and 10% testing data is conducted. Here, we present the selected most successful values and the investigated parameter intervals in parentheses:

- number of neurons per dense hidden layer $nn = 64$ [2, 4, 8, 16, 64],
- L1 regularization coefficient $\tau = 0.0001$ [0, 1e-6, 1e-5, 1e-4, 1e-3,1e-2, 1e-1],
- L2 regularization coefficient $\upsilon = 0.000001$ [0, 1e-6, 1e-5, 1e-4, 1e-3,1e-2, 1e-1],
- dropout rate $\varphi = 0.3$ [0.3, 0.5],
- learning rate $\eta = 0.01$ with a learning rate reduction on a plateau with factor 0.7125 after 50 epochs of no improvement,
- number of epochs $eps = 500$ [100, 500, 1000].

A few different parameter values of the $\alpha\beta$ - backpropagation rule are manually investigated on exemplary datasets. By visually inspecting the resulting LRP vectors and their corresponding DeepCOMBI p-values, the combination of $\alpha_{LRP} = 1$ [0, 1, 2] and $\beta_{LRP} = 0$ [0, 1, 2] is found to be most successful.

For post-processing the global relevance scores and selecting the most relevant SNPs, we assume that the most successful values found for COMBI would also be a good choice for DeepCOMBI. Hence, we set the window size of the moving average filter to $l_{filter} = 35$, the norm parameter of the moving average filter to $p_{filter} = 2$ and the SNP selection parameter to $k = 30$.

Preprocessing and parameter selection for WTCCC data

Given the original WTCCC dataset[38], a case-control dataset for each disease is created, removing all SNPs and samples that did not fulfill the quality control criteria of the original paper using the lists provided at the WTCCC website (www.wtccc.org.uk). This results in seven datasets, including more than 500,000 SNPs distributed across all 23

chromosomes. The sex chromosome is left out of our analysis since it would have to be treated differently than the other chromosomes[38]. Based on lists provided by the WTCCC, we remove an additional set of 579 false-positive SNPs from the analysis (*e.g.* SNPs that are significant, isolated hits, with no significance in the surrounding high LD SNPs, *i.e.* with no tower around them). Since (at the time of our study in June 2015) these lists lack the information corresponding to the CAD study, all genome-wide significant CAD SNPs ($< 5 * 10^{-7}$) that do not appear in the original WTCCC paper are manually removed. The one-hot-encoded genotypic feature encoding method is utilized and all features are normalized such that the 6th centered moments are one.

Since the findings on optimal parameter values on the generated datasets are in agreement with related literature and biologically sensible, the optimal settings there are assumed to be good choices for the application of COMBI and DeepCOMBI to real data. For example, we transfer the optimal values of the filter window size $l_{filter} = 35$ and the norm parameter $p_{filter} = 2$ to the application on the WTCCC dataset. Similarly, a linear L2 regularized L1-loss dual classifier[215] is employed and the χ^2 test for association is applied instead of the Cochran-Armitage test. Since we find no significant effect of the penalty parameter C on the generated semi-real datasets, we fix it to $C = 0.00001$ for the investigation of the real data, economizing computation time and memory space.

Some parameters of the COMBI and DeepCOMBI methods cannot be investigated within the simulation study but have to be chosen manually for the WTCCC data. The decision to train the SVM separately on each chromosome is one of those tuning steps, as genome-wide training is very time- and memory-consuming on the one hand and can only improve performance marginally on the other hand, as intergenic correlations between chromosomes are very rare. Hence, in agreement with the lack of inter-chromosomal LD, the COMBI method, the DeepCOMBI method and all baseline methods are applied to each chromosome separately. Another parameter that has to be chosen manually is the number of active SNPs in one chromosome, *i.e.* the parameter k, which is set to 100 SNPs per chromosome for the COMBI method after careful consideration**.** This choice is admittedly a wide, arbitrary upper bound for the number of SNPs that can present a detectable association with a given phenotype. As of June 2015, the maximum total number of SNPs (not independent signals) associated with any phenotype is ~450 for human height and 922 for Crohn's Disease (GWAS Catalog), so with $k = 100$ per chromosome, one is well within what evidence would support. After all, for future applications of COMBI, k is a tuning parameter, which has to be chosen by the researcher according to the assumed number of relevant loci.

To choose hyperparameters for the DNN trained on WTCCC data in the first step of DeepCOMBI, a parameter search is run on a single dataset. The Crohn's disease chromosome 3 dataset is selected as a good representative and an accuracy-based

parameter search with a stratified split in 90% training and 10% testing data is conducted. We study the effect of the hyperparameters on the performance of the DNN and the best performing hyperparameters are as follows (tested intervals in parentheses):

- number of neurons per dense hidden layer *nn* = 64 [2, 4, 8, 16, 64],
- L1 regularization coefficient τ = 0.001 [0, 1e-6, 1e-5, 1e-4, 1e-3,1e-2, 1e-1],
- L2 regularization coefficient υ = 0.0001 [0, 1e-6, 1e-5, 1e-4, 1e-3,1e-2, 1e-1],
- dropout rate φ = 0.3 [0.3, 0.5],
- *p*-value threshold κ = 1e-2 [1e-4, 1e-2, 1],
- learning rate η = 0.00001 [1e-7, 1e-6, 1e-5, 1e-4, 1e-3,1e-2, 1e-1],
- number of epochs *eps* = 500 [100, 500, 1000].

Detailed results on the classification performance of the final training parameter settings can be found in **Chapter 3.3.2**.

As before, we visually investigate a few different parameter values of the $\alpha\beta$-backpropagation rule and their influence on both the resulting relevance scores and *p*-values. On the Crohn's disease chromosome 3 dataset, the combination of $\alpha_{LRP} = 2$ [0, 1, 2] and $\beta_{LRP} = 1$ [0, 1, 2] is found to be optimal.

After manually investigating the global LRP scores and the corresponding DeepCOMBI *p*-values of the exemplary dataset (Crohn's disease chromosome 3), we find that slightly different settings than for the analysis of the generated datasets should be applied for post-processing the relevance vectors of DeepCOMBI and selecting the most relevant SNPs. Namely, the window size of the moving average filter is set to $l_{filter} = 21$ and the SNP selection parameter is increased to $k = 200$. The need for a decreased filter window size and an increased number of selected SNPs might be caused by the application of the *p*-value based preselection step for limiting the number of model parameters, which is only applied to the real dataset and not the generated datasets.

Regarding significance levels, we aim to stay as close in line with the original WTCCC study as possible, reporting not only the strong associations at the significance level of 5×10^{-7} but also weak associations at 1×10^{-5}. Within our validation pipeline, we consider the full NHGRI GWAS Catalog[1] with the inclusion criterion of having achieved a *p*-value of 1×10^{-5} in a GWAS. The "*somewhat liberal statistical threshold of* $p \leq 1 \times 10^{-5}$ *was chosen to allow examination of borderline associations and to accommodate scans of various sizes while maintaining a consistent approach*"[38].

When comparing the performance of the COMBI and DeepCOMBI methods with that of RPVT on WTCCC data, the challenge now is to determine the type 1 error level α to be used in the permutation test procedure that corresponds to the error level applied in the original study. We apply the basic idea of estimating the empirical distribution of *p*-values using the Westfall-Young[212] procedure and select the error level that the RPVT threshold used in the WTCCC publication corresponds to.

In the original study, when considering the stringent thresholding, $t^* = 5 \times 10^{-7}$ is determined to be a reasonable threshold for type 1 error control in RPVT stating that, "*the posterior odds in favor of a 'hit' being a true association would be 10:1*"[38] using this threshold. A sound upper bound on the *ENFR* level that this threshold implies is obtained by calculating the empirical distribution of *p*-values using the Westfall-Young[212] procedure. It turns out that the threshold of $t^* = 5 \times 10^{-7}$, on average, produces 0.17 non-replicated discoveries per disease, or 1.19 for all seven. Out of the 24 SNPs reported in WTCCC at $t^* = 5 \times 10^{-7}$, only one can be expected to be a false-positive, which corresponds to a true-to-false-positives ratio of 23:1. WTCCC also reports SNPs at the level $t^* = 1 \times 10^{-5}$, stating that "*if we relax the significance threshold by a factor of ten [...], the posterior odds that a 'hit' is a true association would also be reduced by a factor of ten.*"[38] The relaxed threshold of $t^* = 1 \times 10^{-5}$ thus refers to posterior odds of 1:2. According to our simulations, the controlled number of non-replicated discoveries to be expected is 3.32 per disease on average. This suggests that out of the 82 loci reported by the WTCCC at $t^* = 1 \times 10^{-5}$, we can expect that approximately 23 are false-positives, corresponding to an actual rate of true-to-false-positives of ~ 3:1.

To compare the performances of the COMBI and DeepCOMBI methods with that of RPVT in a fair way, we consequently calibrate both of these methods (using the Westfall and Young-type procedure described in **Algorithm 2** presented in **Chapter 3.2.2**) in a way such that the number of non-replicated discoveries is bounded by 3.32 per disease (using the augmentation for *ENFR* control of both algorithms). It should be noted that, when investigating the performance of the COMBI method with semi-real data simulations, we observe that the COMBI method produces only approximately 20% of the number of type 1 errors it is aiming to control for (see **Appendix Chapter I.**). Although it is not known whether this relation is true in the case of real data, one can still expect substantially fewer errors than what the calibration aims for, *i.e.* around 0.664 instead of 3.32 per disease if the data distribution was identical to that in the simulations.

3.2.5 Baseline methods

In order to evaluate the performances of the proposed COMBI and DeepCOMBI methods in comparison to competitor approaches, we select a set of representative baseline methods. RPVT is chosen as the most widely used traditional, purely statistical testing approach. In addition, we investigate two more important competitor methods, where we interpret the raw SVM weights and LRP scores as test statistics and threshold them directly. The three baseline methods of RPVT, the thresholding of raw SVM weights and the thresholding of raw LRP scores represent the separate components of the COMBI and DeepCOMBI methods and are therefore crucial in order to validate the

proposed thesis of this dissertation of combinatorial approaches being superior over their individual components in this setting.
Finally, three other combinatorial ML-based approaches (by Roshan *et al.*[62], Meinshausen *et al.*[213] and Wasserman and Roeder[210]) and two purely statistical analysis tools (by Lippert *et al.*[188,192]) are investigated.

All baseline methods are described in detail in the following sections.

RPVT as a baseline method

RPVT is the statistical framework traditionally used in GWAS for identifying significant associations between SNPs and traits. Please refer to **Chapter 2.2.1** and **Chapter 3.2.1** for an introduction. The single-locus null hypothesis to be tested states that the SNP at locus j is independent of the binary trait of interest, *i.e.* that there is no correlation between this particular SNP and the development of the disease under investigation. A standard statistical test for this hypothesis is the χ^2 -test[233], which tests for independence of the two multi-level variables genotype (three different levels: 0, 1 or 2 minor alleles) and phenotype (two different levels: case or control) by calculating the test statistic $\widehat{\chi}^2 = \sum_{c,g} \frac{(O_{c,g} - E_{c,g})^2}{E_{c,g}}$ where $O_{c,g}$ and $E_{c,g}$ are the observed and expected frequencies of genotype g in combination with the phenotype case-control status c. To compute a p-value, $\widehat{\chi}^2$ is then compared to a χ^2 distribution with two degrees of freedom. It represents the probability of observing a sample statistic as extreme as $\widehat{\chi}^2$ under the assumption of no association between genotype and phenotype. If the p-value is smaller than a predefined threshold t^*, the null hypothesis is rejected and we declare the SNP under investigation to be significantly associated with the phenotype. If there was a single test to perform, t^* would usually be equal to the significance level $\alpha = 0.05$. When performing multiple testing, however, the threshold is modified to take the multiplicity of the problem into account. The simplest method is the so-called Bonferroni correction[70], where t^* is divided by the number of tests performed, *i.e.* the number of SNPs d in this case, which guarantees that the FWER, the probability of one or more erroneously reported associations, is bounded by α. The Bonferroni correction performs well under the assumption that all null hypotheses are independent of each other, which is not the case here. Indeed, since SNPs show high degrees of correlation through LD, the Bonferroni correction can become extremely conservative, leading to a high rate of false negatives, which is why the scientific community mostly applies a fixed threshold that remains constant for multiple GWAS. Here, based on the original publication of the data we are analyzing (WTCCC data, see **Chapter 3.2.3**[38]), we present not only the strong associations at a significance level of $t^* = 5x10^{-7}$ but also weak associations at $t^* = 1 \times 10^{-5}$.

Raw SVM weights and LRP scores without statistical testing as baseline methods

Instead of interpreting the SVM weights from COMBI and the LRP scores from DeepCOMBI as relevance scores to select a subset of SNPs to calculate *p*-values for, this step can be skipped to instead use them as direct test statistics. For evaluation, the vector of raw SVM weights and LRP scores is treated like the vector of *p*-values of RPVT to calculate performance curves. We compare COMBI and DeepCOMBI to these baseline methods of raw relevance scores and RPVT to show that only the combination of ML / deep learning and multiple testing produce the desired performance increase, which cannot be achieved individually by one of the components.

Other baseline methods

In addition to comparing the proposed methods with the RPVT approach and the raw relevance scores, we investigate whether slight alterations and simplifications of the COMBI method can achieve the same level of effectiveness. On the generated datasets, we investigate the performance of the COMBI method without the moving average filter and the performance of RPVT when such a filter is applied to the raw *p*-values to show that both the SVM and the filter are crucial.

A number of experiments are performed to show that the novel methods also outperform other combinatorial ML-based approaches. There are only a few related ML methods, out of which we select three techniques as representatives to be compared to the COMBI method on the generated datasets. The first one was proposed by Roshan *et al.*[62] and is a version of the COMBI method where the order of the two steps (SVM training and statistical testing) is reversed. The second method proposed by Wasserman and Roeder[210] separates the two steps of the COMBI method and performs the ML step on one half of the data and the statistical testing step on the other half. The third competitor method by Meinshausen *et al.*[213] is an extension of this method, which aggregates the results of multiple random splits.

In a functional study on the WTCCC dataset, we also compare COMBI to two methods proposed by Lippert *et al.*[188,192], one linear and one identifying epistatic interactions, which both employ linear mixed models (LMM) and are widely used purely statistical analysis tools.

3.2.6 Performance metrics

To assess the performance of COMBI, DeepCOMBI and the baseline methods, a number of statistical metrics are evaluated. The performances of both the intermediate classification step (*i.e.* either SVM or DNN) and the final step of predicting informative SNPs need to be explored.

Assuming we know the ground truth, the metrics are defined as follows:

- Number of true positives = TP; False positives = FP; True negatives = TN; False negatives = FN
- Accuracy = (TP + TN)/(TP + TN + FP + FN)
- Precision = TP/(TP + FP)
- True-positive rate TPR = TP / (TP + FN)
- False-positive rate FPR = FP / (TP + FN)
- Family-wise error rate FWER = P(FP >=1)
- Expected number of false rejections ENFR = E(FP)
- Balanced accuracy = (TPR + TNR)/2

The following performance curves and the area under these curves (AUC) are investigated:

- Receiver operating characteristic curve (ROC): Power vs. error rate, *e.g.* TPR vs. FPR, TP vs. FP, TPR vs. FWER or TPR vs. ENFR for varying thresholds
- Precision-recall curve (PR): Precision vs. power, *e.g.* Precision vs. TPR or Precision vs. TP for varying thresholds

3.3 Results

In the following sections, we present the results of the proposed COMBI and DeepCOMBI methods evaluated on generated as well as on real-world data. Performance in terms of both classification accuracy and SNP prediction is examined in comparison to a number of baseline methods, which are presented in full detail in **Chapter 3.2.5**. As evaluation criteria, we report prediction accuracy for the classification step and *FWER*, precision and *TPR* for the SNP selection step. See **Chapter 3.2.6** from above for a detailed description of the evaluated performance metrics.

3.3.1 Results on generated datasets

Here, we report the results averaged over 1,000 datasets generated in the simulation process described in **Chapter 3.2.3** ("*Validation on generated datasets*"). We show that on these datasets, COMBI and DeepCOMBI perform better than the traditionally used method for analyzing GWAS, RPVT, and other competitor methods. In addition, it is demonstrated that the deep learning-based approach DeepCOMBI outperforms the SVM-based method COMBI.

Prediction performance on generated datasets

The first steps of both COMBI and DeepCOMBI consist of training a learning algorithm for the classification of subjects into their respective phenotypic group given their genotypic information. Since all following steps depend on the performance of these classifiers, high prediction accuracy is crucial. On the generated datasets, the SVM (as part of the COMBI method) achieves 59% accuracy on average and 54% balanced accuracy. In comparison, the DNN (as part of the DeepCOMBI method) performs significantly better with an average of 74% classification accuracy. It also achieves higher balanced accuracy (74%) by counteracting against the negative effects of unbalanced datasets through the application of class weights in the DNN training process. All accuracy scores and summary statistics are presented in **Table 2**.

Table 2. Classification performances of COMBI and DeepCOMBI on generated datasets from Mieth *et al.* (2020)[15]. Summary statistics of the classification accuracy of the SVM (as in the first step of COMBI) and the DNN (as in the first step of DeepCOMBI) are presented. Values corresponding to accuracy and balanced accuracy in parentheses are given.

	Mean accuracy (balanced accuracy)	Standard deviation of accuracy (balanced accuracy)	Minimum of accuracy (balanced accuracy)	Maximum of accuracy (balanced accuracy)
SVM (as in COMBI)	0.59 (0.54)	0.05 (0.06)	0.41 (0.35)	0.76 (0.71)
DNN (as in DeepCOMBI)	0.74 (0.74)	0.07 (0.07)	0.55 (0.50)	0.97 (0.98)

Following these promising intermediate results, we investigate whether the entire workflow of the DeepCOMBI method can outperform COMBI and the other baseline methods in terms of SNP prediction in the next sections.

SNP selection performance on generated datasets

To provide some intuitive understanding of the advantages of the novel methods and to compare the relevance scores and *p*-values obtained with the LRP-based method DeepCOMBI to those derived from the SVM weights in the COMBI method, we look at three exemplary synthetic datasets and the corresponding results (See **Figure 13**). They can be distinguished by the level of association of the 20 informative SNPs (highlighted in pink) with the phenotype. In the first column of subfigures, the strength of association for each replication at positions 5001 - 5020 is shown in the corresponding RPVT Manhattan plots. While the first row of subfigures represents a replication with very weak associations (small pink tower), the second has a moderate association (medium-sized tower) and the third shows a very strong association (large tower). In the second and third columns, the raw SVM weights and LRP scores are shown. For strong associations (bottom row), both COMBI and DeepCOMBI not only precisely identify the correct tower but also flatten out any noise SNPs, even when - by chance - they achieve considerably high significance. The methods thus not only increase the probability of finding the correct tower but also, and potentially more importantly, decrease the probability of falsely selecting a noise tower.

Furthermore, It can be seen that LRP yields clearer relevance distributions in comparison to the SVM-based method. Even with the huge number of trained parameters, the explanation scores of DeepCOMBI yield a lot less noise than the SVM weights of COMBI. This results in the COMBI method only being able to classify the very strong association correctly (third row of subfigures in **Figure 13**) while it misses the weak and moderate ones. In contrast, DeepCOMBI is successful for both the second and third replication with moderate and strong associations and only misses the very weak association (last row of subfigures in **Figure 13**). Please note that moderate associations (second row of subfigures in **Figure 13**), again, DeepCOMBI not only identifies the correct informative tower but also filters out a relatively high noise tower at position ~ 600, which - just by chance - achieved a *p*-value $< 10^{-5}$ and is therefore incorrectly classified as an informative locus by RPVT.

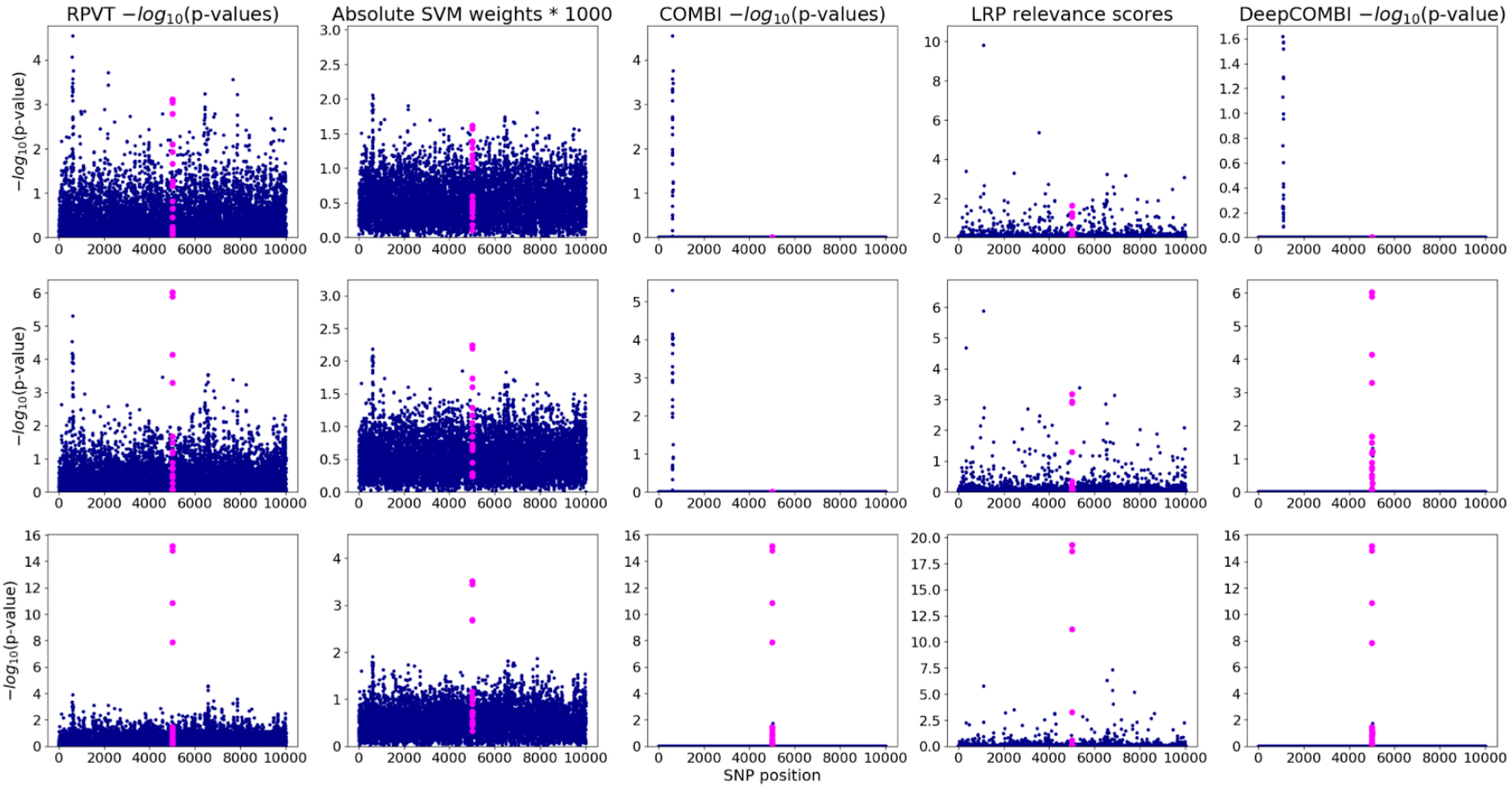

Figure 13: Manhattan plots of three exemplary generated datasets of varying strength of association and corresponding COMBI and DeepCOMBI results from Mieth *et al.* (2020)[15]. The negative logarithmic *p*-values are plotted against position for three exemplary generated datasets: one with weak (first row), one with medium (second row) and one with strong (third row) associations of the 20 informative SNPs at position 5001-5020 (highlighted in pink in all subfigures). Manhattan plots of standard RPVT *p*-values are plotted in the first column of subfigures. Absolute SVM weights and Manhattan plots of the corresponding COMBI *p*-values are shown in the second and third column. Finally, LRP relevance scores and Manhattan plots of the corresponding DeepCOMBI *p*-values are presented in the fourth and last column.

To investigate whether these exemplary findings represent a general trend, we now examine the results of all competitor methods averaged over all 1,000 generated datasets. In **Figure 14**, the corresponding ROC and PR curves are shown. By increasing the significance threshold of each method from very conservative (*i.e.* $t^* = 0$, no significant SNPs) to very liberal (*i.e.* $t^* = \infty$, all SNPs significant), we investigate here how the different methods perform for different levels of error. In both subfigures, COMBI (dark blue line) outperforms RPVT (light blue line) in terms of power (as measured by the *TPR*) and precision.

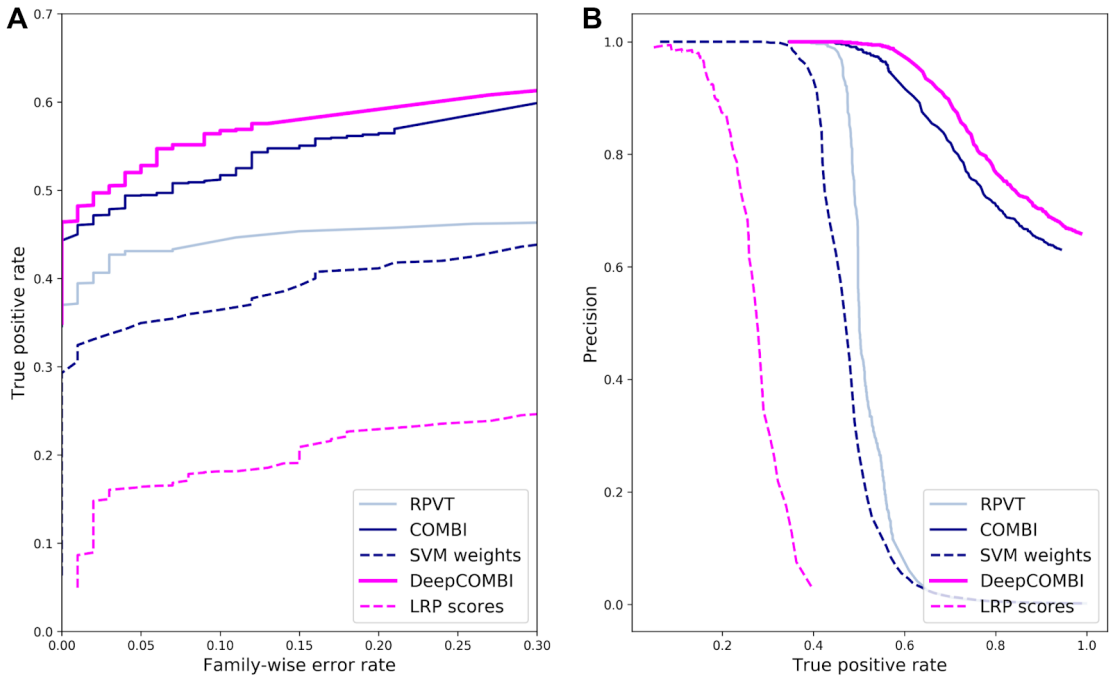

Figure 14: Performance curves of COMBI, DeepCOMBI and competitor methods on generated datasets from Mieth *et al.* (2020)[15]. A ROC curves and **B** PR curves of RPVT, the COMBI method, direct thresholding of SVM weights, the DeepCOMBI method and direct thresholding of LRP scores averaged over the 1,000 generated datasets are shown.

The COMBI method improves performance by correctly filtering out the noise SNPs and identifying the informative SNPs accurately in the selection step of the COMBI method. The identification of SNPs that have an effect on the phenotype in the semi-real datasets can be improved even further by applying the DeepCOMBI method (pink line), which consistently achieves better results than the COMBI approach. The LRP based relevance scores predict more accurately than the SVM weights where the informative SNPs can be found.

The combinatorial approaches, COMBI and DeepCOMBI, also perform better than their individual components of an ML algorithm (*i.e.* SVM or DNN with LRP) and a multiple testing step (*i.e.* RPVT). This can be deduced from the fact that RPVT, as well as the other two baseline methods of directly thresholding the raw LRP scores (dashed

pink line) and SVM weights (dashed dark blue line), cannot achieve the same performance as their combinations (*i.e.* DeepCOMBI and COMBI).

During the development phase of the COMBI method and for the original publication of 2016[10], a number of additional experiments were conducted. The remainder of **Chapter 3.3.1** and **Appendix I** present the corresponding results. Please note that during this initial phase of the dissertation, a slightly different experimental setup for the generated datasets was employed and DeepCOMBI was not introduced yet. For example, 10,000 datasets are generated here instead of 1,000 in the DeepCOMBI publication[15].

To understand the results in more detail, we now investigate in which cases from the original publication (2016[10]) the COMBI method can or cannot increase the performance of the SNP prediction. **Figure 15** shows for all 10,000 synthetic datasets the level of difficulty of the problem (represented by the *TPR* of RPVT) and how well it can be solved using the COMBI method (represented by the gain in *TPR* of the COMBI method over RPVT). In the majority of cases, the COMBI method helps performance, *i.e.* increases the *TPR*. However, it decreases performance in some cases (about 3% of the 10,000 datasets). As expected, those cases represent difficult problems with high noise where the baseline *TPR* of RPVT is very low.

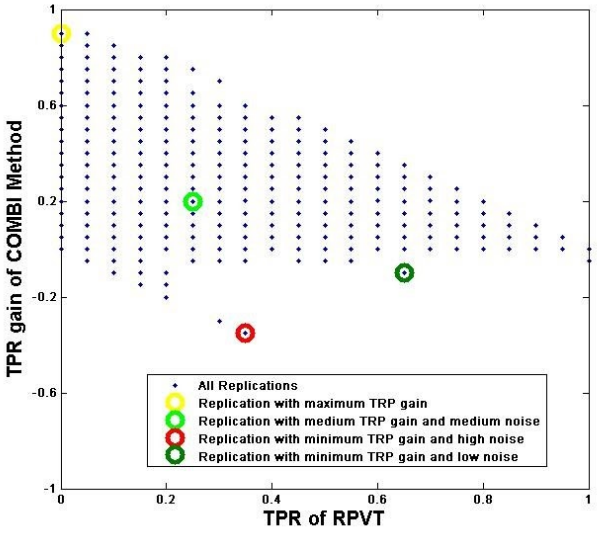

Figure 15: *TPR* gain of the COMBI method against *TPR* of RPVT on generated datasets from Mieth *et al.* (2016)[10]. Each dot represents one dataset and indicates how much is gained in terms of *TPR* by applying the COMBI method instead of RPVT to this specific dataset. *TPR* of RPVT can be interpreted as a measure of the difficulty of the problem. Four replications are highlighted. Three of them represent special cases with extraordinary characteristics (*i.e.* maximum *TPR* gain, minimum *TPR* gain with low and high noise) and the fourth represents an average run with medium *TPR* gain and medium noise. See **Figure 16** for the individual results of those replications.

In order to investigate the different situations to be encountered in a real-world setting, we now analyze a number of special replications. Detailed plots of exemplary runs that either represent an average run (*i.e.* medium *TPR* gain and medium noise) or have extraordinary characteristics (*i.e.* maximum *TPR* gain, minimum *TPR* gain with low and high noise) are shown in **Figure 16**.

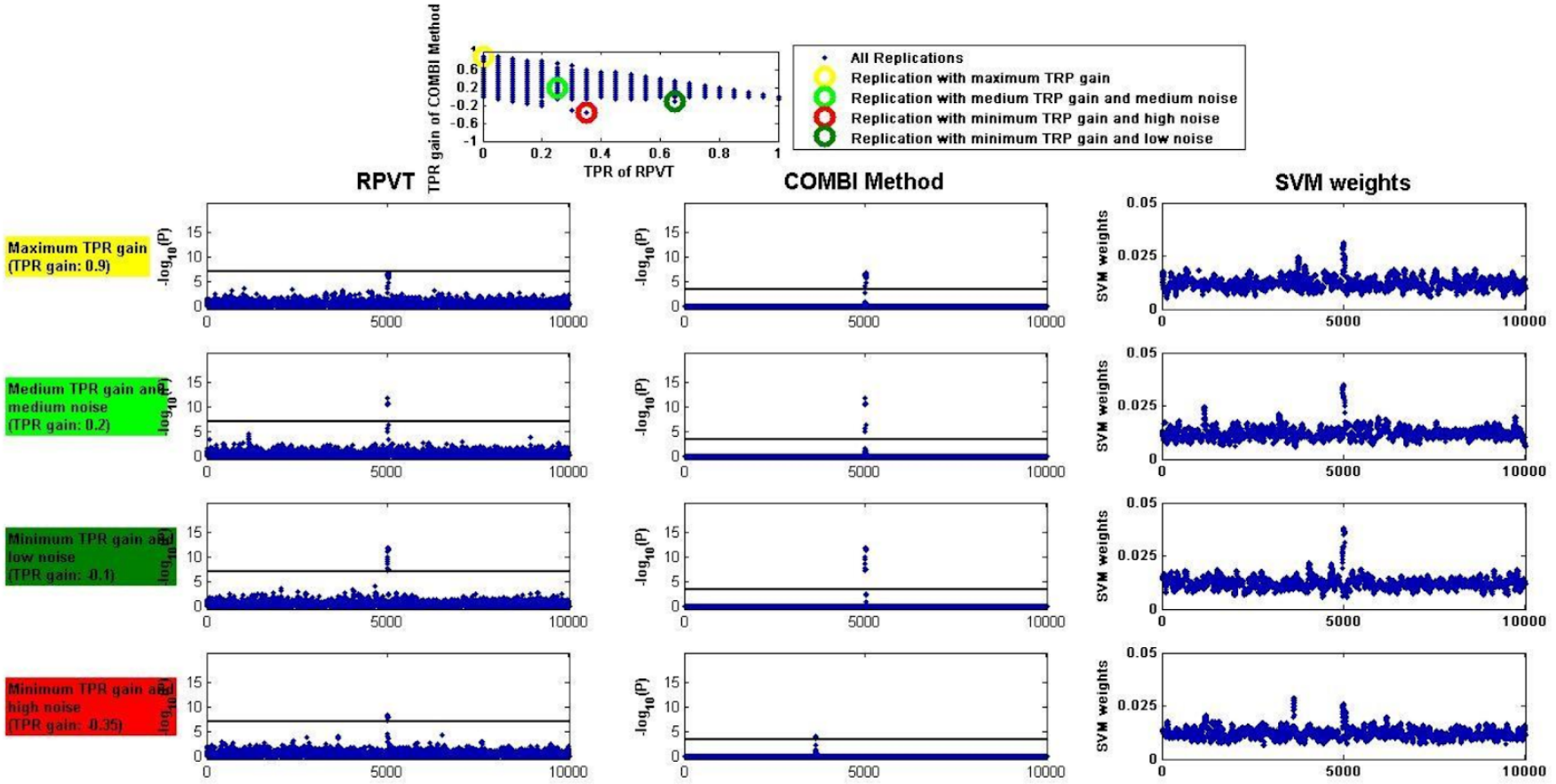

Figure 16: Manhattan plots of four exemplary generated datasets of varying levels of *TPR* gain and corresponding COMBI results from Mieth *et al.* (2016)[10]. The negative logarithmic *p*-values are plotted against position for four exemplary datasets: one with with maximum *TPR* gain (yellow, first row of subfigures), one with medium *TPR* gain and medium noise (light green, second row), one with minimum *TPR* gain and high noise (red, third row) and one with minimum *TPR* gain and low noise (dark green, fourth row). The Manhattan plots of the corresponding RPVT *p*-values (first column) as well as the *p*-values of the COMBI method (second column) and the SVM weights (third column) are presented. Thresholds indicating statistical significance are represented by horizontal lines.

The first row in **Figure 16** represents the replication with maximum *TPR* gain, where the COMBI method performs extremely well. Although there is a lot of noise and the tower of SNPs associated with the phenotype located at positions 5,001-5,020 is not very high in this example, the COMBI method finds it accurately. An average replication with medium noise and medium *TPR* gain, *i.e*. where both methods find the tower and the COMBI method can only moderately help performance, is presented in the second row. The third example illustrates that RPVT is sufficient for very easy problems (*i.e.* low noise and high tower yielding minimum *TPR* gain) and that using the COMBI method does not decrease performance. The most crucial case is presented in the last row. In this worst-case scenario of minimum *TPR* gain and high noise, there is an extremely small tower that is very hard to identify. In addition, there is another high-noise tower with very high SVM weights. In contrast to the RPVT approach, which identifies the correct tower, the COMBI method selects the wrong tower. This example shows that the COMBI method selects the wrong towers in very few cases.

We now investigate whether these pathological cases can be identified *a priori*. As observed in **Figure 16**, they are characterized by high noise levels, indicating that very hard problems must be solved. Identifying the datasets where an SVM trained for the classification of subjects does not have high accuracy is an intuitive idea. The COMBI method would be expected to fail in those cases, which is indeed what can be seen when investigating the SVM accuracies for each replication in **Figure 17**. As expected, the problematic cases - where power is lost with the COMBI method - are characterized by low SVM classification accuracies. These cases can thus roughly be estimated in advance and a measure of trust in the results of the COMBI method could be reported along with the results of the COMBI method for each dataset.

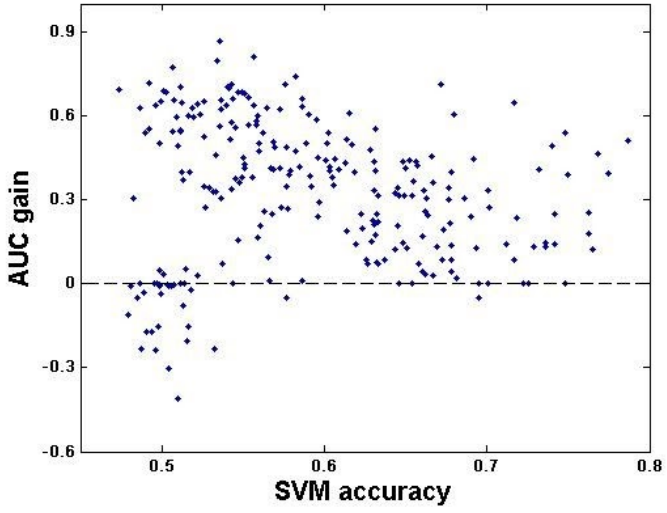

Figure 17: AUC gain of the COMBI method against SVM accuracy on generated datasets from Mieth *et al.* (2016)[10]. Each dot represents one dataset and indicates how much is gained in terms of *AUC* of the ROC curve by applying the COMBI method for varying degrees of SVM accuracy. Negative AUC gain marks the cases where power is lost in comparison to RPVT.

Comparison to other baseline methods on generated datasets

In addition to comparing the proposed methods with the RPVT approach, we investigate here whether slight alterations and simplifications of the COMBI method can achieve the same level of effectiveness and also examine other appropriate state-of-the-art algorithms. Since DeepCOMBI was previously shown to outperform COMBI on the generated datasets, it is sufficient here to only look at the performance of the latter in order to show that both proposed methods outperform the competitor methods.

Beginning with the investigation of simplifications of the COMBI method, we show now that applying a moving average filter to the SVM weights prior to the selection step is crucial to significantly improve performance. Observe in **Figure 18** that the COMBI method cannot increase the power or precision of RPVT at all without this filtering step. As a consequence, one might suspect that the filtering step alone and not the SVM screening step is responsible for improving performance in comparison to RPVT. To refute this thesis, we apply the moving average filter to the *p*-values in *log*-space and then employ RPVT in the original *p*-value space. **Figure 18** illustrates that this decreases the performance of the RPVT method and thus cannot reach the effectiveness of the COMBI method. In conclusion, the filter is not the only effective tool in COMBI and both screening approaches (*i.e.* SVM and filter screening) are crucial in order to achieve high performance rates.

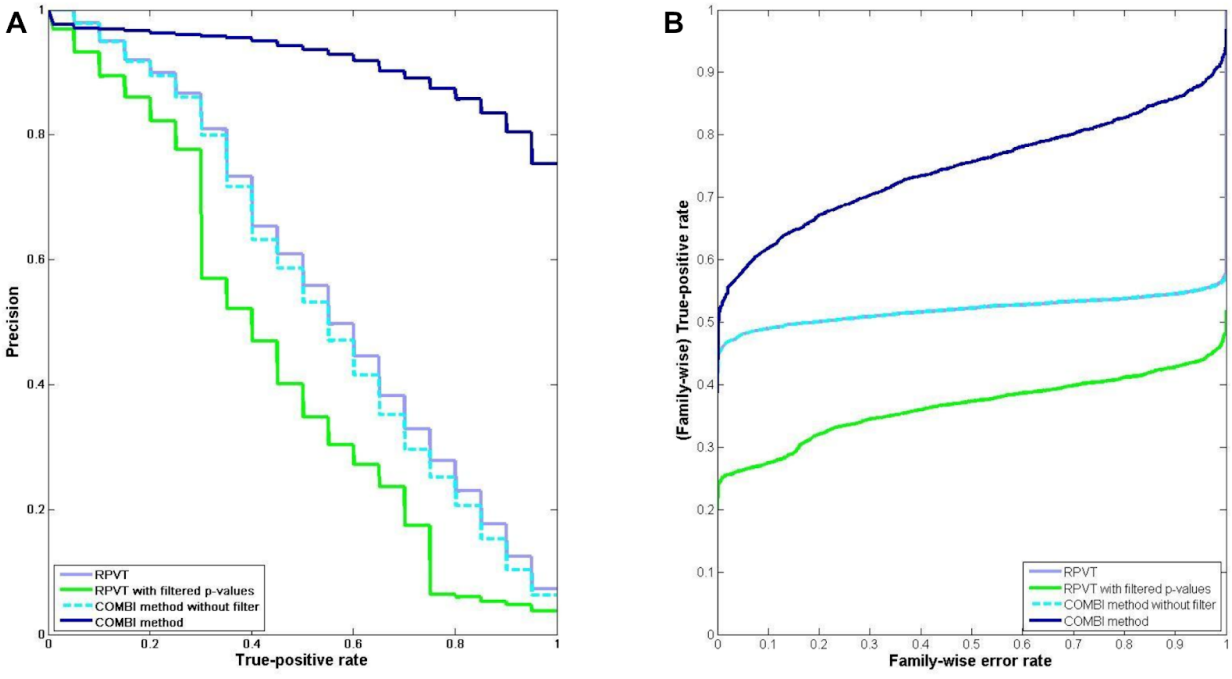

Figure 18: Performance curves of COMBI, RPVT and modifications thereof on generated datasets from Mieth *et al.* (2016)[10]. A PR curves and **B** ROC curves of RPVT, RPVT with a filter directly applied to the *p*-values on log-scale, the COMBI method without the filter and the complete COMBI method, including the filter, averaged over the 10,000 generated datasets are shown.

Besides checking whether easy simplifications of the COMBI method achieve the same effect, the performances of other competitor methods are investigated. There are a number of related ML methods, out of which we select three as representatives to be compared to the COMBI method in this simulation setup. See **Figure 19** for the corresponding results.

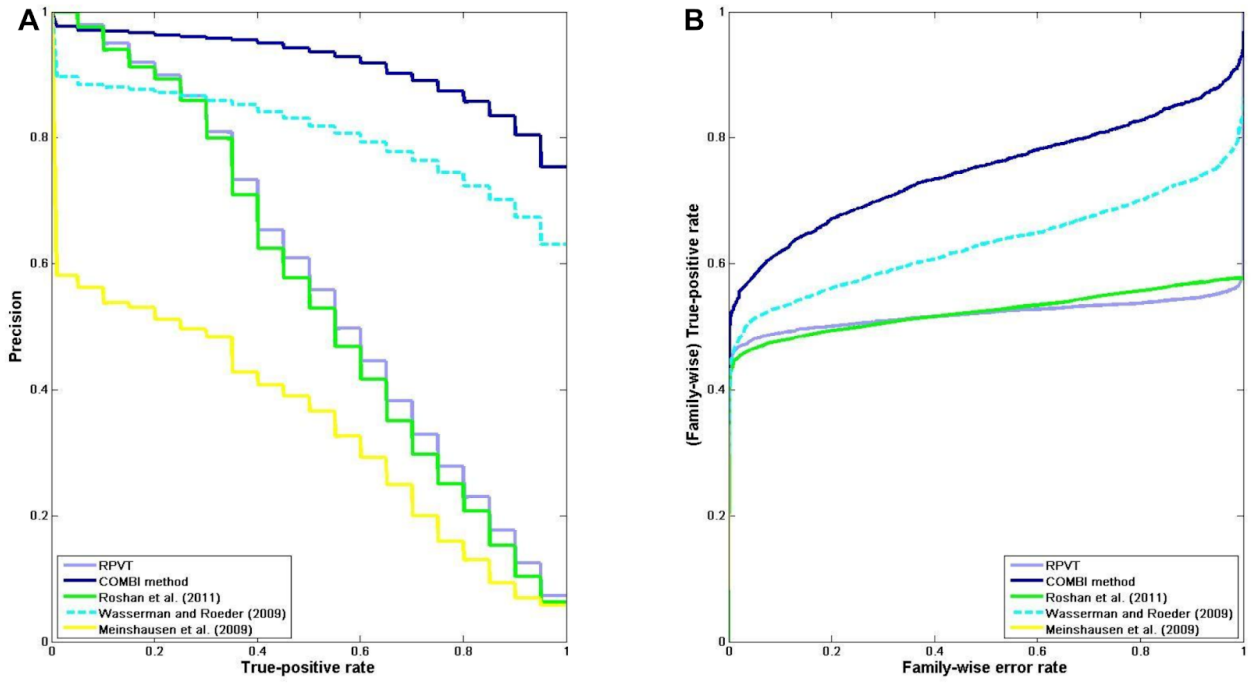

Figure 19: Performance curves of COMBI, RPVT and additional competitor methods on generated datasets from Mieth *et al.* (2016)[10]. A PR curves and **B** ROC curves of RPVT, the COMBI method, the reversed-order COMBI method proposed by Roshan *et al.*[62], the single-split COMBI method proposed by Wasserman and Roeder[210] and the multi-split COMBI method proposed by Meinshausen *et al.*[211] over the 1,000 generated datasets are shown. Please note that in **B**, the yellow curve is hardly visible because it is almost identical to the green one.

The most closely related method proposed by Wasserman and Roeder[210] separates the two steps of the COMBI method, *i.e.* ML and statistical testing, from each other in terms of the data they use in each step. First, they randomly select half of the datapoints (*i.e.* half of the individuals) and train an SVM to identify the k SNPs with the highest corresponding weights. In the second step, p-values are computed on the other half of the dataset, considering only the SNPs identified in the previous step. Even though the significance threshold α can now simply be corrected with $\frac{\alpha}{k}$ (which is much less conservative than the Bonferroni correction $\frac{\alpha}{d}$ on the complete dataset considering $k << d$), this method comes with a great loss of power, where the corresponding curves are constantly below the curve of the COMBI method.

Meinshausen *et al.*[211] present an extension of the method proposed by Wasserman and Roeder[210]. Instead of splitting the data once into two sets and using one for SVM training and the other for statistical testing, they suggest aggregating the results of multiple random splits, arguing that this would decrease error rates and increase power.

They propose using quantiles as summary statistics for the *p*-values of the multiple splits. In our simulation, this method does not reach the performance levels of the COMBI method and also fails to reach that of the RPVT approach.

To summarize, the results here indicate that it is more effective to use the full dataset for both the selection of candidate SNPs and multiple testing (as in the COMBI method), rather than using a subset for selection and another subset for testing (as in the single- and multi-split methods by Wasserman and Roeder[210] and Meinshausen *et al.*[211]).

Another method for identifying associated regions was proposed by Roshan *et al.*[62]. It consists of a statistical testing step, where the top χ^2-ranked SNPs are selected and put into the next step to train an SVM. The SVM weights are then used directly as a test statistic. This basically boils down to a version of the COMBI method, where the order of the two steps (SVM training and statistical testing) is reversed. Applying this method to our 10,000 simulated datasets yields no gain in performance, suggesting that it is important to select SNPs for multiple testing according to their relevance in SVM prediction (as in the COMBI method) rather than selecting SNPs according to their *p*-value for SVM training (as Roshan *et al.* propose).

3.3.2 Results on WTCCC data

Here, we report the results of the proposed methods on the seven WTCCC datasets described in **Chapter 3.2.3** (“*Validation on WTCCC data”*). On these datasets, we compare the performance of COMBI and DeepCOMBI to that of the traditionally used method, RPVT, and other competitor methods. We also show that the deep learning-based approach DeepCOMBI outperforms the SVM-based method COMBI.

Prediction performance on WTCCC data

In order to perform well in the SNP selection step, both COMBI and DeepCOMBI depend on high accuracies in the prediction steps. In **Figure 20**, we present the classification performances on all diseases and chromosomes of the WTCCC dataset of the SVM as used in the first step of the COMBI method and of the DNN as used in the first step of the DeepCOMBI method. The DNN of DeepCOMBI performs consistently better than the SVM of COMBI in terms of all four validation metrics described in **Chapter 3.2.6**.

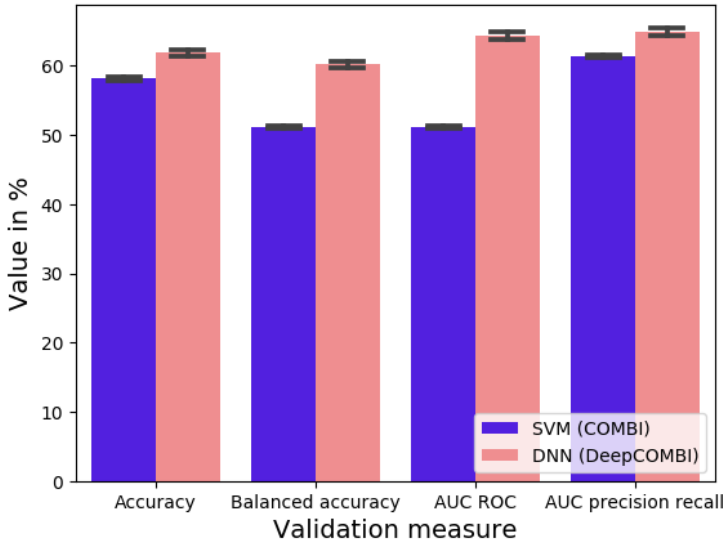

Figure 20: Classification performance on WTCCC data from Mieth *et al.* (2020)[15]. Mean validation measures of the SVM (as in the first step of COMBI) and the DNN (as in the first step of DeepCOMBI) averaged over all diseases and chromosomes are given with standard deviation. All seven datasets are split into 80% training and 20% validation data.

SNP selection performance on WTCCC data

In the following section, we first present the SNP selection results of the COMBI method applied to the WTCCC dataset as published in Mieth *et al.* 2016[10]. Afterwards, we present the SNP selection results of the DeepCOMBI method as published in Mieth *et al.* 2020[15]. Please note that for the latter publication, COMBI is re-applied to the same dataset and due to the nondeterministic nature of the permutation procedure, slightly

different results are obtained. Hence, DeepCOMBI is compared to the re-calculations of COMBI, not necessarily to those of the original COMBI publication. In addition, the GWAS catalog - which is the basis for all validation procedures - included a very different (to be specific a much smaller) set of associations in 2015 than in 2020, which - without loss of generality - causes some of the DeepCOMBI performance curves to look slightly different than the ones presented first in the COMBI figures. Please refer to **Appendix Chapter II.** for an investigation of internal stability of the COMBI method, where we find that the COMBI method produces more stable results than RPVT.

Please also note that the RPVT results correspond to the χ^2 *p*-values we have calculated here, not necessarily to the original WTCCC publication, where they also investigate Cochran-Armitage trend test *p*-values and presumably apply slightly different preprocessing steps.

The COMBI method on WTCCC data

Figure 21 displays Manhattan plots for all seven diseases resulting from the standard RPVT approach (left) and the COMBI method (center), as well as the corresponding SVM weights (right). In each Manhattan plot, the negative logarithmic *p*-values of all SNPs at a given position in a chromosome are shown. Chromosomes are shown in alternating colors for clarity. SNPs that show genome-wide statistical significance are highlighted in green in the left and right panel and all statistically significant SNP associations are highlighted in green. For standard RPVT, the threshold indicated by the horizontal dashed line is fixed *a priori* genome-wide to $t^* = 1 \times 10^{-5}$ (*i.e.* 5 in the plot). For the COMBI method, it is determined chromosome-wise via the permutation-based threshold over the whole COMBI procedure described in **Chapter 3.2.2.** to match the ENFR of RPVT as described in **Chapter 3.2.4**.

The center and right graphs illustrate that the COMBI method discards SNPs with low SVM scores. Hence, the *p*-values for such SNPs are set to one without performing a statistical test, thereby drastically reducing the number of candidate associations. In contrast, the RPVT method results in *p*-values based on a formal significance test for every SNP, where many of these *p*-values are small by chance and produce a lot of statistical noise.

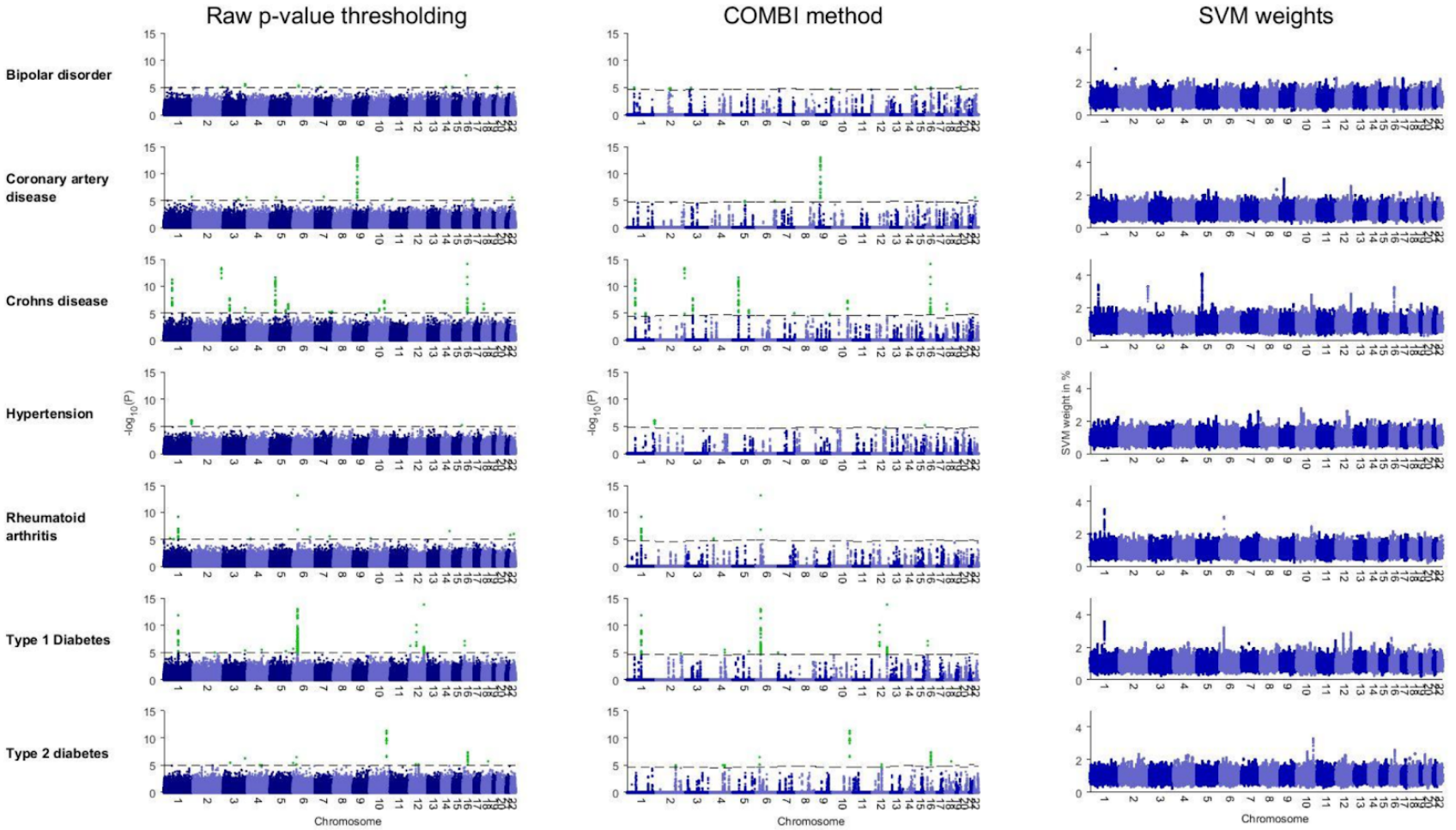

Figure 21: Manhattan plots of seven WTCCC datasets and corresponding COMBI results from Mieth *et al.* (2016)[10]. The negative logarithmic *p*-values are plotted against position on each chromosome for all seven diseases. The Manhattan plots of the corresponding RPVT *p*-values (first column) as well as the *p*-values of the COMBI method (second column) and the SVM weights (third column) are presented. Thresholds indicating statistical significance are represented by dashed horizontal lines and significant *p*-values are highlighted in green. Please note that the y-axes of all plots have the same limits (0 to 15 for *p*-values and 0 to 4 for SVM weights) to enable direct comparison.

In **Table 3**, we present all significant associations reported by the COMBI method. Besides showing basic information (associated disease, chromosome, identifier and χ^2 *p*-value) for all of these SNPs, the fifth column indicates whether they are found to be significant by RPVT with the application of $t^* = 1 \times 10^{-5}$. Associations with a raw *p*-value $>10^{-5}$ are not reported using only RPVT. If they are selected by the COMBI method, we consider them to be new findings. To validate all reported associations, the sixth and seventh columns of **Table 3** report whether - and if so in which external study - they have been found significantly associated with the given disease according to the GWAS catalog. By investigating whether the identified SNPs have been discovered as significant in an independent GWAS published after the original WTCCC study, it can be determined whether those novel findings can be confirmed to be true associations. The COMBI method finds 46 significant locations. 34 of these 46 significant locations have a *p*-value below 10^{-5} and are thus also found by the RPVT approach. Crucially, the COMBI method finds 12 *additional* SNPs. Out of these, ten (>83%) have already been replicated in later GWAS or meta-analyses. The COMBI discoveries that have been replicated independently using individual SNP testing are for bipolar disorder rs2989476 (Chr. 1), rs1344484 (Chr. 16), rs4627791 (Chr. 3) and rs1375144 (Chr. 2); for coronary artery disease rs6907487 (Chr. 6) and rs383830 (Chr. 5); for Crohn's disease rs12037606 (Chr. 1), rs10228407 (Chr. 7) and rs4263839 (Chr. 9) and for type 2 diabetes rs6718526 (Chr. 2). Given the current debate on the replicability of GWAS findings obtained by single-SNP analyses[226], it is remarkable that GWAS published later have already replicated more than 83% of novel SNPs the COMBI method detects by reanalyzing data published in 2007.

Table 3 : Significant SNPs of the COMBI method on seven WTCCC datasets and related association details from Mieth *et al.* (2016)[10]. For each SNP identifier on a specific chromosome that is found to be significantly associated with a disease by the COMBI method in Mieth *et al.* (2016)[10], the corresponding χ^2 test *p*-value is shown and it is indicated whether the RPVT *p*-value is $< 1 \times 10^{-5}$ (*i.e.* the SNP is a significant finding of RPVT as well) and whether the SNP has been found significant with a *p*-value $< 1 \times 10^{-5}$ in an external study with a corresponding PubMed identification number (PMID). Please note that the RPVT result in the fifth column corresponds to the χ^2 *p*-values calculated here, not necessarily to the original WTCCC publication, where they investigate Cochran-Armitage trend test *p*-values and presumably apply slightly different preprocessing steps.

Disease	Chromosome	Identifier	χ^2 *p*-value	Significant in RPVT	*p*-value < 10^{-5} in at least one ext. GWAS	References (PMID)
Bipolar disorder (BD)	1	rs2989476	1.05e-05		YES	19416921
	2	rs1375144	1.26e-05		YES	21254220
	2	rs7570682	1.77e-06	YES	YES	21254220
	3	rs4627791	1.18e-05		YES	21254220
	14	rs11622475	8.02e-06	YES	YES	21254220
	16	rs1344484	1.10e-05		YES	21254220
	9	rs7860360	1.82e-06	YES		
	20	rs3761218	7.15e-06	YES	YES	21254220
Coronary artery disease (CAD)	5	rs383830	1.35e-05		YES	21804106
	6	rs6907487	1.22e-05		YES	17634449
	9	rs1333049	1.12e-13	YES	YES	21606135
	22	rs688034	2.75e-06	YES		

Disease	Chromosome	Identifier	χ^2 *p*-value	Significant in RPVT	*p*-value < 10^{-5} in at least one ext. GWAS	References (PMID)
Crohn's disease (CD)	1	rs11805303	6.35e-12	YES		
	1	rs12037606	1.02e-05		YES	17554261
	2	rs10210302	4.52e-14	YES	YES	23128233
	3	rs11718165	2.04e-08	YES	YES	21102463
	5	rs6596075	3.11e-06	YES		
	5	rs17234657	2.42e-12	YES	YES	18587394
	7	rs10228407	1.08e-05			
	9	rs4263839	1.61e-05		YES	21102463
	10	rs10883371	5.23e-08	YES	YES	21102463
	16	rs2076756	7.55e-15	YES	YES	21102463
	18	rs2542151	1.93e-07	YES	YES	18587394
Hypertension (HT)	1	rs2820037	7.41e-07	YES		
	12	rs11110912	1.58e-05			
	15	rs2398162	6.01e-06	YES		
Rheumatoid arthritis (RA)	1	rs6679677	<1.0e-15	YES	YES	20453842
	4	rs3816587	7.28e-06	YES		
	6	rs9272346	7.38e-14	YES		
Type 1 diabetes (T1D)	1	rs6679677	<1.0e-15	YES	YES	19430480
	2	rs231726	1.43e-06	YES		
	4	rs17388568	3.07e-06	YES	YES	21829393
	5	rs17166496	5.97e-06	YES		
	6	rs9272346	<1.0e-15	YES	YES	18978792
	7	rs6950410	1.03e-05			
	12	rs17696736	1.55e-14	YES	YES	18978792
	12	rs11171739	8.36e-11	YES	YES	19430480
	16	rs12924729	7.86e-08	YES	YES	17554260
Type 2 diabetes (T2D)	2	rs6718526	1.00e-05		YES	20418489
	4	rs1481279	9.44e-06	YES		
	4	rs7659604	9.61e-06	YES		
	6	rs9465871	3.38e-07	YES		
	10	rs4506565	5.01e-12	YES	YES	23300278
	12	rs1495377	7.21e-06	YES		
	16	rs7193144	4.15e-08	YES	YES	22693455
	18	rs1025450	1.98e-06	YES		

Two out of the 12 SNPs with *p*-values exceeding 1×10^{-5} have not yet been reported in any GWAS or meta-analyses as being associated with the corresponding diseases. Those are rs11110912 (Chr. 12) for hypertension and rs6950410 (Chr. 7) for type 1 diabetes. SNP rs11110912 is included in the original WTCCC analysis, but a *p*-value higher than 1×10^{-5} is obtained (1.94×10^{-5})[38], so it was not collected in the GWAS Catalog. SNP rs6950410 has been detected as associated with multiple complex diseases[234]. Regarding the biological plausibility of these two non-replicated SNPs, we examine a number of functional indicators to assess their potential role in disease (See **Table 4**). In particular, we explore the genomic regions in which they map, their potential roles as regulatory SNPs, their status as expression quantitative trait loci (eQTL) and their role in Mendelian disease.

Table 4: Functional analysis of unreplicated SNPs from WTCCC datasets detected by the COMBI method from Mieth *et al.* (2016)[10]. The two SNPs - detected by the COMBI method - which, according to **Table 3**, were not replicated in a subsequent independent GWAS are functionally analyzed.

	SNP detected by COMBI	
	rs11110912	**rs6950410**
Disease	HT (Hypertension)	T1D (Type 1 diabetes)
Chr / Position (hg19)	12:102042213	7:4038917
Functional consequence	Intronic MYBPC1 (myosin binding protein C)	Intronic SDK1 (sidekick cell adhesion molecule 1)
OMIM (Role in disease evidence of the gene the associated SNP lies in (available at http://omim.org/)**)**	Yes (involved in familial hypertrophic cardiomyopathy)	-
GWAS Catalog (Presence in the "*reported gene*" field in the GWAS Catalog (http://www.genome.gov/gwastudies/)**)**	No	No
Genes (in 200 Kb window**)**	MYBPC1, CHPT1, SYCP3	SDK1
eQTL activity (*p*-value) (Evidence about the activity as eQTL in blood (gathered from the "*Blood eQTL browser"*; http://genenetwork.nl/bloodeqtlbrowser/)**)**	CHPT1 (P<10-8)	-
RegulomeDB (RegulomeDB. Summary of DNA regulatory evidence (in http://regulomedb.org/)**)**	1d ("*strong*")	-
Haploreg (Noncoding regulatory evidence of the haplotype block (www.broadinstitute.org/mammals/haploreg/haploreg.php)**)**	Transcription factor activity *(BATF,PU1)*	DNAse activity (in Osteoblasts)

Overall, there is no strong evidence of functional roles for the two non-replicated SNPs, but SNP rs11110912 (Chr. 12), for which COMBI suggests a link to hypertension, is an intronic SNP mapping on a gene, MYBPC1, that has been previously linked to familial hypertrophic cardiomyopathy, suggesting that COMBI has given rise to another interesting true-positive finding.

Instead of investigating the significant findings of the two methods achieved by matching a specific error level, we now examine the performance of those methods for different levels of error. **Figure 22** shows the ROC and PR curves that have been generated based on the replication of SNPs according to the GWAS catalog (here, due to the absence of basic truth knowledge, replicated reported associations are again counted as true positives and non-replicated associations as false-positives). The COMBI method outperforms the RPVT approach for different type 1 error levels. As the dark blue lines are consistently above the light blue lines, the COMBI method achieves both higher numbers of *true positives* (*i.e*. higher TPR) as well as a higher *precision* (*i.e.*

proportion of replicated associations amongst the SNPs classified as associated with the trait) for given numbers of *false and true positives* than RPVT for almost all levels of error. For comparison, we also show the result achieved when selecting SNPs based on the highest SVM weights in absolute value (after filtering). The results show that discarding either one of the two steps in the COMBI method (ML or statistical testing step) leads to a decrease in performance. Please note that the COMBI lines end at some point and the RPVT and the raw SVM lines continue. At the endpoint of the COMBI curve, all SNPs selected in the SVM step are also significant in the statistical testing step; *i.e.* if one wanted to add just one more SNP to the list of reported associations, all other SNPs would also become significant, as they have a *p*-value of 1.

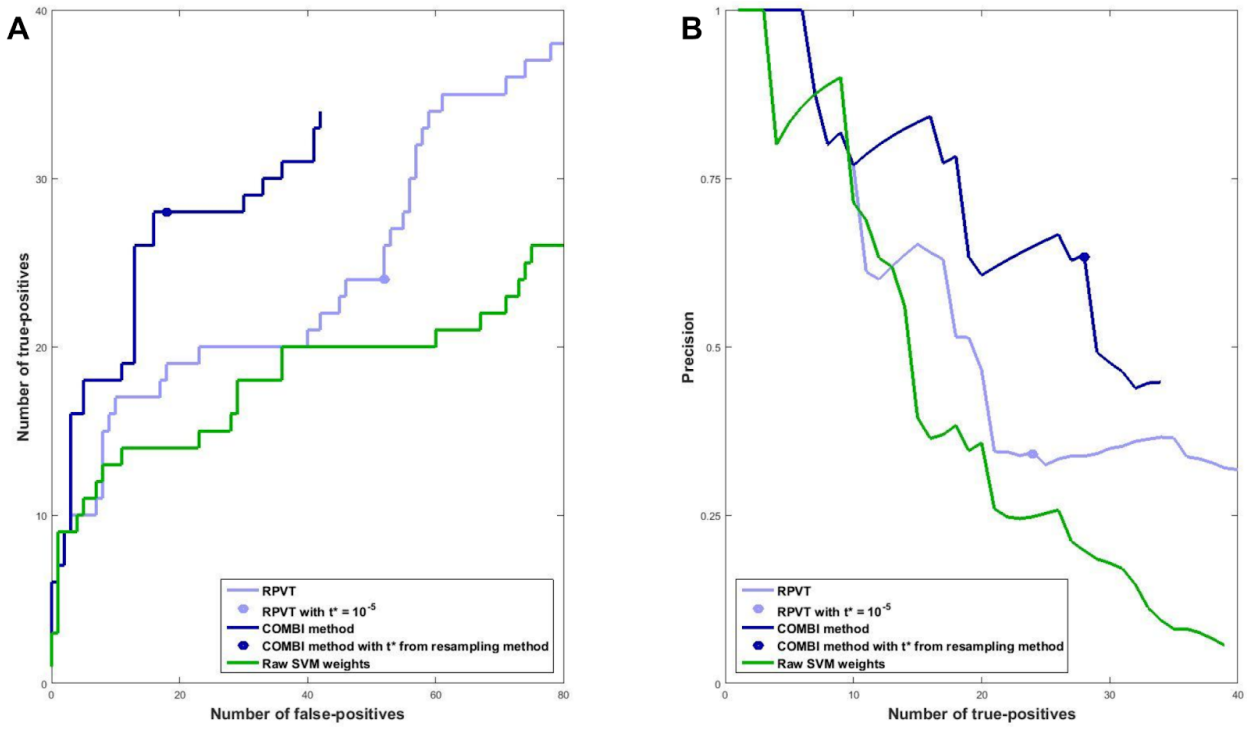

Figure 22: Manhattan plots of seven WTCCC datasets and corresponding COMBI results from Mieth *et al.* (2016)[10]. The negative logarithmic *p*-values are plotted against position on each chromosome for all seven diseases. The Manhattan plots of the corresponding RPVT *p*-values (first column), as well as the *p*-values of the COMBI method (second column) and the SVM weights (third column) are presented. Thresholds indicating statistical significance are represented by dashed horizontal lines and significant *p*-values are highlighted in green. Please note that the y-axes of all plots have the same limits (0 to 15 for *p*-values and 0 to 4 for SVM weights) to enable direct comparison.

We now investigate the points on the curves that correspond to the application of $t^* = 1 \times 10^{-5}$ in the case of RPVT and to the values of t^* resulting from the permutation-based method in the case of the COMBI method in more detail. See **Table 5** for the numbers corresponding to those points, which summarize the findings of **Table 3** and a potential equivalent list of the significant findings of RPVT. A total of 78 SNPs are found to be significant with RPVT, which only performs the statistical testing step and only 46 with the COMBI method since it has the additional layer of the ML

screening step prior to statistical testing. Although the COMBI method finds fewer SNPs than RPVT, the number of replicated SNPs is greater (28 in contrast to 24 of RPVT). The COMBI method also classifies only 18 unreplicated SNPs as associated with the trait (yielding a precision of 61%). This is in contrast to RPVT, which classifies 52 unreplicated SNPs as associated with the trait (yielding a precision of only 32%). In other words, if both methods are calibrated with respect to the same type I error criterion, the COMBI method reports significantly more replicated associations (Fisher's exact test *p*-value of 0.0014).

Table 5: Quantitative summary of the significant findings of RPVT and COMBI from Mieth *et al.* (2016)[10]. For each method, the number of replicated and unreplicated hits (*i.e.* the number of true and false positives) as well as precision and error rates are presented. A pairwise test for the null hypothesis of identical distributions for COMBI and RPVT is performed and the corresponding *p*-value is given. The table represents the information given by the points on the RPVT and COMBI lines in **Figure 22**.

	Number of SNPs reaching significance applying	
	RPVT	**COMBI Method**
SNPs that have achieved < 10^{-5} in at least one external study	24 (32% precision)	28 (61% precision)
SNPs that have not achieved < 10^{-5} in an external study	52 (68% error)	18 (39% error)
Overall	76	46
***p*-value (one-sided Fisher's exact test)**	0.0014	

The DeepCOMBI method on WTCCC data

We now present the results of the DeepCOMBI method as published in Mieth *et al.* 2020[15]. For this publication, COMBI is re-applied to the same dataset as a competitor method and due to the nondeterministic nature of the permutation procedure, slightly different results are obtained. DeepCOMBI is, therefore, now compared to the re-calculations of COMBI, not necessarily to those of the original COMBI publication. In addition, the GWAS catalog - which is the basis for all validation procedures - included a very different (to be specific a much smaller) set of associations in 2015 than in 2020, which - without loss of generality and amongst other things - causes some of the performance curves to look slightly different than in previously presented figures. Please refer to **Appendix Chapter II** for an investigation of the internal stability of the COMBI method.

In **Figure 23**, we present the results of the traditional RPVT approach, the (re-calculated) COMBI method and the DeepCOMBI method applied to the seven diseases of the WTCCC 2007 dataset. While RPVT assigns *p*-values smaller than one (*i.e.* nonzero in the plots on a logarithmic scale) to all SNPs and, in consequence, produces a lot of statistical noise, both COMBI and DeepCOMBI discard most SNPs by assigning *p*-values of one (*i.e.* zero in the plot on a logarithmic scale) and hence reduce the level of noise significantly. The COMBI method selects 100 SNPs with high SVM weights per chromosome and DeepCOMBI chooses 200 SNPs with high LRP scores.

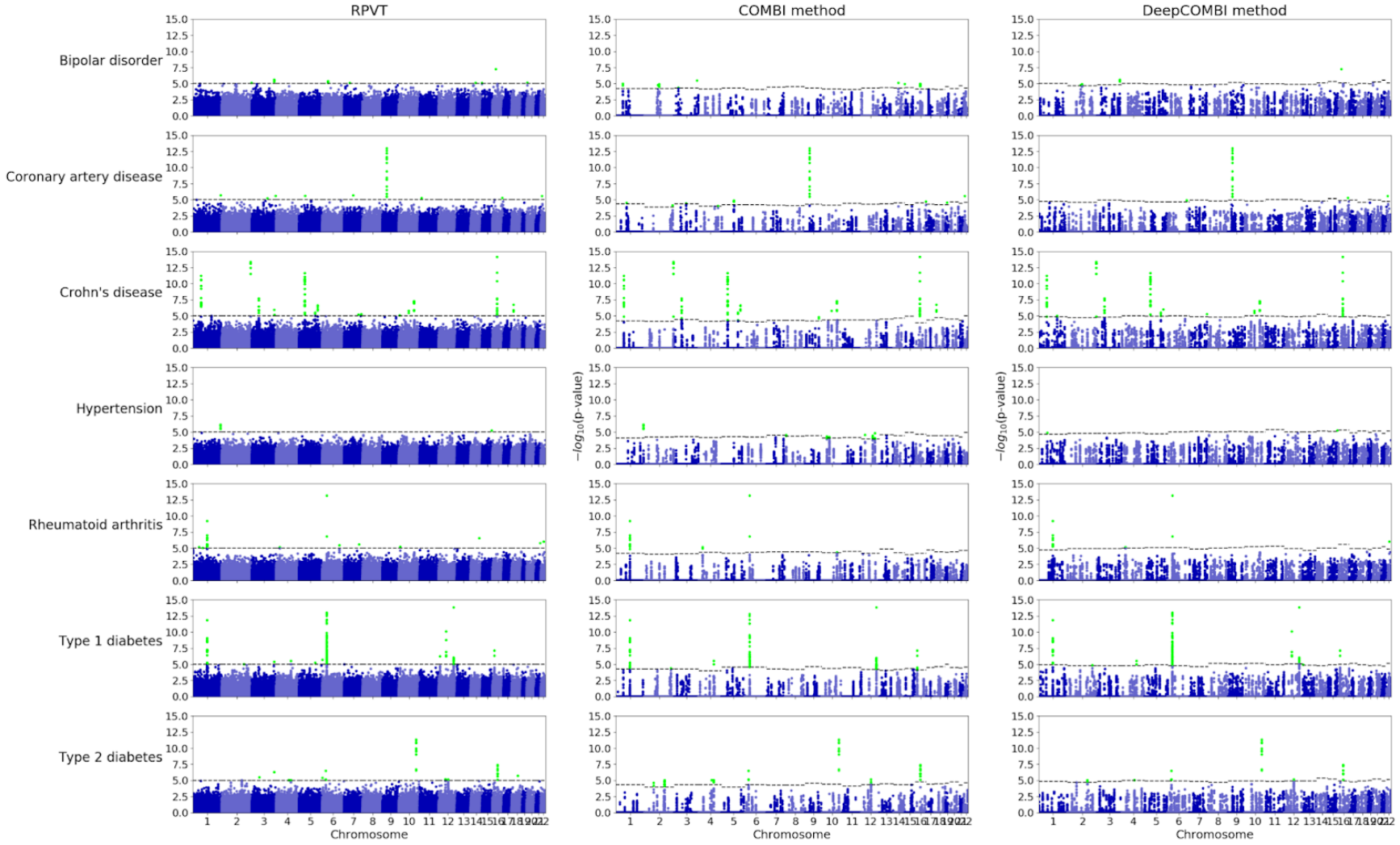

Figure 23: Manhattan plots of seven WTCCC datasets and corresponding COMBI and DeepCOMBI results from Mieth *et al.* (2020)[15]. The negative logarithmic *p*-values are plotted against position on each chromosome for all seven diseases. The Manhattan plots of the corresponding RPVT *p*-values (first column) as well as the *p*-values of the COMBI method (second column) and the *p*-values of the DeepCOMBI method (third column) are presented. Thresholds indicating statistical significance are represented by dashed horizontal lines and significant *p*-values are highlighted in green. Please note that the y-axes of all plots have the same limits (0 to 15) to enable direct comparison.

All SNPs reaching statistical significance in the permutation-based thresholding procedure of the DeepCOMBI method are presented in **Table 6**. As in **Table 3** for the findings of COMBI, we present basic information (associated disease, chromosome, identifier and χ^2 *p*-value) for all of the findings of DeepCOMBI. The fifth and sixth columns indicate whether they are found to be significant by RPVT with the application of $t^* = 1 \times 10^{-5}$ or by the COMBI method. Again, to validate all findings, the seventh and eighth columns report whether - and if so, in which external study - they have been found significantly associated with the given disease according to the GWAS catalog.

The DeepCOMBI method finds 39 significant associations. According to the fifth column of **Table 6**, 31 of these SNPs are also discovered by the traditional RPVT approach because they have *p*-values $< 1 \times 10^{-5}$. The other 8 of those 39 SNPs have *p*-values $> 1 \times 10^{-5}$ and are hence not determined to be associated with the disease with RPVT in the original WTCCC publication. They are of special interest because they represent additional SNP disease associations, which the traditional analysis of the data is not able to identify. Out of these eight novel discoveries, six have been validated independently in later GWAS or meta-analyses: rs7570682 on chromosome 2 and rs1375144 on chromosome 2 for bipolar disorder; rs6907487 on chromosome 6 for coronary artery disease; rs12037606 on chromosome 1 for Crohn's disease; rs231726 on chromosome 2 for type 1 diabetes and rs6718526 on chromosome 2 for type 2 diabetes.

Table 6: Significant SNPs of the DeepCOMBI method on seven WTCCC datasets and related association details from Mieth *et al.* (2020)[15]. For each SNP identifier on a specific chromosome that is found to be significantly associated with a disease by the DeepCOMBI method in Mieth *et al.*(2020)[15], the corresponding χ^2 test *p*-value is shown and it is indicated whether the RPVT *p*-value is $< 1 \times 10^{-5}$ (*i.e.* the SNP is a significant finding of RPVT as well), whether its COMBI *p*-value is smaller than the corresponding COMBI threshold (*i.e.* the SNP is a significant finding of the COMBI method as well) and whether the SNP has been found significant with a *p*-value $< 1 \times 10^{-5}$ in an external study with a corresponding PubMed identification number (PMID). Please note that the RPVT result in the fifth column corresponds to the χ^2 *p*-values calculated here, not necessarily to the original WTCCC publication, where they investigate Cochran-Armitage trend test *p*-values and presumably apply slightly different preprocessing steps. Similarly, the COMBI result in the sixth column corresponds to the re-calculations of COMBI we perform in the course of validating DeepCOMBI, not necessarily to those of the original COMBI publication, where slightly different results were produced due to the random nature of the permutation procedure.

Disease	Chromosome	Identifier	χ^2 *p*-value	Significant in RPVT	Significant in COMBI	*p*-value < 10^{-5} in at least one external GWAS or meta-analysis	References (PMID)
Bipolar disorder (BD)	2	rs7570682	1.77e-05		YES	YES	21254220
	2	rs1375144	1.26e-05		YES	YES	21254220
	3	rs514636	2.53e-06	YES	YES	YES	21254220
	16	rs420259	5.87e-08	YES	YES	YES	21254220
Coronary artery disease (CAD)	6	rs6907487	2.92e-05			YES	17634449
	9	rs1333049	1.12e-13	YES	YES	YES	17634449
	16	rs8055236	5.32e-06	YES	YES		
	22	rs688034	2.75e-06	YES	YES		

Disease	Chromosome	Identifier	χ^2 *p*-value	Significant in RPVT	Significant in COMBI	*p*-value < 10^{-5} in at least one external GWAS or meta-analysis	References (PMID)
Crohn's disease (CD)	1	rs11805303	6.35e-12	YES	YES	YES	17435756
	1	rs12037606	1.02e-05			YES	17554261
	2	rs10210302	4.52e-14	YES	YES	YES	23128233
	3	rs11718165	2.04e-08	YES	YES	YES	21102463
	5	rs6596075	3.11e-06	YES	YES		
	5	rs17234657	2.42e-12	YES	YES	YES	18587394
	5	rs11747270	1.05e-06	YES	YES	YES	18587394
	7	rs7807268	5.43e-06	YES		YES	26192919
	10	rs10883371	5.23e-08	YES	YES	YES	21102463
	10	rs10761659	1.69e-06	YES	YES	YES	22936669
	16	rs2076756	7.55e-15	YES	YES	YES	21102463
Hypertension (HT)	1	rs10889923	1.38e-05				
	15	rs2398162	6.01e-06	YES			
Rheumatoid arthritis (RA)	1	rs6679677	<1.0e-15	YES	YES	YES	20453842
	4	rs3816587	7.28e-06	YES	YES		
	6	rs9272346	7.38e-14	YES	YES		
	22	rs743777	1.01e-06	YES		YES	23143596
Type 1 diabetes (T1D)	1	rs6679677	<1.0e-15	YES	YES	YES	19430480
	2	rs231726	1.43e-05			YES	30659077
	4	rs17388568	3.07e-06	YES	YES	YES	21829393
	6	rs9272346	<1.0e-15	YES	YES	YES	18978792
	12	rs17696736	1.56e-14	YES	YES	YES	18978792
	12	rs11171739	8.36e-11	YES		YES	19430480
	13	rs4769283	1.20e-05				
	16	rs12924729	7.86e-08	YES	YES	YES	17554260
Type 2 diabetes (T2D)	2	rs6718526	1.00e-05		YES	YES	20418489
	4	rs1481279	9.44e-06	YES	YES	YES	28869590
	6	rs9465871	3.38e-07	YES	YES	YES	21490949
	10	rs4506565	5.01e-12	YES	YES	YES	23300278
	12	rs1495377	7.21e-06	YES	YES	YES	22885922
	16	rs7193144	4.15e-08	YES	YES	YES	22693455

Observe from **Table 6** that two out of the eight novel DeepCOMBI SNPs with *p*-values $> 1 \times 10^{-5}$ have not yet been replicated in an independent GWAS or meta-analyses. They have also not been identified by the COMBI method. Those entirely novel DeepCOMBI discoveries are rs10889923 on chromosome 1 for hypertension and rs4769283 on chromosome 13 for type 1 diabetes. To determine whether those two SNPs are biologically plausible discoveries for an association with the respective disease, their genomic regions are investigated in terms of functional indicators. Strong evidence of potential functional roles in the diseases is found.

Firstly, rs10889923 maps on an intron for *NEGR1* (neuronal growth receptor 1), a very important gene many times linked to obesity, body mass index, triglycerides, cholesterol and many other phenotypes highly correlated with hypertension[235–237]. Even though *NEGR1* has been associated with many phenotypes in the GWAS Catalog, no GWAS

has yet been able to link it to hypertension directly. Furthermore, rs10889923 is part of a high LD region (according to LDmatrix Tool[238]) with variants that have been reported to be significantly associated with a number of psychiatric disorders and phenotypes, *e.g.* educational attainment (in Lee *et al.*[45] rs12136092 with p-value $< 1e^{-11}$ and a degree of LD $R^2 = 0.86$ to rs10889923; rs11576565 with p-value $< 1e^{-8}$ and $R^2 = 0.63$). This link suggests a potential connection between hypertension and related phenotypes with mental traits. rs10889923 can thus altogether be considered an excellent candidate for association with hypertension.

Secondly, rs4769283 on chromosome 13 lies in an intergenic region very close to a gene called MIPEP (mitochondrial peptidase), which cannot be directly linked to T1D but is reported as a significant eQTL for two other genes, namely C1QTNF9B and PCOTH[239]. Thus, MIPEP and therefore rs4769283 significantly control expression levels of mRNAs from these two genes in a particular tissue. Most remarkably, rs4769283 is a significant eQTL (with p-value $= 1.1e^{-6}$) for C1QTNF9B (complement C1q and tumor necrosis factor-related protein 9B) in (amongst several other tissues) the pancreas, which produces very little or no insulin in T1D patients. So even though the association of rs4769283 with Type 1 diabetes is not an obvious one, it is indeed an interesting novel discovery of the DeepCOMBI method.

To present a more condensed view of these discoveries, **Table 7** summarizes the findings of the three competitor methods, RPVT, COMBI and DeepCOMBI.

Table 7: Quantitative summary of the significant findings of RPVT, COMBI and DeepCOMBI on seven WTCCC datasets from Mieth *et al.* (2020)[15]. For each of the three competitor methods, the numbers of replicated and unreplicated hits (*i.e.* the number of true and false positives) as well as precision and error rates are presented. Pairwise tests for the null hypothesis of identical distributions for DeepCOMBI and the two baseline methods are performed and corresponding p-values are given.

	Number of significant SNPs of		
	RPVT	**DeepCOMBI method**	**COMBI method**
SNPs that have achieved $p < 10^{-5}$ in at least one external study	33 (49% precision)	31 (79% precision)	31 (58% precision)
SNPs that have not achieved $p < 10^{-5}$ in an external study	35 (51% error rate)	8 (21% error rate)	22 (42% error rate)
Overall	68	39	53
Pairwise p-value (one-sided Fisher's exact test)	DeepCOMBI vs. RPVT = 0.00106		DeepCOMBI vs. COMBI = 0.01910

When no screening step is conducted and RPVT p-values are calculated for all SNPs, 68 locations with $p < 10^{-5}$ are identified as significant RPVT hits. COMBI and DeepCOMBI both apply a learning-based SNP preselection step and thus, find fewer significant associations. The DNN-based approach to this is seen to be more

conservative than the SVM-based one, with only 39 identified locations of DeepCOMBI in comparison to 53 findings of the COMBI method. Even though the DeepCOMBI method finds fewer significant SNPs than RPVT and COMBI, the number of independently replicated SNPs of DeepCOMBI (= 31 replicated SNPs, yielding a precision of 79%) is identical to that of COMBI (31, precision = 58%) and almost identical to that of RPVT (33, precision = 49%). In addition, the DeepCOMBI method misclassifies only eight unreplicated SNPs as associated with the disease (yielding an error rate of only 21%), while RPVT wrongly classifies 35 SNPs (error rate = 51%) and the COMBI method makes 22 mistakes (error rate = 42%). These observations are quantified with pairwise one-sided Fisher's exact tests for the null hypothesis of equal error rates for both methods. They produce significant *p*-values for both comparisons: DeepCOMBI vs. RPVT (*p*-value of 0.00106) and DeepCOMBI vs. COMBI (*p*-value = 0.01910).

In **Figure 24**, we present the ROC and PR curves of the three competitor methods, where we interpret the replication of SNPs according to the GWAS catalog as a validation, *i.e.* we count a SNP as a true positive if it has achieved $p < 10^{-5}$ in at least one external study. Overall, the findings obtained by the DeepCOMBI method are better replicated than those obtained by RPVT and COMBI for all levels of error. The performance metrics of the DeepCOMBI method (pink line) are consistently better than that of RPVT (light blue lines) and COMBI (dark blue lines). The DeepCOMBI method finds more true positives for different levels of error and yields higher levels of precision for different levels of recall than COMBI and RPVT.

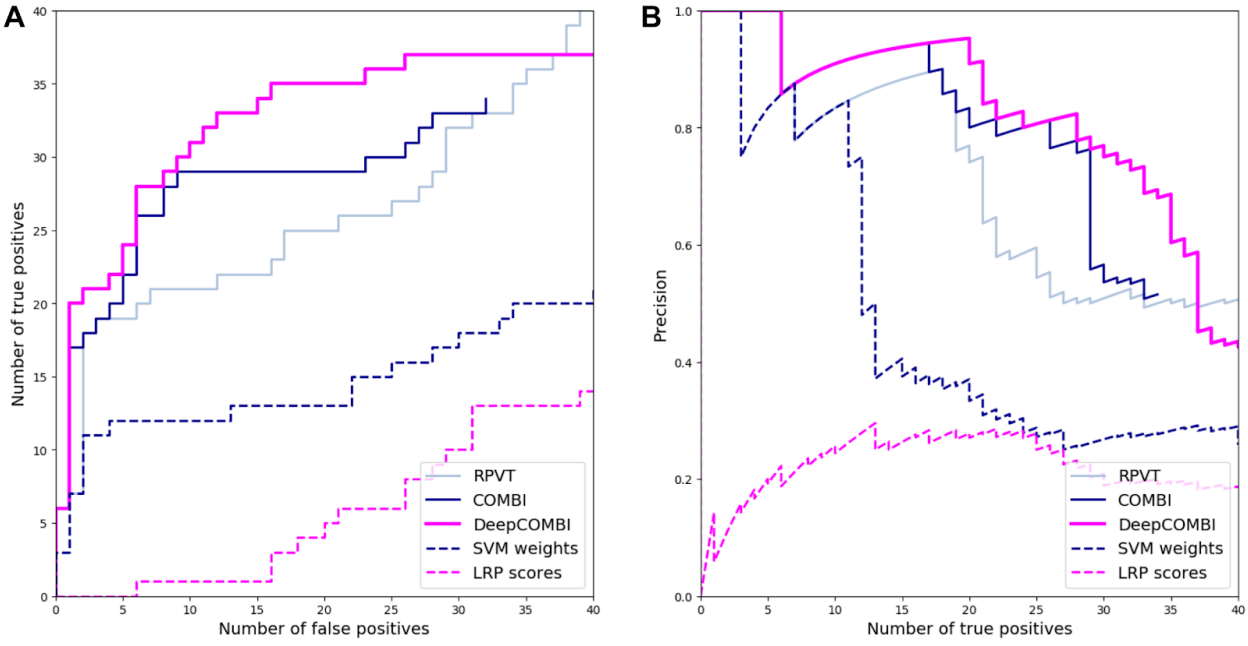

Figure 24: Performance curves of DeepCOMBI and competitor methods on seven WTCCC datasets from Mieth *et al.* (2020)[15]. **A** ROC curves and **B** PR curves of RPVT, the COMBI method, the DeepCOMBI method, direct thresholding of SVM weights and direct thresholding of LRP scores averaged over all diseases and chromosomes are shown. Replicability according to the GWAS catalog is used for validation.

Figure 24 also shows the performance curves of the other two baseline methods that threshold SNPs solely based on raw LRP relevance scores or raw SVM weights, respectively. As we can view these two methods and RPVT as the individual components of the combinatorial approaches and neither of these three can achieve the same level of performance as COMBI and DeepCOMBI, it can be deduced that all components are essential. Only the combination of the two components of the COMBI method (SVM and statistical testing) and the DeepCOMBI method (DNN with LRP and statistical testing) can achieve the desired performance increase.

Comparison to other baseline methods on WTCCC datasets

In a functional study, we now compare COMBI against two other state-of-the-art methods proposed by Lippert *et al.*[188,192]. They devise a novel univariate analysis method to improve WTCCC findings and also implement an LMM to uncover new epistatic associations by means of brute force comparison of pairwise interactions. They apply both methods to the seven WTCCC datasets, searching for new univariate signals and for epistatic associations. For the univariate analysis, they report a total of 573 novel SNP-disease associations[188] with *p*-values less than 5×10^{-7} distributed over all WTCCC diseases except for CAD, for which no novelty is reported. A lot of these 573 SNPs are part of small SNP clusterings, so we select representative markers for each locus through the LD pruning option in PLINK and compute pairwise LD with a sliding window of two SNPs (with steps of 1 SNP at a time). We then discard one SNP out of each pair if they are in high LD ($R^2 \geq 0.8$). We run the final list of SNPs, consisting of 1 discovery for BD, 0 for CAD, 19 for CD, 1 for HT, 3 for RA, 39 for T1D and 9 for T2D, through our validation pipeline (described in **Chapter 3.2.3**, **Figure 12**) using the same parameters that we use for COMBI in this dissertation (physical distance to tag-SNP: < 200 kb. LD with tag-SNP: $R^2 \geq 0.8$). The results are presented in **Table 8**.

Table 8: Quantitative comparison of the significant findings of the COMBI method from Mieth *et al.* (2016)[10] and the univariate method from Lippert *et al.*[188]. The significant SNPs of the COMBI method from **Table 3** are compared with the significant findings of the univariate analysis presented by Lippert *et al.* BD, CAD, HT and T1D are summarized because no discoveries of Lippert *et al.* for these diseases are validated in independent studies.

Disease	COMBI		Lippert *et al.* univariate analysis	
	Discoveries	Validated discoveries	Discoveries	Validated discoveries
CD	11	8	19	3
RA	3	1	3	1
T2D	8	3	9	1
BD, CAD, HT, T1D	24	16	41	0
Overall	46	28	72	5
	p-value of one-sided Fisher's exact test: < 0.00001			

In Table 8, the number of true-positive SNPs from Lippert *et al.* (that is, of discoveries that have been validated in the literature) is very small, with only five of the reported SNPs featured in GWAS published after the WTCCC study (CD: 3 true positives, RA: 1 and T2D 1 true positive each). This figure is much smaller than for COMBI, which reports 12 validated SNPs for these diseases. Not only does COMBI give rise to more validated discoveries, but these discoveries cover the whole range of WTCCC diseases. Overall, the advantage of COMBI over the univariate analysis of Lippert *et al.* is significant with a p-value < 0.00001.

The epistasis analysis by Lippert *et al.*[188] consists of a brute force computation of all possible pairwise SNP associations for the seven diseases (~63 billion pairs; no hits reported for CD) and testing their epistatic interaction in disease risk for significance. The authors report a final list consisting of 707 pairs of SNPs with p-values lower than 7.9×10^{-13}. Applying the LD pruning method as described above, Lippert *et al.* identify two pairs of SNPs for BD, 32 for CAD, 0 for CD, 2 for HT, 7 for RA, 13 for T1D and 2 for T2D. We evaluate all individual SNPs taking part in the significant reported interactions via our validation pipeline. We are aware that this is only a suboptimal validation since epistasis is not the sum of separated SNP effects, which are those that are registered in the GWAS catalog, but some associations could still emerge. By running the corresponding markers through our validation pipeline, we find a single association for T1D, while no markers are found for the other diseases. A comparison against the validated discoveries of COMBI is presented in **Table 9**. The superiority of the COMBI method over the epistatic analysis of Lippert *et al.* in this context is significant with a p-value < 0.00001 .

Table 9: Quantitative comparison of the significant findings of the COMBI method from Mieth *et al.* (2016)[10] and the epistatic analysis method from Lippert *et al.*[188]. The significant SNPs of the COMBI method from **Table 3** are compared with the significant findings of the epistatic analysis presented by Lippert *et al.* BD, CAD, CD, HT, RA and T2D are summarized in the table because no discoveries of Lippert *et al.* for these diseases are validated in independent studies.

Disease	COMBI		Lippert *et al.* univariate analysis	
	Discoveries	Validated discoveries	Discoveries	Validated discoveries
T1D	9	6	13	1
BD, CAD, CD, HT, RA, T2D	37	22	45	0
Overall	46	28	58	1
	p-value of one-sided Fisher's exact test: < 0.00001			

3.4 Summary and discussion

Numerous different approaches for the analysis of GWAS have been introduced since the first of its kind was published in 2002. Traditionally, they focus either on accurate phenotype prediction[57–59] or the identification of SNP-phenotype associations[63,65,66]. At first, most of these approaches were of a purely statistical nature[42,43], but since ML has become increasingly important in data science, it also found its way to the investigation of genomic data. Like traditional methods, these ML-based approaches can be classified into two groups:

1. methods that construct an ML model from genetic data in order to carry out accurate predictions of a phenotype[60,61,193,196,197,240–242]; and
2. methods that use ML to construct a statistical association test or rank genetic markers according to their predicted association with a phenotype[62,63,66,213,243–246].

The set of papers that fall into the first category, for example, study the predictive performance of non-penalized or penalized, linear or nonlinear regression and classification models, including SVMs[11–13], random forests[247] and sparsity-inducing methods such as the elastic net[154], on various complex diseases (including the ones studied here), showing that ML methods such as SVMs – if appropriately applied - can perform well at predicting disease risks.

This dissertation aimed at both of the goals presented above but put focus on the second category by using ML-based prediction methods in combination with statistical testing to identify SNPs associated with the phenotype under investigation. We first proposed a novel method - called COMBI - which utilizes the advantages of an SVM trained for phenotype prediction to select a set of candidate SNPs for multiple hypothesis testing. It was shown that COMBI outperformed the traditional RPVT statistical testing approach on generated as well as real GWAS datasets. In addition, it was found that the individual components of the proposed combinatorial approach - SVM training, statistical testing and the moving average filter - cannot achieve comparable performance when their raw scores are used as test statistics and directly thresholded for significance. Only the combination of these components can improve the identification of associated SNPs successfully.

The COMBI method was furthermore compared to alternative methods that stem from the second category from above, some of which include two-stage approaches, first performing statistical testing and then ML to refine the set of predicted associations[62,63]. These approaches, however, are unable to take correlation structures of SNPs that have been excluded in the first step into account and neither method was validated on real data in terms of a comparison to the GWAS database. Similarly, Pahikkala *et al.*[248] and He and Lin[249] developed methods for ranking genetic markers based on the sure independence screening strategy[250] and stability selection analyzing only one SNP at a

time. The approach was extended to detect gene-to-gene interactions by Li *et al.*[250,251], but neither of the methods was validated on independent external studies. Another approach was introduced by Alexander and Lange[231], who applied the stability selection method of Meinshausen and Bühlmann[64] to the WTCCC dataset to rank SNPs according to their predicted association with a phenotype. We found that stability selection effectively controls the *FWER* when applied to GWAS data but suffers a loss of power while at the same time rendering conservative results. The work that is probably most closely related to the present research is the two-step algorithm by Wasserman and Roeder[210](and the extension by Meinshausen *et al.*[211]), who split the data into two equal parts performing marker selection on the first part and then testing the selected markers on the second part.

In order to investigate and compare the performance of the COMBI method to other ML approaches, the work of Wasserman and Roeder[210], Meinshausen *et al.*[211] and Roshan *et al.*[62] were selected as representative baseline methods. In **Chapter 3.3.1**, it was shown that the COMBI approach outperformed all of these methods on semi-real generated data. Wasserman and Roeder[210] and also the extension by Meinshausen *et al.*[211] lose a great amount of statistical power by splitting the GWAS dataset under investigation into two parts, performing SNP preselection on one part and statistical testing on the other. This approach is significantly less successful in identifying SNP disease associations than COMBI, which performs all substeps on the complete (and therefore statistically more powerful) dataset. Another important and very closely related method by Lippert *et al.*[188,192] aims to identify putative significant disease-marker associations using two approaches based on LMMs: a univariate test and a test for pairwise epistatic interactions. LMMs, like COMBI, address the issue of population stratification in GWAS, cf. Mimno *et al.*[252]. However, in contrast to COMBI, they still test SNPs (or pairs of SNPs) individually, one after the other and thus potentially lose detection power. Another possible shortcoming of LMMs and related methods over SVMs is that they are tailored for regression and not binary classification. Recently the approach of Lippert *et al.* has been extended for disease risk prediction (Rakitsch *et al.*[253]) and related approaches have been proposed by Loh *et al.*[254] and Song *et al.*[255], suffering the same drawbacks as discussed above. For a comparison of COMBI with Lippert *et al.*[188,192] on real WTCCC data, see **Chapter 3.3.2.** When the results of the univariate method of Lippert *et al.* were checked against the same validation criteria we used for COMBI, it turned out that our method reported 17 more true positives (4.4 times more positives) for the three diseases for which their univariate method reported at least one hit and performed significantly better on all datasets. The COMBI method also holds great potential for testing pairwise SNP-trait associations, as it drastically reduces the number of candidate associations by selecting a subset of the most predictive SNPs in the ML step. Again, a comparison to the method Lippert *et al.*[188,192] proposed for detecting epistatic interactions was also significantly favorable to COMBI. An extension of LMMs to multivariate cases was developed by Zhou and Stephens[256] but

has not yet been applied to WTCCC. Fitting LMMs to multiple phenotypes provides no novel insight into analyzing multiple genotypes/SNPs at once, which is the issue COMBI addresses.

With the increasingly large amounts of available data, deep learning-based approaches and artificial NNs are now also being applied to GWAS datasets[198,199]. However, most of these publications focus on pure classification or regression prediction tasks[193,200–202], rather than the identification of associated SNPs in the corresponding datasets[193,203]. To harness the great potential of DNNs for the analysis of GWAS, we proposed another novel method - called DeepCOMBI - which uses a deep learning-based phenotype prediction in combination with statistical testing for identifying SNPs that are associated with the phenotype under investigation. DeepCOMBI could be considered an extension of COMBI, replacing the rather simple linear SVM with a more sophisticated method and using the concept of explainability to reveal the underlying decision-making process. In particular, DeepCOMBI trains a DNN and extracts SNP relevance scores via LRP[16–18]. To our knowledge, Romagnoni *et al.*[193] were the first and only scientists (up until the publication of DeepCOMBI) to use XAI in the context of GWAS and propose to apply PFI. Even though they were able to identify some novel predictors, the prediction performance of their NN was not better than that of traditional ML-based tools. In addition, PFI is a generalized, model-agnostic approach and with the DeepCOMBI method, we utilized more advanced deep Taylor-based explanation techniques by adopting LRP for the analysis of GWAS data.

DeepCOMBI was shown to compare favorably to its main competitor COMBI on both generated controlled datasets as well as seven real-world GWAS datasets. These findings are in accordance with Romagnoni *et al.*[193], who found that deep learning-based methods can provide novel insights into the genetic architecture of specific traits. By applying LRP, we were able to leverage the power of DNNs and generate relevance scores that are less noise inflicted than the SVM importance scores of COMBI. In return, the preselection of candidates SNPs is better than that of COMBI and higher TPR and precision can be achieved for all levels of error. In addition to the main competitor method, COMBI, we also compared DeepCOMBI to the baseline methods of RPVT, raw LRP relevance scores and raw SVM importance scores and showed that only the combination of deep learning and multiple testing show the desired performance increase, which cannot be achieved individually by one of these components. Since the COMBI method itself was shown before to outperform other combinatorial ML-based approaches (Roshan *et al.*[62], Meinshausen *et al.*[213] and Wasserman and Roeder[210]) and two purely statistical analysis tools (Lippert *et al.*[188,192]), it can be directly deduced that DeepCOMBI also outperforms those approaches.

To summarize, we proposed two novel and powerful methods for analyzing GWAS data that are based on applying a carefully designed ML step before applying a classical multiple testing step. Certain ML models, in particular appropriately designed linear

SVMs and DNNs, take high-dimensional correlation structures into account and thus implicitly incorporate interactions between different loci. A subset of predictive candidate SNPs is extracted within the ML step. The COMBI method implements the ML step with an SVM and interprets the learned weights as importance scores in order to select candidate SNPs for multiple statistical testing. The *p*-values corresponding to association tests are then thresholded with a permutation-based procedure for these candidate SNPs in a subsequent statistical testing step. COMBI was shown to outperform the RPVT approach both on controlled, semi-real data and on data from the WTCCC 2007 study, for which reported associations were validated by their replicability in later external studies. The empirical analysis showed a significant increase in detection power for replicated SNPs while yielding fewer unconfirmed discoveries. Two new (as yet unreplicated) candidate associations were reported. The second method we proposed is called DeepCOMBI and implements the ML step as follows: After training a carefully designed DNN to classify subjects into their respective phenotype, the concept of XAI is applied by backpropagating the class prediction score to the input layer through the network via LRP. The resulting SNP relevance scores are used to select the most relevant SNPs for multiple testing in combination with the same permutation-based thresholding procedure of the COMBI method. On both generated, controlled datasets as well as seven real GWAS datasets, DeepCOMBI was shown to outperform COMBI and a number of other competitor methods in terms of classification accuracy of the DNN and in terms of ROC and PR curves when using either the generated labels or replicability in external studies as a validation criterion. In addition, two very promising, entirely novel SNP disease associations were discovered. Located on an intron for *NEGR1*, an important gene many times linked to obesity, body mass index and other correlated factors, rs10889923 on chromosome 1 was found to be significantly linked to hypertension. Another novel location found by DeepCOMBI to be associated with type 1 diabetes is rs4769283. It is part of an intergenic region on chromosome 13 and was previously found to be an eQTL for C1QTNF9B in the pancreas, the affected organ in T1D patients.

In reference to the author's thesis of this dissertation, in this chapter, two novel methods for the analysis of GWAS were proposed that are both based on a combination of the traditionally used analysis tool - multiple hypothesis testing - and a novel ML-based technique - SVM and DNN, respectively. Validating the proposed thesis, they were successfully applied to GWAS datasets and contributed to a better understanding of the translation of genetic code into phenotypes. The proposed methods were shown to rely on each of its individual components and increase the statistical power and accuracy of existing techniques. In this framework, too - "*the whole is greater than the sum of its parts*"[26] (derived from Aristotle, 4th century BC).

A number of alternatives and possible options for future research exist. The proposed approach can be extended to explore different directions by substituting one of the two steps of the general algorithm (first ML, second statistical testing) with other suitable

procedures. The effects of replacing the χ^2 test in the final step of the two proposed methods with a different, more sophisticated kind of test should be studied. For example, procedures correcting for population structures or other confounding factors[192,257] or investigating pairwise hypotheses or other multivariate effects could be examined. Another goal could be to extend the COMBI and DeepCOMBI methods to a regression setup, where the phenotype is not binary. DNNs and SVMs can easily be adjusted to non-binary phenotypes. Considering the first step of the method, one could apply other ML prediction methods instead of training an SVM. For example, the SVM training could be replaced by SNP selection via random forests or component-wise boosting. Future work on the subject of deep learning and XAI in the context of analyzing GWAS datasets could also focus on replacing the first step of a DNN in the DeepCOMBI method. DNNs with different architectures or other suitable analysis tools could be investigated. For example, future research could aim to harness the potential of CNNs[258] as a promising candidate network architecture since CNNs implement similar feature extracting properties of short-range local effects as the moving average filter of the proposed methods. By integrating multiple output nodes for multiple phenotypes, the DNN could also be extended to cover multivariate output variables and examine multimorbidities. Improvement ideas for the second step of the DeepCOMBI method (explanation) include the application of different explanation methods or LRP backpropagation rules, for example, according to the layer types, as advised by Montavan *et al.*[17]. Great potential lies in finding more sophisticated ways to combine the local LRP explanations of each individual subject to a single global explanation used for SNP selection. A very promising candidate in this regard would be a method called SpRAy[259], which clusters the individual explanations and simplifies the identification of explanatory structures in subsets of subjects.

Considering the weaknesses of the proposed methods, it is important to note that the most crucial limitation in ML is the amount and quality of the training data. DNNs, as utilized in the DeepCOMBI method, especially rely on large datasets. However, in GWAS, the number of features (*i.e.* SNPs) is often much bigger than the number of datapoints (*i.e.* subjects). Trying to recruit subjects for studies is often not only a cost-based problem but can also be difficult when rare phenotypes are studied. On the other hand, more and more genomic sequences can be accessed on public web platforms (such as *openSNP*) where people who have paid commercial genetic testing companies (such as *23andMe*) to sequence their genome and have then decided to share their results for various reasons[260]. In the future, these open-source datasets might present a huge potential for biological and medical research. However, the quality and consistency of such data will be far from the conventional standards of GWAS, which limits the applicability, robustness and stability of ML methods. For example, neither of the methods proposed here would be able to handle datasets where the genetic sequences contain the information of different SNPs. They could only analyze these datasets when limiting their feature space to the SNPs that are available for all

sequences. Possible solutions to this problem could be learning joint and retractable representations of all datapoints in a shared feature space[261–263] or applying the concepts of transfer and multitask learning to allow for datapoints to lie in different spaces[264].

As an additional limitation of the proposed methods, it is essential to note that identifying associated SNPs in GWAS can only indicate association, not causation. The findings of GWAS always need to be biologically verified. Building trust in the corresponding methodologies is especially important in the sensitive field of medical applications. Although the proposed methods utilize explanation techniques, they remain black boxes in the sense that it is unclear *how* the associated SNPs affect, for example, the risk of developing a disease. Our validation pipeline enquiring the GWAS catalog can be a great approach in this respect and we further address the issue of meaningful associations in **Chapter 4**. Additional interpretability and explanation techniques should be adopted in the future to verify the results of GWAS. However, the judgment of experts will always remain crucial and machines can only guide the processes in medical decision-making.

Furthermore, another important limitation of the proposed methods is that they are stochastic and non-deterministic at multiple stages of the algorithms causing reproducibility to be a potential problem. The gradient-based SVM and DNN training and the permutation test procedure both introduce a certain level of randomness. We have shown in **Appendix II** that when applying the methods repeatedly and to varying subsets of the WTCCC data, the results of the COMBI method are more similar and thus more stable than those of RPVT. However, even though high-performing fast-converging optimizers exist, they are never guaranteed to find the optimal solutions during training. In addition, the quality of a given classifier always depends on their architecture, the hyperparameters and the initial weights which are selected by the user. DNNs, especially, are known to be highly sensitive to such *a priori* choices[265]. In this context, it is also important to note that finding a global optimum is often time-consuming[266] and performing the proposed permutation test procedure increases the computation times even further. Dense DNNs scale poorly with the number of datapoints and features studied. However, as mentioned above, GWAS tend to include much fewer subjects than SNPs and we have shown that DeepCOMBI performs well in combination with a *p*-value based SNP preselection step. To avoid an explosion of computing times and required resources, more direct approaches to thresholding could be developed. However, directly thresholding the learned weights of the ML algorithms was shown here not to perform as well as the proposed p-value thresholding.

Another issue to address in the context of the methods' randomness is a common phenomenon in scientific research where people tend to re-analyze a dataset until something "*publishable*" is found[267]. For the proposed methods, the permutation test procedure is aiming to guarantee that data dredging and *p*-hacking is not possible, but of

course, these methods, too, can be applied repeatedly until they produce the desired results.

Whenever training AI algorithms on human-related data, it is crucial to mention the ethical aspect of such studies[268]. Machines do not intrinsically have morals and cannot unsolicitedly distinguish between right or wrong, especially when they are trained on unbalanced data. It is always the researcher's responsibility to ensure a fair and just representation of all humans in their data or communicate if that is not the case. When GWAS are conducted in the social-behavioral field, for example, there is a risk of supporting existing discrimination (*e.g.* on the basis of race, gender or origin) because of unbalanced training datasets and hence failing to appropriately study the targeted phenotypes (*e.g.* personality traits). GWAS datasets need to be carefully created and account for various factors via stratification. There exist numerous techniques to do so[269,270] and both the COMBI and the DeepCOMBI method would benefit greatly from an adaption of such concepts.

To sum up the limitations mentioned above, as always in AI, the performances of the proposed methods depend on the ability and willingness of future users to train a good model and perform sincere research. Additionally, to account for the challenges they face when applied in various biological and medical studies, the presented methods should be developed further, adapted and improved according to the current state of AI research.

4 Combining transfer learning with clustering methods for single-cell RNA sequencing studies

In this chapter, we propose a novel method that combines traditional clustering approaches (from **Chapter 2.2.2**) with an innovative transfer learning technique (from **Chapter 2.3.5** and **2.3.6**) to cluster individual cells according to their transcriptomic output in scRNA-Seq datasets (described in **Chapter 2.1.3**). Given a well-known source dataset with clustering labels, the proposed method improves the clustering of an unlabeled target dataset by transferring knowledge from source to target data via NMF[23–25]. The modified target dataset can then be provided to any kind of clustering algorithm. In support of the author's thesis in this dissertation, the proposed combinatorial approach helps to better understand gene expression in different cell types and therefore examines the translation of genetic code into biological function.
This chapter is based on and contains parts of article **B**[22] from **Chapter 1.4** on previously published work.

4.0 Notation of chapter 4

Symbol	Definition (page it is introduced on)
α_{NMF}	Penalty multiplier of the elastic net in NMF (41 and 115)
g	Number of dimensions, *i.e.* genes, in a scRNA-Seq dataset (115)
$H,\ H_{src}$	Dictionary in NMF (40 and 115)
i	Index of a cell in a scRNA-Seq dataset (116)
k	Number of clusters to find in a scRNA-Seq dataset (115)
λ_{NMF}	Parameter of the elastic net in NMF controlling L1 and L2 regularization (41 and 115)
n_{src}	Number of points, *i.e.* cells, in a source scRNA-Seq dataset (115)
n_{trg}	Number of points, *i.e.* cells, in a target scRNA-Seq dataset (115)
θ	Mixture parameter of the transfer learning method (115)
$vec(.)$	Vectorization of a given matrix (115)
W, W_{src}, W_{trg}	Reconstruction or clustering data matrix in NMF (40 and 115)
W'_{trg}	Simplified version of the target reconstruction matrix W_{trg} (116)
W^*_{src}	Initial starting point of W_{src} in NMF (116)
x_{cells}	Cutoff value for percentage of cells for gene filter (122)
$x_{expression}$	Cutoff value for expression level for cell and gene filter (122)
x_{genes}	Cutoff value for number of genes for cell filter (122)
X_{src}	Source dataset for transfer learning (115)
X_{trg}	Target dataset for transfer learning (115)
X^{new}_{trg}	Newly constructed target dataset (116)
y^{src}	Known cluster memberships of the cells in the source dataset (115)
$\hat{y}^{src}$	Predicted cluster memberships of the cells in the source dataset (117)
$\hat{y}^{trg}$	Predicted cluster membership of the cells in the target dataset (116)
$\|\| \cdot \|\|_1$	L1 Manhattan Norm (115)
$\|\| \cdot \|\|_{Fro}$	Frobenius norm (115)

4.1 Introduction

Sorting objects into groups with limited or no *a priori* knowledge is a common problem in many different areas of scientific research[271,272]. In biological and medical sciences, datasets are often constrained by the scarcity, feasibility and expense of collecting samples. As such, it is not straightforward to apply state-of-the-art methodologies, like deep learning, which requires large and well-annotated datasets to solve many problems sufficiently well. To address this, the concept of using transfer learning (as described in **Chapter 2.3.6**) to integrate *a priori* knowledge from reference datasets into target datasets has been proposed as one way to generate additional insights[173,174]. One of the scientific fields where these problems are of interest is scRNA-Seq, as described in **Chapter 2.1.3**. **Figure 25** shows a graphical representation of the scRNA-Seq procedure and the application of transfer learning to its specific problem setting.

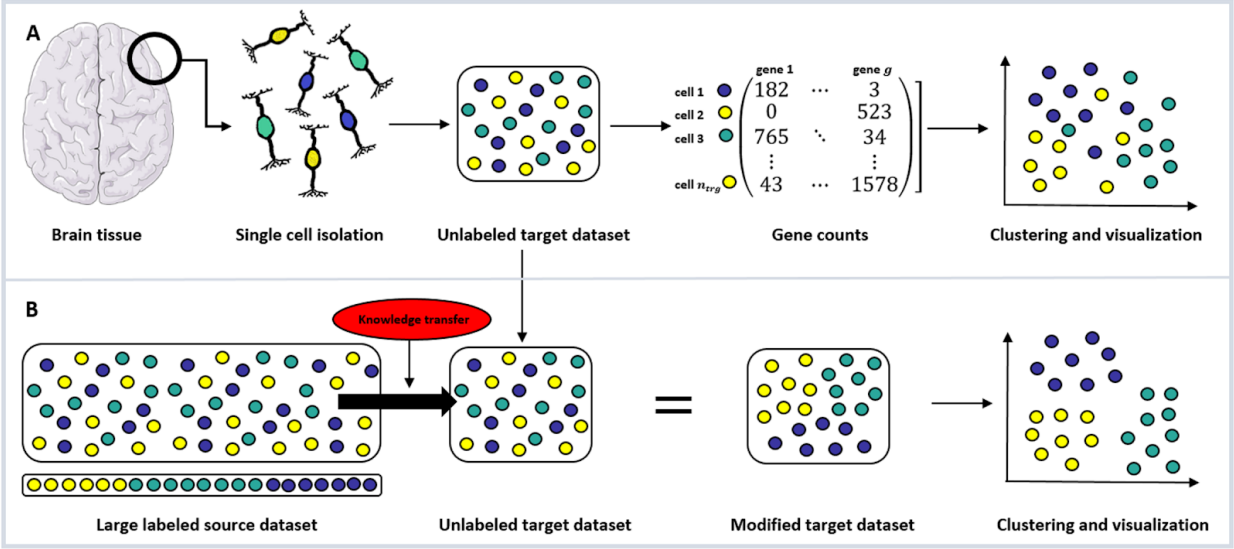

Figure 25: scRNA-Seq and transfer learning. A Recent scientific and biotechnological developments have enabled scRNA-Seq, the accurate measurement of the transcriptional output of individual cells. Once a tissue sample (*e.g.* brain tissue) is extracted from an organism, single cells (*e.g.* neurons) are isolated and sequenced. For each gene, the number of times a corresponding transcript is found in each individual cell is counted. These gene expression profiles of single cells are then used to identify tissue-specific cell types or states through an unsupervised clustering algorithm (*e.g.* SC3), which can eventually be visualized (through *e.g.* t-SNE or PCA plots). **B** When clustering smaller disease or tissue-specific scRNA-Seq datasets, it is often desirable to utilize large labeled reference datasets. The current work proposes to apply the ML concept of transfer learning to modify the unlabeled target dataset via NMF in a way that reflects specific properties of a large labeled source dataset and improves the results of downstream clustering algorithms (in our case, SC3). Please note that even though this graph shows a complete overlap in cell types, both source and target datasets might include cell types that do not appear in the other set.

In recent years, a series of advances in molecular biology[19,273,274], microfluidics[275,276] and data analysis[78] have led to our ability to accurately measure the transcriptional output of large numbers of individual cells through scRNA-Seq (**Figure 25 A**). The application of this technology has already led to insights into cellular development[19,277], dynamics[278] and heterogeneity[279,280] and the pathogenesis of human disease[281]. The advent of major global initiatives focusing on scRNA-Seq such as the Human Cell Atlas[282] means that the importance and impact of this technology are likely to grow, as are the associated data analysis challenges.

Most scRNA-Seq experiments are concerned with the identification and cataloging of cell types or states within a tissue or biofluid[283,284] (**Figure 25 A**). Historically this has been done through measurement, often qualitatively, of small numbers of "*marker*" genes whose expression has been observed to correlate with cellular function. scRNA-Seq complements these approaches by being high-throughput, quantitative and cost-effective in generating high-dimensional data suitable for cell-type classification. Neuronal cell types, for instance, have been deeply studied by scRNA-Seq[285], leading to new, unbiased, data-driven classifications of neurons and other cell types within the mammalian peripheral and central nervous systems[82,286–291]. Unique disease-associated cell states such as microglial subtypes associated with Alzheimer's Disease[281] have also been identified by scRNA-Seq.

In **Chapter 2.2.2**, we describe the traditional challenges and methods for the analysis of scRNA-Seq data. Specifically, we present unsupervised clustering of cells into groups according to their transcriptional state as the fundamental analysis in scRNA-Seq experiments and name a number of approaches to address this problem via hierarchical and iterative clustering[80,81], PCA based approaches[82,83], ensemble clustering[81,84], graph-based approaches[85–89], ML-based[90,91] and deep learning-based[92–95] techniques.

Challenges remain in the field, especially when the number of cells profiled in a given experiment is relatively small and as such rare cell subtypes are poorly represented[86]. Our hypothesis is that large reference scRNA-Seq datasets are a hitherto untapped resource for the clustering of other datasets that may be smaller in size but examine a specific tissue or disease context. Here, we propose that the concept of transfer learning (*i.e.* the ML technique of applying knowledge gained from one context to another distinct but related context) can be effectively implemented to improve clustering of scRNA-Seq data when a suitable reference dataset is available (**Figure 25 B**). Transfer learning covers multiple problems[173], *e.g.* multitask learning, domain adaptation and covariate shift[177]. Specifically, it refers to a setting where the solution of one or multiple source tasks is applied to a related target task. In multitask learning[136], multiple related tasks are learned in a parallel fashion using a shared representation instead of learning a sequence of related tasks. In the analysis of scRNA-Seq data, this translates to a situation where we are interested in simultaneously clustering a number of different datasets stemming from different studies, laboratories or points in time. These kinds of

datasets most likely contain batch effects, which need to be corrected for when combining the datasets for meta-analyses. In scRNA-Seq analysis, clustering and batch effect removal are typically addressed through separate steps, *i.e.* only after removing batch effects and combining multiple datasets into one is clustering analysis performed. These kinds of batch effect correction approaches can be graph-based[136,292–296], dimensionality-reduction based and variance-driven[297–299] or incorporate deep-learning procedures[300,301]. Different approaches to grouping cells of multiple datasets by cell type rather than dataset-specific conditions put emphasis on performing batch effect removal jointly with the clustering analysis[302–304]. More general approaches compare subtypes of cells across different samples[305] and identify clusters with high similarity across datasets[306,307]. All of the aforementioned methods presume that the multiple datasets under investigation are related in some way and are subsequently clustered simultaneously. In this dissertation, we focus on a more specific problem setting, where the user is interested mostly in the clustering of a target dataset, making use of the knowledge from a well-known and well-understood source dataset. A number of tools are available for annotating cells of a target dataset to a predefined reference set of cell types[308–311], but they are limited to target datasets that only include cells of the same types in source data and hence, cannot identify new cell types.

To enable knowledge transfer without having to combine the two datasets and at the same time guarantee a target clustering to be independent of the cell types of the source data, this dissertation focuses on the specific concept of transfer learning to use information from one scRNA-Seq dataset to annotate another without limiting the cell types that may be found in either. The aim is to adjust the target dataset with information from the source data and feed this new target dataset into a downstream clustering algorithm. In this specific setting, the method that is the most closely related to our work is SAVER-X[312]. SAVER-X trains a deep autoencoder on a target set with an initialization of the weights obtained from training on the source target dataset coupled to a Bayesian model to leverage existing data in the denoising of a new scRNA-Seq dataset. SAVER-X is a deep learning-based approach and is thus limited to datasets of very large sample size. This dissertation focuses on improving the clustering of small datasets and does not require large sample sizes. Unlike deep learning-based approaches, our method is convex and always returns the globally optimal solution independent of its initialization. Additionally, instead of focusing on denoising target datasets like SAVER-X, we are trying to insert additional knowledge (*i.e.* to induce certain specific properties of the source dataset that the researcher wants to put special emphasis on) into the target dataset. This is achieved by making use of specific source datasets and, in particular, by including cell type annotations from the source into the analysis. Large reference datasets are often very well studied and come with high-quality annotation of the cell types present within them. Our algorithm is not attempting to re-cluster this already well-clustered data, but it is making use of those pre-existing source labels (**Figure 25 B**).

Another relevant work focussing on transformations of scRNA-seq data for improved cell type clustering[313] is also deep learning-based and consists of three subsequent steps. Firstly, a supervised NN is trained to predict the cell types of a given source dataset. Secondly, the target dataset of cell types not used in training is plugged into the network and the values of the hidden layer are used as a new representation of the target dataset. Lastly, the newly constructed target dataset is clustered with unsupervised *k*-means clustering enabling cell types in source and target data not to be identical. Please note that the focus of the present work lies on transferring knowledge between source and target datasets that have a significant overlap in their cell types. The method proposed by Lin *et al.*[313] is explicitly restricted to non-overlapping settings.

To summarize, the current approach is not directly comparable to the methods presented here because it tackles a very specific problem that - to the best of our knowledge - no other method has addressed. Implementations of the method are available as a Python framework at https://github.com/nicococo/scRNA.

4.2 Methods

We propose a method to apply transfer learning (as described in **Chapter 2.3.6**) to scRNA-Seq data that enables us to transfer knowledge from a relatively well-annotated and large source dataset to a smaller unannotated target dataset. A graphical representation of the method can be found in **Figure 25 B**. The method is based on a transfer learning step that modifies the target dataset to incorporate knowledge gained from the well-annotated source dataset. The newly constructed target dataset can then be analyzed with a clustering algorithm to obtain an improved clustering compared to applying that same method to the target without any transfer learning procedure or a simple concatenation of source and target.

The following sections describe the method in more detail and specify the experimental setup of performance assessments on generated synthetic data, controlled real data and a real-world application of two independent datasets.

4.2.1 Problem setting

There exists a well-known source dataset $X_{src} \in \Re^{g \times n_{src}}$ with scRNA-Seq data from n_{src} cells and g genes for which we have in-depth knowledge about the clustering structure (*i.e.* ground-truth labels $y^{src} \in \Re^{n_{src}}$) and a target dataset $X_{trg} \in \Re^{g \times n_{trg}}$ of n_{trg} cells and g genes, which we want to enhance given the information in X_{src} and y^{src} before clustering into k groups of cells, *i.e.* predicting $\widehat{y}^{trg}$.

4.2.2 Proposed workflow

The basic underlying idea of the proposed method is to factorize the source dataset into a data size-independent part (of size $g \times k$) and a gene-independent part (of size $k \times n_{src}$) and to use the former – which is often called a *dictionary* since it does not depend on the number of cells n_{src} and can thus be used to *translate* between datasets – to modify the target dataset accordingly.

More specifically, the novel approach, based on NMF (as described in **Chapter 2.3.5**), can be described in the following steps:

1. We use NMF[23–25] of our source data $X_{src} \in \Re^{g \times n_{src}}$ to learn a dictionary $H_{src} \in \Re^{g \times k}$ and a data matrix $W_{src} \in \Re^{k \times n_{src}}$ while regularizing the denseness of the results with an elastic net[154]:

$$H_{src}, W_{src} = argmin_{W,H} \left(\tfrac{1}{2} ||X_{src} - HW||_{Fro}^2 + \alpha_{NMF} \lambda_{NMF} \left(||vec(H)||_1 + ||vec(W)||_1 \right) + \tfrac{\alpha_{NMF}}{2} (1 - \lambda_{NMF}) \left(||H||_{Fro}^2 + ||W||_{Fro}^2 \right) \right)$$

   Here, λ_{NMF} is the mixing parameter controlling the L1 and L2 regularization and α_{NMF} is the penalty multiplier.

As an initial starting point, W^*_{src} for W_{src} we provide a one-hot-encoding of the given cluster labels y^{src}, where a non-zero entry in the j-th row of column i in W^*_{src} indicates that cell i is a member of cluster j.

2. Given the learned dictionary $H_{src} \in \Re^{g \times k}$ from step (1) and assuming the genes in source and target data correspond, we now transfer knowledge from the source to the target dataset through the dictionary by learning a target data matrix $W_{trg} \in \Re^{k \times n_{trg}}$:

$$W_{trg} = argmin_W \left(\tfrac{1}{2} \left\| X_{trg} - H_{src} W \right\|^2_{Fro} \right)$$

3. To enable domain adaptation for different levels of cell type overlap between the two datasets, we now construct a new target dataset X^{new}_{trg} based on a convex combination of a reconstructed target dataset $H_{src} W'_{trg}$ and its original version X_{trg}:

$$X^{new}_{trg} = \theta H_{src} W'_{trg} + (1 - \theta) X_{trg} \text{ with } 0 \leq \theta \leq 1$$

θ is a mixture parameter indicating how strongly knowledge from the source dataset should be transferred into the newly constructed target dataset. High values of θ indicate a strong influence of the source dataset on the modified dataset and low values cause the new dataset to be more similar to its original version.

The target reconstruction or clustering matrix $W'_{trg} \in \{0,1\}^{k \times n_{trg}}$ is a simplified version of W_{trg} with ones at the positions of all column-wise maxima and zeros elsewhere, *i.e.*

$$\left(W'_{trg}\right)_{li} = 1_{\left[\left(argmax_{l \in \{1, \ldots, k\}} \left(W_{trg}\right)_{li}\right) = l\right]} \forall i, l .$$

Using W'_{trg} instead of W_{trg} corresponds to reducing the information in this matrix to potential cluster memberships of the target cells, which is appropriate considering the task at hand. To this end, a number of different approaches were implemented (*e.g.* leaving W_{trg} as it is or optimizing it in an additional training step), but it was found that taking the simplified version as described above performed best and most consistently for all scenarios under investigation.

4. The newly derived dataset X^{new}_{trg} can be used as input for a clustering method. To predict $\hat{y}^{trg}$, we use single-cell consensus clustering (SC3)[84] as an exemplary clustering method that is commonly used to solve scRNA-Seq clustering problems. See **Chapter 2.2.2** for a detailed description of SC3.

Please note that the proposed method does not inherently depend on the number of samples in each dataset and can technically (even though not studied in this dissertation) be used to transfer knowledge from datasets of any size, not just from a source that is larger than the target.

The mixture parameter θ dictates how much the newly constructed target dataset should be changed by the information in the source dataset. θ is automatically chosen via an unsupervised assessment of the clustering quality through Kernel Target Alignment (KTA) scores[314], which measure the similarity of kernels. The whole transfer learning and clustering procedure (steps 1 - 4) is computed with a number of values for θ within a prespecified range and the KTA score between the linear kernel of the mixed dataset X_{trg}^{new} over the cells and the linear kernel of the cell type labels predicted by subsequent SC3 clustering is calculated. The parameter value yielding the optimal KTA score is chosen for the final result and can give an indication of the transferability between source and target data. An investigation of the mixture parameter of the transfer learning approach and its automatic selection process based on KTA scores is given in **Figure 32** of **Chapter 4.3.1**, where the relationship of the unsupervised KTA scores and their supervised counterpart, the ARIs, are examined.

If no reliable cluster labels are available for the source dataset $X_{src} \in \Re^{g \times n_{src}}$, one can choose to generate those labels via NMF clustering[23–25] and proceed as if they were the real labels y^{src}. This basically consists of learning a dictionary $H_{src} \in \Re^{g \times k}$ and a data matrix $W_{src} \in \Re^{k \times n_{src}}$ as described above in step 1 and selecting the cluster memberships based on the column-wise maxima of W_{src}, *i.e.* $\widehat{y}_i^{src} = argmax_{l \in \{1, ..., k\}}(W_{src})_{li}$.

Instead of learning the source labels through NMF clustering, one could also avoid providing any initial starting point W_{src}^* for W_{src} when learning the dictionary H_{src} and the data matrix W_{src}.

4.2.3 Datasets and corresponding validation strategies

We analyze a number of different datasets during the current study in order to validate the effectiveness of the proposed method. First, we generate synthetic scRNA-Seq data, where the true underlying cluster structures are known and the method's performance can be investigated in a controlled environment. Afterwards, we generate target and source datasets by subsampling a real scRNA-Seq dataset (Tasic *et al.* (2016)[291]) and hence guarantee an overlap in clustering structures and ensure transferability between the two datasets. Lastly, we investigate two independent datasets and cluster a real target dataset (Hockley *et al.* (2019)[286]) by transferring prior knowledge from a well-known source dataset (Usoskin *et al.* (2015)[82]). The different datasets and corresponding validation strategies are presented in the following sections.

Validation on generated source and target datasets

To test the applicability of our method, we first use it on simulated count level scRNA-Seq data from a defined hierarchical set of clusters that represent the different cell types present in a tissue or biofluid. **Figure 26** shows a graphical representation of the hierarchical clustering structure used to generate the simulated data. Each generated dataset consisted of eight clusters of cells (1-8) deriving from five top-level clusters (V - Z) that share a common background distribution of gene expression levels and some proportion of genes differentially expressed between them.

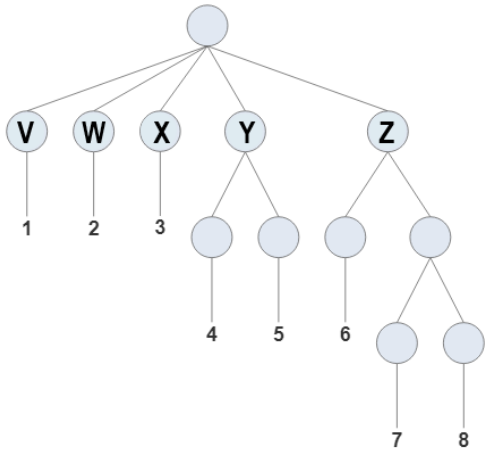

Figure 26: Clustering structure of scRNA-Seq simulation data. Count level scRNA-Seq data is simulated according to a predefined hierarchical clustering structure with eight cell clusters (1-8) that are derived from five top-level clusters (V - Z). Generated datasets are individually split up by randomly assigning the top node clusters V - Z to source or target. Three different settings are considered: 1. Both source and target data contain cells from all top node clusters V - Z (Complete overlap), 2. Three randomly selected top node clusters V - Z are chosen as common to both source and target, the other two are assigned to either one of source and target (Incomplete overlap) or 3. Cells from two of the top-node clusters form the target dataset and cells from the other three top-node clusters form the source (No overlap).

An outline of the data generation procedure is given here. The full code is provided in https://github.com/nicococo/scRNA/blob/master/scRNA/simulation.py.

First, we generate the number of cells in each sub-cluster using a Dirichlet distribution with a concentration parameter of 10. Then we define a common background distribution of gene expression levels sampled from a gamma distribution with shape 2 and rate 0.1. For each cluster, we randomly select 10-40 % of genes to be differentially expressed relative to the background. The difference in expression for each such gene, expressed as a $\log_2$ fold change, is sampled from a normal distribution with a mean of 1 and a standard deviation of 0.5. For clusters that are not themselves top-level clusters (clusters 4-8), this process continues recursively with further expression differences generated for each sub-cluster using the parent cluster as the new background until the final clusters are reached. Finally, we generate count level data by applying a small amount of random normally distributed noise to the expression levels of each cell and

then sampling the per gene counts from a negative binomial distribution with dispersion 0.1. The resulting datasets contain cells with a median count of 215,500 reads per cell. Please see **Chapter 4.2.4** and, in particular, **Figure 27** for details.

Once count level data is generated for the entire dataset, we split it into target and source datasets with different sets of cells according to the cluster structure and the relationship between the target and source. Here, we consider three such relationships that reflect three possible experimental scenarios:

- Cells in the target and source are randomly sampled from the same underlying tissue or biofluid and hence contain cells from all top node clusters V - Z.
- Certain clusters are specified to be only present in the source and some to be only present in the target; the remaining clusters are present in both target and source. Three randomly selected top node clusters V - Z are chosen as common to both source and target; the other two are assigned to either one of source and target.
- The cells in the target and source are drawn from completely non-overlapping clusters. In this scenario, transfer learning is not expected to be successful. Cells from two of the top node clusters form the target dataset and cells from the other three form the source.

The genes measured in source and target are always the same and the top nodes are randomly assigned to either source or target for each repetition of the data generation process. We generate 100 sets of simulated data for each of the three settings simulating the expression levels of 10,000 genes in 1,800 cells. We assign 1,000 cells to the source dataset and the set of 800 target cells is downsampled (*i.e.* 10, 50, 100, 200, 400, 600, 800 target cells) to investigate the performance of the transfer learning approach and its corresponding baseline methods when applied to datasets of varying sizes.

The results of the proposed method applied to the generated source and target datasets and evaluated based on the known underlying clustering structures are presented in **Chapter 4.3.1**.

Validation on subsampled source and target datasets

Following the analysis of simulated data, we subsequently examine a real scRNA-Seq dataset. By subsampling both source and target datasets from the same single original dataset, we create an environment where the potential benefit of transfer learning can be determined on real-world gene expression data. For this, we utilize gene expression data provided as RPKM derived from over 1,600 cells of the primary visual cortex of the adult mouse brain[291].

We run 100 repetitions splitting the data into a source dataset of 1,000 cells and a target dataset of 650 cells each time, which is subsampled even further (to 25, 50, 100, 200, 400, 650 target cells) to assess performance for different sample sizes. To investigate the influence of complete and incomplete overlap between the clusters of source and target datasets, transcriptomic cell types assigned to either dataset are controlled. Complete overlap means randomly assigning cells into source and target. Incomplete overlap is achieved by assigning the two largest clusters of the dataset (*Glutamatergic L4* cells and *GABAergic Pvalb* cells) to be either an exclusive source cluster or an exclusive target cluster, respectively. All other clusters are shared amongst both source and target in this setting.

The transfer learning approach and its baseline methods are now investigated under two different conditions. Firstly, we assume that no ground-truth labels are available and generate labels for 18 cell clusters via NMF clustering[23–25] on the whole dataset. We interpret this clustering, based as it is on the totality of the data, as a ground-truth clustering and apply our method and the baseline algorithms to a subset of the dataset to see how each method performs relative to this definition of the ground truth when not all of the data is available. Secondly, we use the data-driven clustering labels provided in the original paper and take those as the ground-truth labels. Specifically, we use a cut-off in the provided clustering hierarchy that results in 18 clusters. Given those alternative ground-truth labels, we once again run the proposed transfer learning method and its competitors as described in **Chapter 4.2.5**.

The validation results using the subsampled source and target datasets are presented in **Chapter 4.3.2**.

Validation on independent source and target datasets

As a real-world application of the transfer learning approach, we analyze two entirely independent but biologically related datasets. To improve the clustering results of a relatively small target dataset from Hockley *et al.* 2019[286], we transfer knowledge from a larger source dataset from Usoskin *et al.* 2015[82], both derived from the rodent somatosensory system. The somatosensory system is responsible for detecting mechanical, thermal and chemical stimuli to which an organism can choose to elicit a behavioral response. Primary sensory neurons innervate the vast majority of internal hollow organs, joints, muscles and the skin evoking conscious sensation in the event of these stimuli. This is most clearly exemplified by pain in the case of potentially harmful or noxious stimuli, such as burning or cutting the skin. In Usoskin *et al*., transcriptomic analysis of 622 primary sensory cell bodies, which reside within the dorsal root ganglia (DRG), reveals significant diversity in cell type (11 types) and sensitivity to a diverse range of stimuli modalities (*e.g.* thermosensitive, itch sensitive, nociceptive) to which an organism is exposed. However, previous retrograde tracing experiments show that only 5-10 % of DRG neurons project to internal (visceral) targets, such as the

gastrointestinal tract, and as such are likely only represented by ~ 30-60 cells in the Usoskin *et al.* dataset. Such small cell numbers limit subtype assignment of cells in this organ. In order to overcome this limitation, scRNA-Seq has been performed on retrograde-labeled DRG neurons known to selectively innervate the gastrointestinal tract (colonic DRG neurons), providing transcriptomic analysis of 314 cells from this specific organ that cluster into seven distinct subtypes[286]. However, it is unclear whether *de novo* clustering of colonic DRG neurons identifies established clusters previously identified in larger datasets such as Usoskin *et al.* (hereafter designated 'Usoskin') or whether novel cell types exist within this dataset (hereafter designated 'Hockley').

In initial experiments, the original source and target data are used; however, in later experiments, a batch effect removal approach is applied to control for the integration of single-cell transcriptomic data across different conditions and technologies. Here, we apply Seurat batch effect removal[297] to combine the Hockley and the Usoskin data and separate the result back into the original datasets, which are then provided to the proposed transfer learning method and its competitors as described in **Chapter 4.2.5**.

As an additional preprocessing step, we investigate the effect of imputation on the clustering results. MAGIC[315], a widely used method for imputing missing values to overcome zero-inflation in scRNA-Seq data, is applied to both datasets and the preprocessed datasets are then provided to the three methods under investigation.

Using either the original datasets or the preprocessed, batch effect removed or imputed datasets, the results of the proposed transfer learning method and its competitors as described in **Chapter 4.2.5** are assessed in terms of performance via comparison to the clustering of the original paper[286], evaluation of t-SNE plots[79] and examination of differentially expressed genes to determine putative cellular functions of neuronal subtypes. Since SC3 is a non-convex method, it yields different results for each run. In order to provide quantification of the stability of the three methods, we apply each method 1,000 times and count the number of times three key clusters of interest are successfully identified. These clusters are selected based on their biological relevance as described in the original paper, further details can be found in the results section.

Once again, experiments are run under two conditions. Firstly, we assume that reliable source data labels are not available and we generate cell labels for the Usoskin dataset via NMF clustering. Secondly, we use labels from Usoskin *et al.* (generated via an iterative PCA approach). Usoskin *et al.* provide labels at three different levels of the hierarchy producing 4, 8 or 11 clusters. We investigate results based on all of those, calling them level 1, 2 and 3 labels, respectively. We also investigate a scenario where we generate the labels via NMF clustering instead of using the labels presented in Usoskin *et al*. In the main text, however, we only present results based on using level 3 labels from the original publication. Please see **Appendix Chapter III.** for the clustering results using NMF, level 1 and level 2 labels for the source datasets.

To assess the performance of our method, we are unable to compute ARI scores in this setting. In contrast to the simulations described above, the true underlying clustering architecture of the cells under study is largely unknown. Hence, we assess clustering performance based on differential gene expression and biological relevance to known somatosensory pathways. The validation results using the independent source and target datasets are presented in **Chapter 4.3.3**.

4.2.4 Preprocessing and parameter selection

Three steps for preprocessing scRNA-Seq data are applied:

- **Cell filter:** Remove all cells containing fewer than x_{genes} genes with $expression > x_{expression}$.
- **Gene filter:** Remove ubiquitous genes that are expressed in almost all cells (*i.e.* with $expression > x_{expression}$ in at least $x_{cells}\%$ of cells) and rare genes that are not expressed in almost all cells (*i.e.* with $expression < x_{expression}$ in at least $x_{cells}\%$ of cells).
- **Log-transformation:** Log-transform the expression matrix after adding a pseudo-count of 1.

Preprocessing is performed once for all datasets (source and target separately) before the different clustering methods (*i.e.* transfer learning or baseline methods) are applied and are repeated, for example, after the concatenation of two datasets. All free preprocessing parameters should be selected by future users based on an inspection of the data, *i.e.* expression histograms of both source and target dataset.

The mixture parameter θ is automatically chosen via an unsupervised assessment of the clustering quality through KTA scores as described in **Chapter 4.2.2**. See **Chapter 4.3.1** for an investigation of the performance changes induced by varying θ on the generated datasets.

Other free parameters, *e.g.* the elastic net parameters λ_{NMF} and α_{NMF}, are selected based on results from the simulated data. The generated datasets are used to determine performance changes induced by varying the free parameters of the method and identify optimal settings, which are assumed to be good choices for the application of the proposed method to real datasets.

The specific values of the free parameters of the preprocessing steps and the TransferCluster method selected for the datasets in this dissertation and the corresponding expression histograms are presented in the following sections.

Preprocessing and parameter selection for generated source and target datasets

After generating 100 datasets with 1,800 cells and 10,000 genes, the median overall number of reads for each cell is 215,500 reads. The corresponding histogram is shown in **Figure 27**.

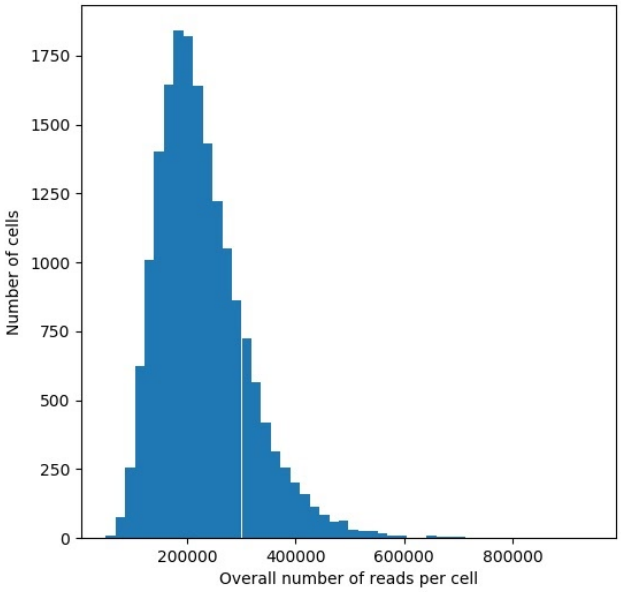

Figure 27: Histogram of cell counts in generated scRNA-Seq datasets. 100 datasets with 1,800 cells and 10,000 genes are generated and a histogram of the overall number of reads is shown.

The preprocessing steps are not applied to the generated datasets because the generation process does not produce any unfavorable genes or cells.

For each overlap setting (described in **Chapter 4.2.3** on generated datasets), this process is repeated and the 100 datasets are separated into 1,000 source cells and 800 target cells. All three competitor methods are applied to down-sampled target datasets where for each repetition, 10, 25, 50, 100, 200, 400, 600 and 800 cells are randomly selected from the complete target dataset.

There are a number of parameters in the NMF step of the proposed method that need specification. In the controlled environment of the generated datasets, the elastic net parameters are set to $\alpha_{NMF} = 10.0$ and $\lambda_{NMF} = 0.75$ and the maximum number of iterations until convergence up to a relative error of 0.001 is set to 4,000. The range of mixture parameters θ to be put into the KTA score selection process is [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]. The number of clusters to find is selected in agreement with the true underlying cluster structure.

Preprocessing and parameter selection for subsampled source and target datasets

Please see **Figure 28** for the histogram of the expression levels in the Tasic dataset[291], which is investigated to choose the parameter values for preprocessing. Before preprocessing, the original Tasic dataset contains 1,679 cells and 24,057 genes. The parameters of the preprocessing filters as described above are set to $x_{genes} = 2000$, $x_{expression} = 2$ and $x_{cells} = 94$ following the inspection of the expression histogram in **Figure 28**. After removing 21 cells containing fewer than 2,000 genes with expression > 2 and 14,510 genes with expression < 2 or > 2 in at least 94% of cells, the dataset contains expression levels of 9,547 genes in 1,658 cells. The expression matrix is log-transformed after adding a pseudo-count of 1.

We deem this dataset to be of sufficient complexity in terms of taxonomic diversity (it contains 23 GABAergic neuronal, 19 glutamatergic neuronal and 7 non-neuronal cell types) and in terms of total cell count to enable cluster-restricted subsampling and thus the application of transfer learning approaches.

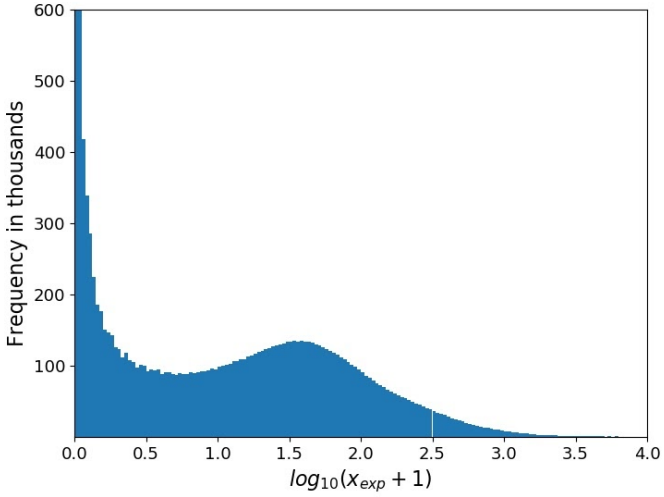

Figure 28: Histogram of expression values in Tasic dataset[291]. For 24,056 genes and 1,679 cells, there are a total of 40,390,024 gene expression values. 27,596,688 of those equal zero. x- and y-axes are cropped. The location of the frequency minimum after the zero-inflation at 0.5 implies choosing 2 as the cut-off expression value for preprocessing.

The free parameters in the NMF step of the method are chosen according to the best results in the controlled environment of the generated datasets, *i.e.* $\alpha_{NMF} = 10.0$ and $\lambda_{NMF} = 0.75$ and the maximum number of iterations until convergence up to a relative error of 0.001 is set to 4,000. The range of mixture parameters θ to be put in the KTA score selection process is [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0].

A number of adjustments have to be made when the data-driven clustering labels of the original publication are used for the source data and not the generated NMF labels. After careful investigation of the Tasic data with the labels from the original publication, it is best to avoid having very high mixture parameters. Consequently, the

range of mixture parameters θ to be put in the KTA score selection process is [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]. The parameters of NMF are set to $\alpha_{NMF} = 1.0$ and $\lambda_{NMF} = 1.0$ in this case, indicating that a stronger L1 regularization is favorable here. The number of clusters to find is selected in agreement with the true underlying cluster structure.

Preprocessing and parameter selection for independent source and target datasets

Please see **Figure 29** for the histogram of the expression levels in the Hockley[286] and Usoskin[82] datasets, which are investigated here to choose the parameter values for preprocessing. Before preprocessing, the original Hockley dataset contains 314 cells and 45,513 genes. The parameters of the preprocessing filters described above are set to $x_{genes} = 2000$, $x_{expression} = 1$ and $x_{cells} = 94$ after inspection of the expression histogram in **Figure 29 A**. No cells contain fewer than 2,000 genes with expression > 1 and 35,862 genes with expression < 1 or > 1 in at least 94% of cells are removed. The dataset now contains expression levels provided as TPM of 9,651 genes in 314 cells. The expression matrix is log-transformed after adding a pseudo-count of 1.

Before preprocessing, the original Usoskin dataset contains 622 cells and 2,0191 genes and provides gene counts as CPM. The parameters of the preprocessing filters are set to $x_{genes} = 2000$, $x_{expression} = 1$ and $x_{cells} = 94$ following the inspection of the expression histogram in **Figure 29 B**. After removing 121 cells that contain fewer than 2,000 genes with expression > 1 and 10,911 genes with expression < 1 or > 1 in at least 94% of cells, the dataset now contains expression levels of 9,280 genes in 501 cells. The expression matrix is log-transformed after adding a pseudo-count of 1.

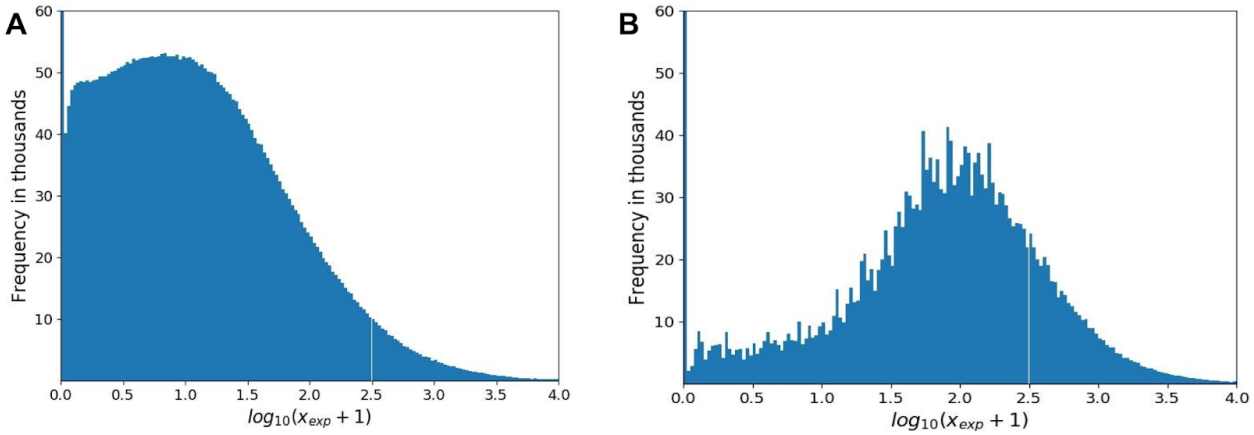

Figure 29: Histogram of expression values in the Hockley[286] and the Usoskin[82] dataset: A Histogram of all expression values in the Hockley dataset. For 45,513 genes and 314 cells, there are a total of 14,291,082 gene expression values. 10,181,090 of those equal zero. x- and y-axes are cropped. The location of the frequency minimum after the zero-inflation at 0.25 implies choosing 1 as the cut-off expression value for preprocessing. **B** Histogram of all expression values in the Usoskin dataset. For 20,191 genes and 622 cells, there are a total of 12,558,802 gene expression values. 10,368,845 of those equal zero. x- and y-axes are cropped. The location of the frequency minimum after the zero-inflation at 0.25 implies choosing 1 as the cut-off expression value for preprocessing.

The proposed transfer learning method, as well as its competitor method of concatenating the source and the target dataset (as described in **Chapter 4.2.5**), can only be applied when the set of genes in the source and the target dataset are identical. Using only the subset of 4,402 genes that appear in both sets, the Hockley dataset now contains 4,402 genes and 314 cells and the Usoskin dataset contains 4,402 genes and 501 cells. The free parameters in the NMF step of the method are selected according to the best results in the controlled environment of the generated datasets, *i.e.* $\alpha_{NMF} = 10.0$ and $\lambda_{NMF} = 0.75$ and the maximum number of iterations until convergence up to a relative error of 0.001 is set to 4,000. In order to assess whether rare cell types are present in the Hockley dataset, the number of clusters to group the cells of the target dataset in is chosen to be $k = 7$, which is the number of cell types identified in the original Hockley publication. The mixture parameter θ is, again, selected automatically (see **Chapter 4.2.2**).

4.2.5 Baseline methods

For assessing the quality of our unsupervised transfer learning solution, we are interested in investigating the clustering accuracy of our method on a target dataset compared to two competitor methods. As baseline methods, we implement the original SC3 clustering method on the target dataset alone (TargetCluster) and on the concatenated dataset of source and target (ConcatenateCluster). For a visualization of the baseline methods, see **Figure 30**.

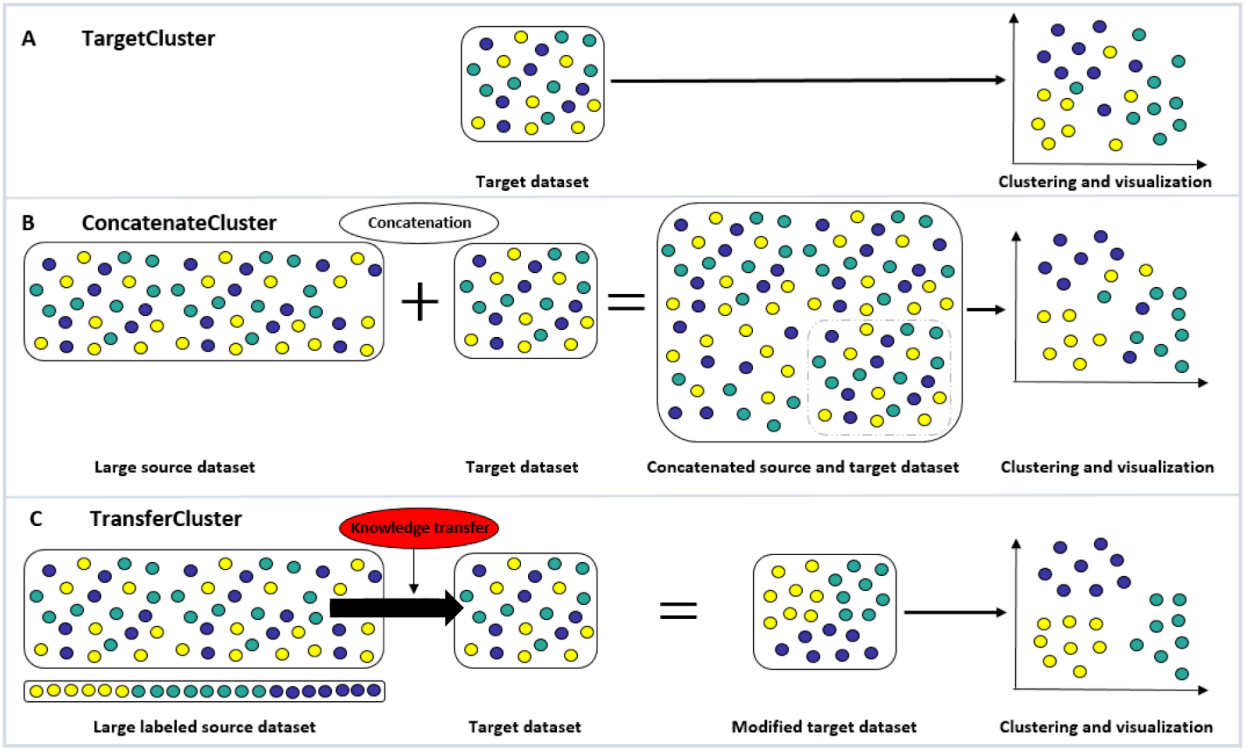

Figure 30: Visualization of the three competitor methods for analyzing scRNA-Seq data. A TargetCluster. Clustering is applied to the target dataset alone. **B** ConcatenateCluster. Source dataset and target dataset are combined into one large dataset via simple concatenation before clustering the new dataset as a whole. Performance measures (*i.e.* accuracies) are calculated on the target dataset only since it is the main focus of interest for clustering. **C** TransferCluster. The proposed method of knowledge transfer is applied to the target dataset learning from a large labeled source dataset. The resulting, modified target dataset is then provided to the clustering procedure.

4.2.6 Performance metrics

As a supervised performance metric, we use the adjusted Rand index (ARI)[316], which measures the similarity between two data clusterings adjusted for the chance of randomly grouping datapoints together in one cluster. In order to evaluate the clustering result of an algorithm, it can be used to compare the given clustering to the true underlying class labels. Here, we compare the transfer learning results (TransferCluster) and the baseline results (TargetCluster and ConcatenateCluster) with the known clustering labels. These are known perfectly in the case of the simulated data and retrieved from the original publication in the case of the real data. ARI scores are computed only on the target data, even in the case of ConcatenateCluster, where labels are computed for both source and target cells.

4.3 Results

In the following sections, we present the results of the proposed transfer learning method evaluated on generated, subsampled and independent target and source datasets. When the underlying truth is known or can be estimated, performance is investigated in terms of similarity (ARI scores) to the true clusterings. A functional analysis of the corresponding transcripts and their translational outcomes is performed where the underlying clustering structures are unknown. In both settings, performance is examined in comparison to a number of baseline methods, which are presented in full detail in **Chapter 4.2.5**.

4.3.1 Results on generated datasets

To assess the performance of the proposed method in comparison to the two baseline methods in a controlled environment, we conduct a number of simulation experiments with generated data, where the ground truth of the clustering structure is controlled and known. This allows us to compute supervised performance metrics for each method and make objective statements about which method performs best. **Figures 31 B-D** show the ARI curves of all three methods on the simulated scRNA-Seq datasets generated according to the clustering structure in **Figure 31 A** (from **Figure 26**) for the three different settings of overlap between the source and the target, as described in **Chapter 4.2.3** ("*Validation on generated source and target datasets*").
For complete overlap in the clustering structures of the two datasets, *i.e.* identically sampled data, our method, TransferCluster, outperforms the baseline methods for all sample sizes of the target dataset (**Figure 31 B**). It exceeds not only the clustering on the target dataset alone (TargetCluster) but also performs better than concatenating and clustering source and target data simultaneously (ConcatenateCluster). The latter can improve the clustering of the target dataset but fails to achieve the same levels of performance as TransferCluster. The main reason for this is that instead of predicting the labels of the source dataset - like ConcatenateCluster - TransferCluster uses the true source labels and incorporates that knowledge into the clustering of the target dataset. This effect is very strong here since the true source labels are completely known for the generated datasets.
The ARI curves on simulated data with both overlapping and non-overlapping clusters in source and target data show that, in this case, transferring knowledge can still help the analysis of the target dataset and that TransferCluster outperforms both baseline methods, however not by the same amount as when a complete overlap is present (**Figure 31 C**). Concatenating the two datasets (ConcatenateCluster) can lead to decreased performance for larger target sample sizes, where clustering the target data alone (TargetCluster) is more successful. Only incorporating the source knowledge via our transfer learning procedure (TransferCluster) can consistently improve the clustering results for all sample sizes.

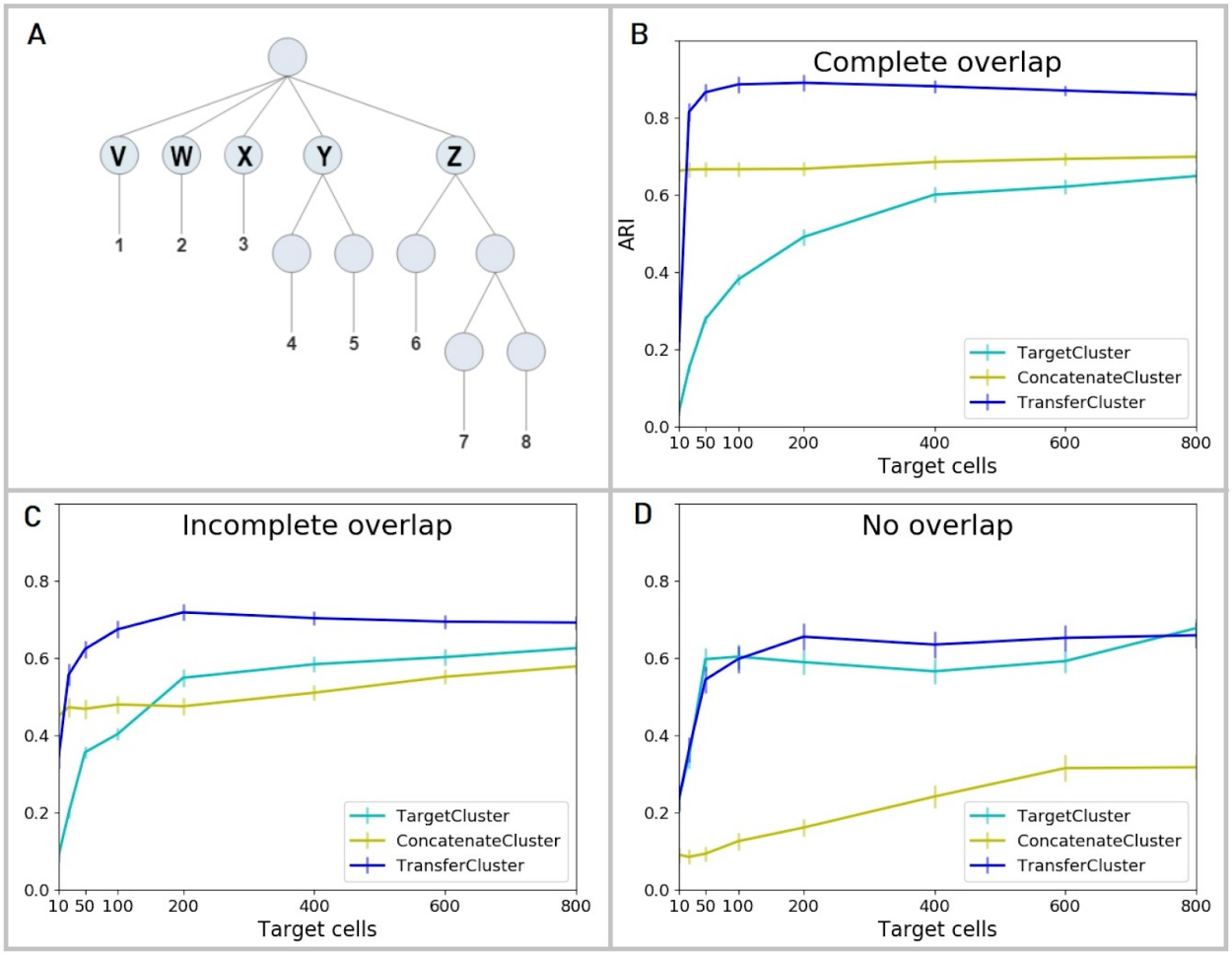

Figure 31: Performance curves of the three competitor methods on generated datasets for different levels of overlap. A Count level scRNA-Seq data is simulated according to a predefined hierarchical clustering structure with eight cell clusters (1-8) that are derived from five top-level clusters (V - Z). Generated datasets are individually split up by randomly assigning the top node clusters V - Z to source or target. Three different settings are considered: 1. both source and target data contain cells from all top node clusters V - Z (Complete overlap), 2. three randomly selected top node clusters V - Z are chosen as common to both source and target, the other two are assigned to either one of source and target (Incomplete overlap) or 3. cells from two of the top node clusters form the target dataset and cells from the other three form the source (No overlap). **B** Clustering performances of the baseline methods, TargetCluster (clustering on the target dataset alone) and ConcatenateCluster (concatenating and clustering source and target data simultaneously) and the transfer learning approach (TransferCluster) when the clustering structures of source and target data are identical (Complete overlap). **C** Clustering performances of the baseline methods, TargetCluster and ConcatenateCluster and the transfer learning approach (TransferCluster) for an incomplete overlap between the cell clusters in source and target data (Incomplete overlap). **D** Clustering performances of the baseline methods, TargetCluster and ConcatenateCluster and the transfer learning approach (TransferCluster) for a setting with two exclusive target top nodes and three exclusive source top nodes and no cell types that appear in both sets (No overlap). Please note that due to the sampling procedures described above, the number of top-level nodes in the target datasets decreases from 5 in B to 4 in C and 2 in D and hence the performance of TargetCluster improves from B to D. 95% confidence intervals are shown.

Specifically, one should note that the performance, as measured by ARI, of ConcatenateCluster decreases when there is a non-perfect overlap (in comparison to a complete overlap) and is greatly impaired when there are no overlapping clusters in source and target data. Combining two sets into one is not to be preferred in those cases.

The ARI curves on disparate, non-overlapping clusters show that, as expected, transferring information from a source dataset that is unconnected to the target dataset cannot improve clustering significantly (*i.e.* confidence intervals of TargetCluster and TransferCluster overlap) and using SC3 on the target dataset alone (TargetCluster) is to be preferred (**Figure 31 D**). For two exclusive target top nodes and three exclusive source top nodes and no cell types that appear in both sets (No overlap), concatenating source and target into one dataset (ConcatenateCluster) has a negative effect on the clustering of the target cells and should be avoided. Importantly and in contrast to the ConcatenateCluster, the use of TransferCluster does not significantly reduce clustering performance compared to *de novo* clustering of the target data alone and can keep the levels of performance as high as not taking the source data into account at all, as the method can choose a low mixture parameter when there is no overlap. To conclude, the transfer learning approach outperforms both baseline methods and works as expected for simulated scRNA-Seq data.

We now investigate the effect of the mixture parameter θ, which dictates how much the newly constructed target dataset should be influenced by the information of the source dataset. See **Chapter 4.2.2** for a detailed description of the parameter selection procedure of θ. It is automatically selected via an unsupervised assessment of the clustering quality through KTA scores[314], which measure the similarity of kernels. The whole transfer learning and clustering procedure is applied with a number of values for θ within a prespecified range and the KTA scores between the linear kernel of the mixed dataset (not its original version) over the cells and the linear kernel of the predicted labels are calculated. The scores give an indication of how well the predicted labels are represented in the mixed dataset and thus show how well the clustering procedure performs for the corresponding parameter value. The parameter value yielding the optimal KTA score is chosen as the parameter for the final clustering computation and can give an indication of the transferability between source and target data. Low values mean source and target do not match very well (*i.e.* low transferability) and high values hint at high similarities (*i.e.* high transferability).

The simulation study on generated scRNA-Seq data is used to investigate the performance of this parameter selection procedure. **Figure 32** gives insight into the procedure within TransferCluster that automatically selects the mixture parameter θ based on KTA scores. The first row of performance plots shows the original results on the generated datasets, which can also be found in **Figure 31**. The second row presents the results of TransferCluster for a number of fixed mixture parameter values θ. The investigation of the mixture parameter θ of the proposed method for various levels of overlapping cluster structures in source and target data shows that it has to be chosen carefully. Zero mixture corresponds to not modifying the target dataset at all, *i.e.* not transferring any knowledge from the source dataset (equals TargetCluster). Depending on the overlap in the clustering structures of source and target data, increasing the mixture parameter might improve the performance up to a certain point and then

decrease when there is an incomplete overlap. A high overlap makes the use of high mixture values necessary. If there is low or no overlap, one needs to use low values or avoid using the method.

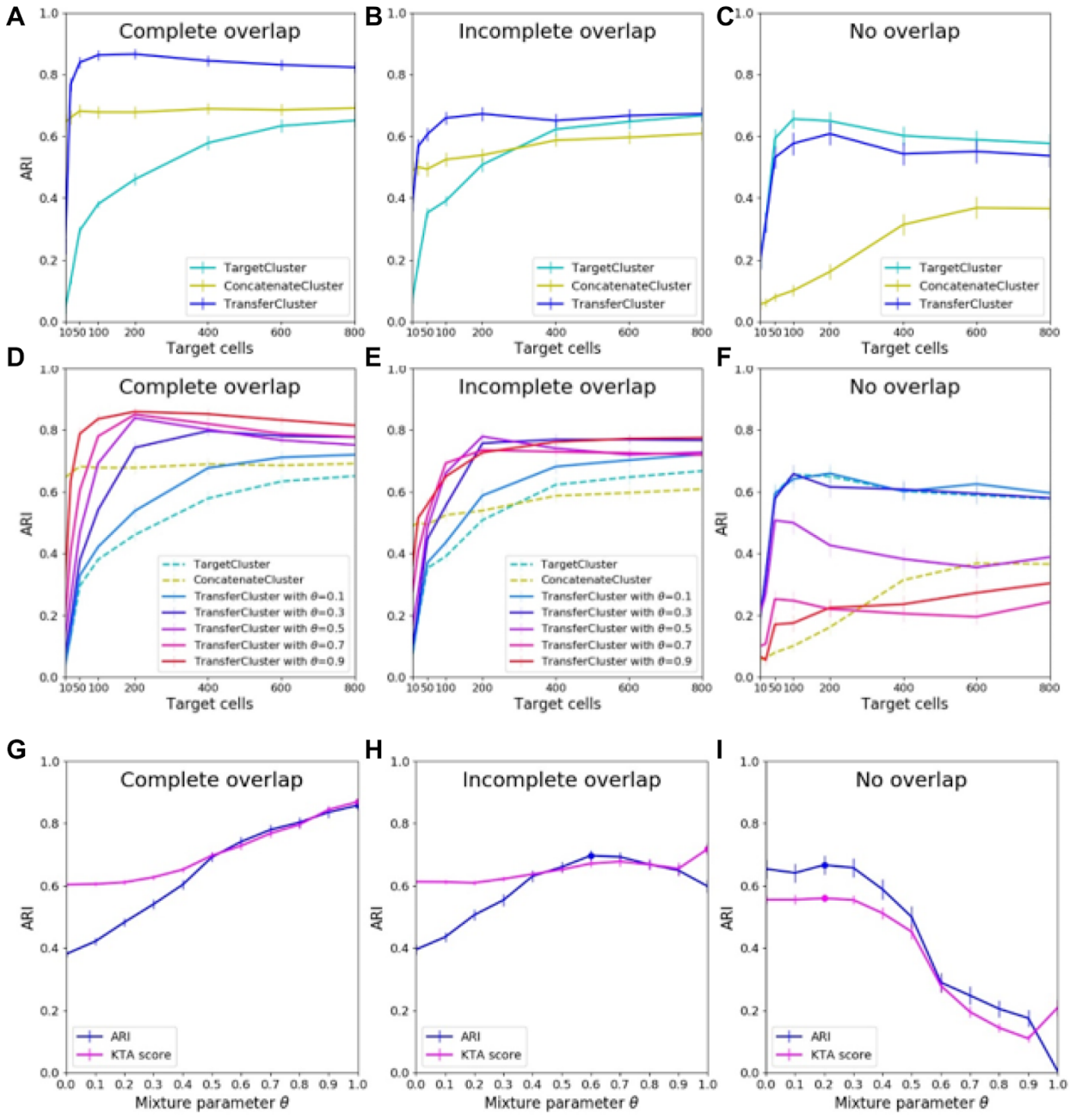

Figure 32: Performance curves of the three competitor methods on generated datasets and investigation of the mixture parameter selection process of the transfer learning method for different levels of overlap. A Main results of the three competitor methods (as seen in **Figure 31**) for three different settings of overlap in the cluster structures of source and target data: Complete, incomplete and no overlap. **B** Results of the baseline methods and TransferCluster for a number of fixed mixture parameter values θ. The complete range of θ values is [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]. Not all are shown for greater clarity. **C** Influence of the mixture parameter θ on both the supervised performance measure ARI and its unsupervised counterpart, the KTA score for an exemplary target sample size of 100 cells (other sample sizes show similar results).

The third row in **Figure 32** shows how the mixture parameter θ influences both the supervised performance measure ARI and its unsupervised counterpart, the KTA score, for an exemplary target sample size of 100 cells (other sample sizes show similar results). For each overlap setting, we investigate how changing the mixture parameter influences performance measured via supervised (ARI) and unsupervised accuracy measures (linear KTA). It can be seen that the curves of the two metrics have very similar shapes for all three overlap settings and, most importantly, have maxima at the same or at least very close parameter values of θ. This supports the theory that KTA scores are a good choice for selecting the mixture parameter θ based on the arguments of the maxima of KTA scores.

4.3.2 Results on subsampled source and target datasets

Now we present the results of subsampling both source and target from the same real scRNA-Seq dataset[291] and comparing the performance of our method to that of the baseline methods. In order to validate our approach for a scenario where no reliable ground-truth labels exist, we first generate synthetic labels of 18 clusters via NMF clustering[23–25] on the whole dataset, which we then consider to be the ground truth for this experiment. **Figures 33 A** and **B** show the corresponding ARI curves for complete and incomplete overlap between source and target dataset. For both scenarios, transferring knowledge into the target dataset improves its clustering in subsequent SC3 clustering and outperforms both baseline methods. When source and target datasets share the complete clustering structure (panel A), concatenating the two datasets (ConcatenateCluster) improves the clustering results of the target data (TargetCluster) but transferring knowledge via the proposed method (TransferCluster) is seen to improve it even more. While for a complete overlap, ConcatenateCluster can improve target clustering by a large margin, especially when the target dataset is relatively small in comparison to the source dataset (for example, 1/10th of source), the method fails to find additional gains over *de novo* clustering of larger target datasets when the clustering structure in source and target are similar but not identical (Incomplete overlap, panel B). In this setting, which is the more realistic one in most cases, ConcatenateCluster does not always perform well and only the knowledge transfer via the proposed method can consistently improve the target clustering results. Hence, it should be the preferred option to incorporate source information into a target clustering.

Now, instead of generating labels of the complete dataset via NMF clustering, we use the data-driven clustering labels provided in the original paper[291] as ground-truth labels and apply the same subsampling procedure as above. **Figures 33 C** and **D** show the corresponding results for complete and incomplete overlap. Again, for both settings, the transfer learning approach improves TargetCluster clustering on target data alone. Knowledge is successfully transferred from the source to the target dataset no matter how big the overlap in the clustering structure of the two sets is.

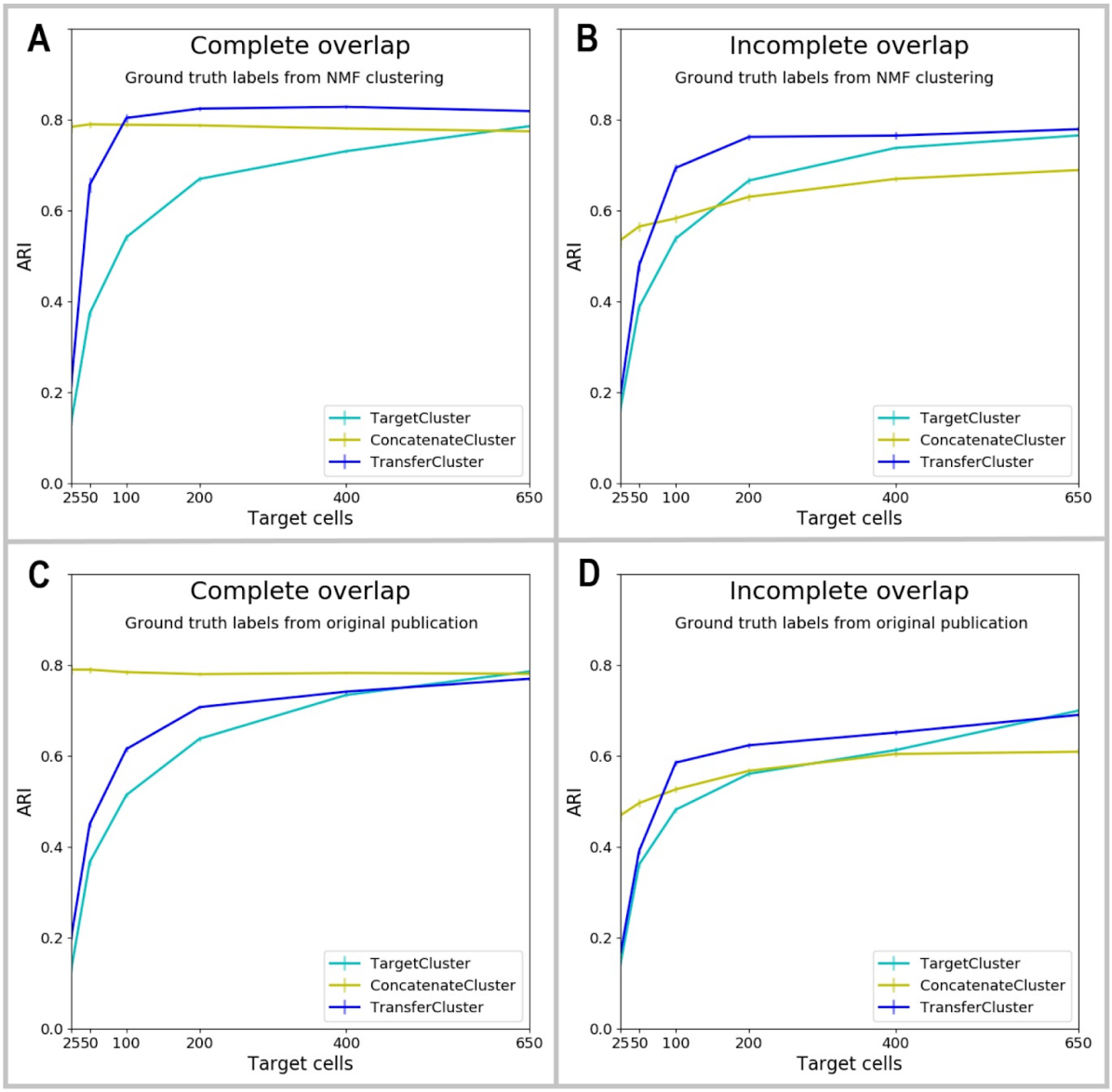

Figure 33: Performance curves of the three competitor methods on subsampled source and target data from mouse visual cortex cells[291] for different levels of overlap. **A** Clustering performances of all three methods using NMF clustering labels of 18 clusters generated on the whole dataset as ground-truth labels. Source and target datasets share the complete clustering structure, *i.e.* all cell types appear in both source and target data (Complete overlap). **B** Clustering performances of all three methods using NMF clustering labels of 18 clusters generated on the whole dataset as ground-truth labels. The overlap is not complete, *i.e.* the two biggest clusters of the dataset are assigned to be either exclusive source or target clusters (Incomplete overlap). **C** Clustering performances of all three methods using the data-driven clustering results from the original paper[291] as ground-truth labels. Source and target datasets share the complete clustering structure, *i.e.* all cell types appear in both source and target data (Complete overlap). **D** Clustering performances of all three methods using the data-driven clustering results from the original paper[291] as ground-truth labels. The overlap is not complete, *i.e.* the two biggest clusters of the dataset are assigned to be either exclusive source or target clusters (Incomplete overlap). 95% confidence intervals are shown.

The comparison to the second baseline method of concatenating both sets into one shows for complete overlap of the clusters in both datasets that transfer learning helps but cannot outperform ConcatenateCluster. However, in the more realistic setting of an incomplete overlap in the clustering structures, concatenating the two datasets has a negative effect on the target clustering, especially for large sample sizes. ConcatenateCluster collapses and it performs even worse than not using the source data at all (TargetCluster) for some larger target sample sizes. Transfer learning is able to avoid this effect and succeeds in incorporating valuable information from the source data into the target data improving its clustering results consistently for all target sample sizes. Transfer learning is clearly to be preferred in this setting.

4.3.3 Results on independent source and target datasets

Leaving the controlled environment where source and target data are sampled from the same distribution, we lastly investigate a real-world application where source and target are completely independent but biologically related datasets collected at different times and places. To assess the performance of our method and the baseline methods, we investigate differential gene expression and biological relevance to known somatosensory pathways. In **Figure 34,** we show t-SNE plots for the Hockley data overlaid with cluster memberships corresponding to the results of methods following the use of the Usoskin data as source. These results correspond to applying the transfer learning approach with the level 3 labels from the original Usoskin[82] publication as *a priori* knowledge about the source dataset. See **Appendix Chapter III.** for a detailed analysis using source labels that are generated via NMF clustering and using level 1 and 2 labels of the original publication.

As predicted, using TargetCluster (*i.e.* the method utilizing SC3 clustering of the Hockley data alone), we identify a similar cluster structure to that observed by the authors in their original study[286]. Specifically, we identify six well-defined clusters (and a 7th poorly defined cluster) that can be separated based on gene expression and also an important anatomical difference related to the spinal region from which the neuron was collected (*i.e.* in **Figure 34 A**, clusters 2 and 4 are both predominantly populated by lumbosacral sensory neurons as indicated by the use of circles, whilst the neurons within the other clusters are mainly thoracolumbar in origin as shown by triangles). In contrast to the original study, TargetCluster does, however, fail to robustly segregate two biologically distinct groups of cells, which, using the author's original nomenclature, are named mNP and mNFa, respectively. In our hands, they correspond to cluster 1 in **Figure 34 A**. The first mNP cluster comprises 15 neurons and expresses Mas-related G-protein coupled receptor D (*Mrgprd*; **Figure 34 B**) and Lysophosphatidic acid receptor 3 (*Lpar3*); genes previously associated with non-peptidergic nociceptive pruriceptors[317]. The second mNFa group of 16 neurons expresses P2Y purinergic receptor 1 (*P2ry1*; **Figure 34 C**) and BAI1-associated protein 2-like 1 (*Baiap2l1*) and is indicative of mechanosensitive nociceptors[318].

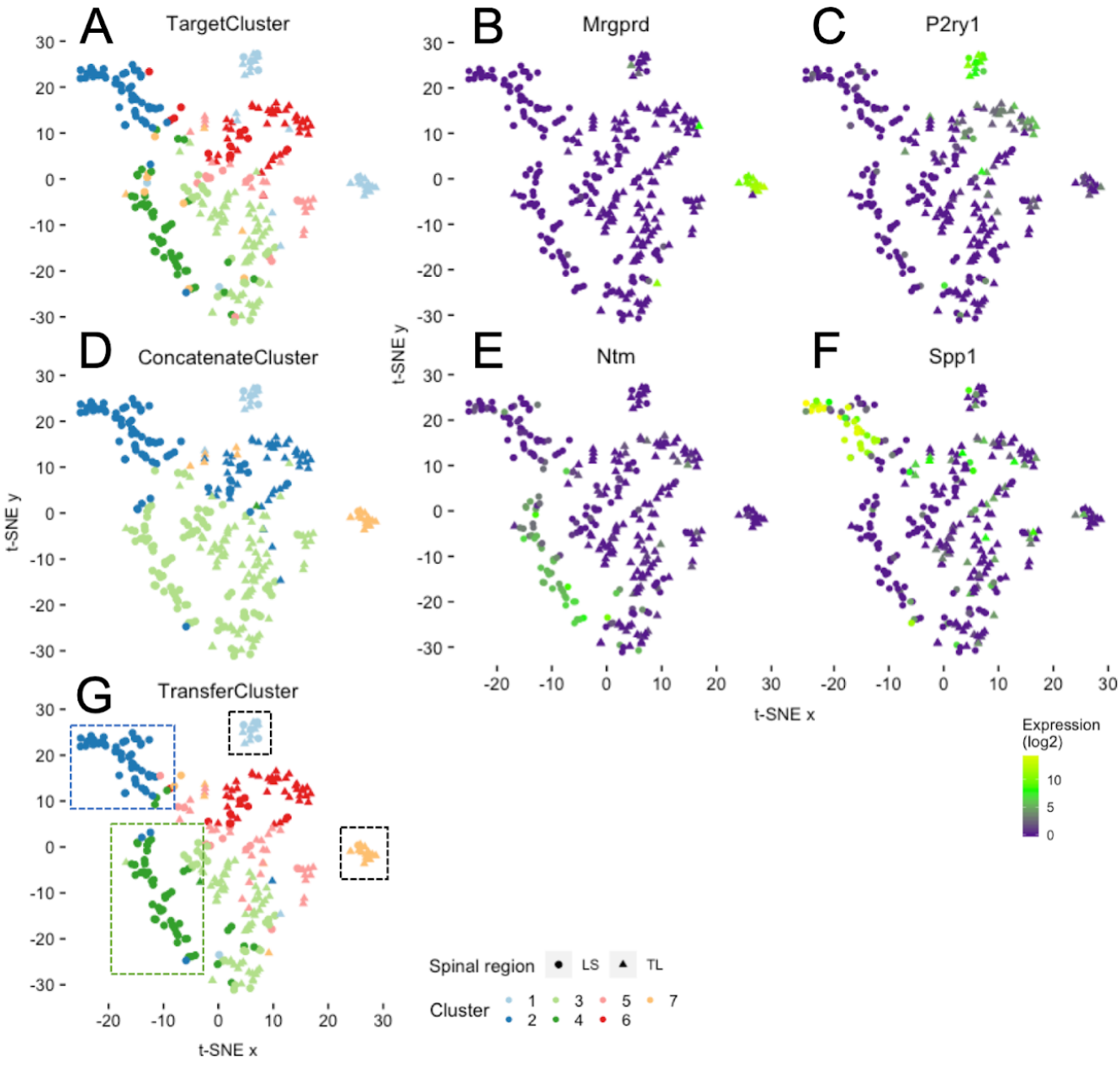

Figure 34: Clustering results of independent source (Usoskin *et al.*[82]) and target (Hockley *et al.*[286]) datasets. t-SNE plots of the mouse colonic sensory neurons from the Hockley dataset are shown. **A** TargetCluster, using only data from Hockley *et al.* to assign clusters. **D** ConcatenateCluster, using a concatenation of data from Hockley *et al.* and Usoskin *et al.* (mouse sensory neurons) to assign clusters. **G** TransferCluster, using the novel transfer learning approach with Usoskin *et al.* as source and Hockley *et al.* as target. Colors in **A**, **D** and **G** refer to the clusters derived from the three approaches. In **G**, clusters 1 and 7 (black dashed boxes), cluster 2 (blue dashed box) and cluster 4 (green dashed box) represent biologically distinct groups of cells with differing sensory functions. This is exemplified by the cluster-specific expression of specific genes by cluster 7 (**B**, Mrgprd), 1 (**C**, P2ry1), 2 (**E**, Ntm) and 4 (**F**, Spp1). Colors in **B**, **C**, **E** and **F** represent expression levels of these genes [log(TPM)]. Shapes refer to the spinal segment from which the neuron was isolated (triangle, TL (thoracolumbar); circle, LS (lumbosacral)).

Whilst SC3 was used by the authors to cluster in their original study, the corresponding algorithm is not deterministic and produces different results when solving the same clustering problem multiple times. Indeed, when we count the number of times the mNP and mNFa clusters are separated when repeating the procedure, TargetCluster is only correct 224 times out of 1,000. See **Table 10** for the corresponding stability analysis.

Table 10: Stability analysis of the three competitor methods on independent source (Usoskin *et al.*[82]) and target (Hockley *et al.*[286]) datasets. For each method, we present the number of times a specific cell type is identified correctly out of 1,000 replications. In each field of the table, the first number corresponds to applying the method to the original datasets with no additional preprocessing, the second number is the result of applying Seurat batch effect removal[297] before the analysis and the third number represents results on datasets that have been imputed with MAGIC[315].

	TargetCluster	ConcatenateCluster	TransferCluster
mNP / mNFa cluster separation counts	224 | 230 | 479	506 | 998 | 902	352 | 605 | 919
pPEP cluster separation counts	984 | 1,000 | 921	4 | 33 | 431	887 | 1,000 | 944
pNF cluster separation counts	999 | 1,000 | 801	481 | 962 | 579	1,000 | 1,000 | 831

The use of ConcatenateCluster on both the Hockley and Usoskin datasets improves the stability of clustering these two groups as separate clusters (506/1,000, *e.g.* in **Figure 34 D**, clusters 1 and 7); however, this comes at the expense of clustering accuracy within the remaining neurons. For example, in **Figure 34 D**, ConcatenateCluster identifies a more simplistic cluster structure with 4 clusters and no longer distinguishes separations between spinal segmental regions (*e.g.* thoracolumbar and lumbosacral) from which neuronal subtypes have been collected. As such, the concatenation of target and source, in this instance at least, may miss biologically relevant clusters. Specifically, what the original authors suggest as a putative novel peptidergic subtype (pPEP) unique to the lumbosacral DRG with high expression of neurotrimin (*Ntm*; **Figure 34 E**), tyrosine hydroxylase (*Th*) and calcitonin polypeptide alpha (*Calca*) and a second group of lumbosacral neurons, the pNF subtype, which is thought to represent a low-threshold mechanoreceptor group within the colorectum with selective expression of secreted phosphoprotein 1 (*Spp1*; **Figure 34 F**) and the mechanotransducer *Piezo2*, is missed using ConcatenateCluster.

When knowledge from the larger Usoskin dataset is instead transferred using TransferCluster, not only is the clustering accuracy of the overall data retained (identifying seven well-defined clusters), but the probability of separating the clusters mNP and mNFa is partially increased (for TransferCluster with level 3 labels, 352/1,000; **Figure 34 G**). Unlike ConcatenateCluster, TransferCluster correctly identifies not only mNP and mNFa clusters (as highlighted by the black dashed boxes around clusters 1 and 7 in **Figure 34 G**) but also spinal region dependent clusters pPEP (green dashed box, cluster 4, **Figure 34 G**) and pNF (blue dashed box, cluster 2, **Figure 34 G**). In order to quantify these effects, we measure how frequently TransferCluster

separated cluster 2 (*e.g.* pNF) from cluster 6 (1,000/1,000) compared to ConcatenateCluster (481/1,000) and likewise, how frequently cluster 4 is separated from cluster 3 (887/1,000) compared to ConcatenateCluster (4/1,000).

See **Appendix Chapter III.** for a detailed stability analysis using different source labels (*i.e.* generated via NMF clustering or using level 1 and 2 labels of the original publication) as *a priori* knowledge in TransferCluster.

In additional experiments, we apply an established batch effect removal preprocessing step[297] to combine the Usoskin and Hockley datasets, which are then separated and our three clustering methods applied as described above. Batch effect removal improves the performances of both ConcatenateCluster and TransferCluster; however, transfer learning still outperforms simultaneous clustering on the combined dataset. For example, ConcatenateCluster fails to reliably identify pPEP cells (33/1,000), whilst TransferCluster following batch effect preprocessing finds all three cell types of interest in the majority of cases (mNP/mNFa split: 605, pPEP: 1,000 and pNF: 1,000, **Table 10**).

Table 10 also shows the results of applying a widely used imputation method[315] to the original datasets before applying the three clustering methods. It can be seen that imputation improves the performances of all methods on (almost) all clusters, but transfer learning still outperforms clustering on the target dataset alone and simultaneous clustering on the combined dataset in some areas. Specifically, for identifying mNP and mNFa clusters, transfer learning improves the results and yields almost twice as many correct results as TargetCluster (919/1,000 vs. 479/1,000). TransferCluster is still the only method that identifies all three clusters in the majority of cases (919/1,000, 944/1,000 and 831/1,000 for the three clusters of interest). In comparison, TargetCluster does not perform as well when looking at the mNP/mNFa clusters (479/1,000) and ConcatenateCluster does not do as well considering the pPEP and the pNF clusters (431/1,000 and 579/1,000). Please note that imputation through MAGIC[315] greatly increases the overlap in genes between the two datasets after gene filtering from 4.402 to 20.125 common genes. The larger common feature space provides an explanation for the positive effect of MAGIC on the performance of clustering after concatenation or transfer learning. However, ConcatenateCluster - which also profits from the increased number of common genes - does not perform as well as TransferCluster (looking at the pPEP/pNF clusters). Hence, knowledge transfer is necessary and improves clustering regardless of whether MAGIC is used or not.

As an additional analysis, we now present an investigation of the internal procedure for choosing a mixture parameter θ for the knowledge transfer from the Hockley to the Usoskin dataset. As described in **Chapter 4.2.2**, it is selected based on KTA scores, which are calculated for assessing the similarity between the linear kernel of the mixed dataset and the linear kernel of the cell type labels predicted by subsequent SC3

clustering. Since this similarity score can be interpreted as a measure for the quality of the clustering result, we select the value of θ that results in the highest KTA score. In **Figure 35**, we present the KTA scores of the Hockley dataset with transfer from Usoskin for a range of the mixture parameter θ between 0 (meaning no mixture, *i.e.* no transfer learning) and 1 (meaning full mixture) and note that high values of θ are to be avoided in this case and lower values of θ should be preferred. The maximal KTA score is obtained for θ = 0.7, which is the value that is consequently selected by the automatic procedure. These findings indicate that the proposed transfer learning method is able to identify relatedness but also differences in the two datasets by automatically choosing a mixture parameter that lies in the middle of the range of possible values of θ. This is in accordance with the fact that the source and target datasets are completely independent but biologically related, datasets collected at different times and places, which are expected to share some cell types but not all.

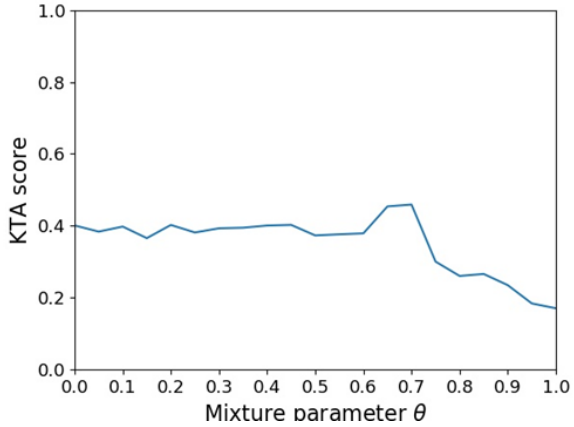

Figure 35: Investigation of the mixture parameter selection process of the transfer learning method for independent source (Usoskin *et al.*[82]) and target (Hockley *et al.*[286]) datasets. Influence of the mixture parameter θ on the unsupervised performance measure - the KTA score - for the Hockley target dataset and the Usoskin source dataset. The automatic selection process chooses the argument of the maximum of this curve, which is 0.7, as the mixture parameter to use for the final clustering analysis.

To summarize, we show that TransferCluster is able to consistently improve the reliability of clustering small datasets through the transfer of knowledge from larger, biologically relevant, yet independent datasets. The proposed method automatically estimates the level of similarity or transferability between two datasets and adjusts the corresponding mixture parameter accordingly. The method is improved by and amenable to existing preprocessing approaches.

4.4 Summary and discussion

To address challenges in the field of clustering scRNA-Seq datasets, a number of methods have been presented in the literature to make use of datasets from different studies, laboratories or points in time. These approaches can be classified into two groups:

1. Multitask learning approaches that solve clustering problems of multiple datasets simultaneously while correcting for batch effects[292,293,295–304] and
2. Transfer learning approaches that use large reference datasets to improve the clustering of target datasets that are often smaller in sample size[308–313].

The main point of interest of this dissertation laid in transferring knowledge without having to combine datasets and thus, our focus was on methods that fall into the second category. Rather than limiting a clustering method to a reference set of cell types[308–311], we aimed to enable the annotation of new target clusters. This left us with only one method, called SAVER-X[312], that is most closely related to the present research in aiming to adjust a target dataset with information from a source dataset. By training a deep autoencoder on the target dataset and initializing it with weights obtained from training on the source dataset SAVER-X achieves denoising of the target dataset. Denoising, however, was not the only goal of our method, which can additionally be used to induce certain specific properties of the source dataset into the target dataset by making use of pre-existing source labels. In contrast to our method, SAVER-X depends on large sample sizes and is also not convex. Another relevant deep learning-based approach[313] focuses on improving the clustering of a target dataset with the help of a source dataset that does not share any cell types with the target dataset. The method is not comparable to our transfer learning approach because we concentrated on problems where source and target data share a significant number of cell types.

For the aforementioned reasons and to our knowledge, this work presented a novel approach to a unique problem setting that had so far not been addressed in previous literature.

To summarize, we proposed a novel and powerful method for transferring knowledge from a well-annotated source dataset to a target dataset of a smaller sample size for which new cluster annotations are desired. Source clustering labels can be incorporated as part of this knowledge when available but are not required. The knowledge transfer procedure is based on the application of an NMF step on the source dataset before transferring the learned knowledge to the target dataset by reconstruction of a new target dataset. Finally, this modified target dataset can potentially be provided to any clustering algorithm. We have shown here that it can be successfully applied to SC3 clustering and improved the results of SC3 consistently for a range of different settings.

Specifically, transferring knowledge from a large well-annotated source dataset to a smaller target dataset was not only more successful than applying SC3 to the original target set alone but also to a simple concatenation of the source and target. This was found to be true in both simulated and real-world environments where source and target were either sampled from identical distributions of cells or only shared a subset of cell clusters. In real-world applications, the method is thus especially helpful when the overlap between source and target data is not perfect and concatenation of the two datasets is not a good option. The method was shown to perform well regardless of whether reliable clustering labels of the source data are available or not. The performance of the proposed method was further improved by applying appropriate preprocessing batch effect removal or imputation before clustering.

In regards to the thesis investigated in this dissertation, the proposed method represents a combinatorial approach of a widely used clustering algorithm - SC3 - and a novel transfer learning technique - based on NMF - for the analysis of scRNA-Seq datasets. In accordance with the proposed thesis, it was successfully applied to better understand the translation of genetic code into phenotypes and biological function in the context of cell types and was shown to increase the clustering accuracy of existing techniques. In addition, it was shown that only the combination of the individual components of the approach can achieve the desired increase in clustering accuracy and that - in this context too - "*the whole is greater than the sum of its parts*"[26] (derived from Aristotle, 4th century BC).

The proposed approach can be extended to explore a number of different research directions since it is relatively easy to apply, modify and adjust. Other downstream analysis methods instead of the SC3 clustering methods or even instead of clustering, in general, could be used. As mentioned before, it is also possible to use the proposed method to transfer knowledge from small to large datasets. Additionally, the transfer learning approach can be applied to other areas of scientific research in biological and medical fields.

Potential future research directions making adjustments to the method itself might include the incorporation of different source and target feature spaces. If X_{src} and X_{trg} only share a small set of transcripts, a loss of (probably) vital information is inevitable since only the set of genes present in both datasets can be used. Most of the multitask learning methods listed above only use the intersection of genes of all datasets when combining the datasets. Future work should thus focus on making adjustments to the method that allow the inclusion of different sets of source and target genes. One important technical point here is that scRNA-Seq experiments often make a trade-off between high cell numbers and high gene numbers. While technologies like 10X [275] enable high cell numbers but low gene coverage, other tools like SMARTSeq2 [319] use low cell numbers but generate high gene coverage. Ideally, one would like to use a 10X dataset (or similar) to aid the clustering of a SMARTSeq2 dataset (or similar) but

somehow retain the detailed gene information. In this kind of setting, where the target dataset has substantially more genes than the source dataset, a simple modification of our method is straightforward: While the transfer learning procedure can be applied without changes to the genes in both source and target, all other genes in the target dataset can be left constant. A more sophisticated way to modify the method accordingly would be to make use of a learned covariance matrix over the target genes to adjust those genes that are in the target but not in the source. The same procedure can be applied in a setting where there are source genes that are not part of the target dataset.

Finally, we would like to address the limiting factors of the proposed transfer learning method. As we have discussed in **Chapter 3.4**, ML approaches always depend on the amount and quality of the available training data. Here, the performance of the proposed transfer learning technique specifically relies on a high-quality source dataset. As we have shown earlier, our approach can be combined with quality-improving techniques (such as imputation or batch effect removal), but future adoptions of the method could potentially focus on the robustness of the method and consider low-quality cell readings. In this respect, another potential limitation of the transfer learning approach lies in the fact that it depends on the existence of meaningful information in the source dataset that is transferable to the target dataset. To minimize the effect of analyzing mismatched datasets with our method, we introduced the procedure that automatically adjusts the level of transfer via the mixture parameter. However, this cannot guarantee that the method is misused in exceptional cases.

Regarding the usefulness of our method for future scRNA-Seq studies, we need to mention the increasing dataset sizes caused by the fast-moving technological advances in the field. As our approach addresses situations where the target dataset is small, one might think it could become obsolete once more powerful and cost-effective technologies are available. However, some rare tissues and cell types will always be hard to collect from living organisms (especially humans) or even inaccessible. Transfer learning approaches will be helpful for the analysis of such datasets and can also improve the performance on large datasets. Then and when applying the transfer learning method in scenarios other than scRNA-Seq, scalability might become an issue, but several approaches to factorizing large-scale datasets have been presented in the literature[320–322]. Methods for updating NMFs online when new data is available without recalculating from scratch have also been proposed[323]. In this dissertation, we focussed on situations where learning a target task is the main point of interest, but in other scenarios, the goal might be learning two or more related tasks simultaneously. Our approach could be adopted to jointly factorizing multiple interrelated datasets [324,325].

Considering the stability of our method, it is important to note that its non-deterministic characteristics are caused by the SC3 clustering algorithm and the algorithmic search for global minima of the factorizations. In several stability experiments, we have shown

that the findings of our method are consistent when reapplied to the datasets under investigation. However, they still heavily depend on the hyperparameters and the initializations of the learning variables. Throughout our studies we have noticed a high sensitivity to these factors, which is another reason why an automatic selection of the mixture parameter was developed.

Finally, considering the explainability of our transfer learning approach, it is crucial to note that the NMF acts as a black box. So far, it is not intuitively possible to understand what information was transferred from the source to the target dataset and why. The concepts of XAI, as described in **Chapter 2**, should be utilized to increase interpretability and build trust by showing that the knowledge transfer of our method is actually meaningful.

5 Conclusion

The aim of this dissertation was to validate the proposed thesis that combinatorial approaches of traditional methods and novel ML ideas for the analysis of large-scale biological datasets can be developed and successfully put into practice to better understand the translation of genetic code into phenotypes and biological function increasing the statistical power and accuracy of existing techniques.

In this context, the first contribution consisted in proposing to employ the full potential of AI methods to improve and expand on the traditional analysis of GWAS, which is commonly based on multiple testing methods and fails to explain large fractions of heredity for complex phenotypes. To overcome the major drawbacks of these conventional testing procedures, the idea was to combine them with ML algorithms that - in contrast to RPVT - base their prediction not only on the information at a specific SNP but take the entire dataset of all SNPs, including correlation structures, into account. The positions in the genome that had the largest effect on the classifier's decision were then assumed to be good candidates for RPVT. Hence, the first proposed method - called COMBI - trains an SVM[11–13] for phenotype prediction and selects - based on the learned weights of the SVM - a set of candidate SNPs for multiple hypothesis testing. It was shown that COMBI outperforms the traditional RPVT statistical testing approach on generated data as well as on seven real-world GWAS datasets for which validation was achieved via replicability in external studies. It was also found that the individual components of the proposed combinatorial approach cannot achieve comparable performance when applied separately. Neither the SVM weights nor the raw *p*-values as direct test statistics can achieve the same accuracy as the COMBI *p*-values, not even when they are processed through the moving average filter that is a crucial part of the COMBI method. In agreement with the proposed thesis of this dissertation, this proved that only the combination of these components of the algorithm could successfully improve the identification of SNPs that are associated with the phenotype under investigation. Several additional competitor methods were investigated and COMBI was shown to outperform the statistical power and accuracy of existing techniques. Traditional approaches for the analysis of GWAS either concentrate on phenotype prediction[57–59] or aim at the identification of genotype-phenotype associations[64–66]. Typically, these methods are of purely statistical nature[42,43], but recently ML has also emerged as an important tool for investigating genomic data. The most important ML publications in the context of the proposed COMBI method focus on the ML-based identification of phenotype-associated SNPs[62,63,66,213,243–246]. The COMBI method was shown to outperform a set of representative competitor approaches, including other combinatorial ML approaches (Roshan *et al.*[62], Meinshausen *et al.*[213] and Wasserman and Roeder[210]) and two purely statistical analysis

tools (Lippert *et al.*[188,192]). While Wasserman and Roeder[210] lose a great amount of statistical power by splitting the datasets into two parts for separate SNP preselection and statistical testing, the LMMs proposed by Lippert *et al.*[188,192] still test genetic locations and pairs thereof individually instead of simultaneously.

To further elaborate on the proposed thesis of this dissertation and reacting to the development of high-performing DNNs in many fields of data science, we proposed a deep learning-based extension of COMBI, called DeepCOMBI. Deep learning and XAI for the analysis of large-scale biological datasets can improve our understanding of genetic code and biological function by exploring the genetic architectures of phenotypes in GWAS. The novel algorithm replaces the rather simple prediction tool of the COMBI method (*i.e.* the linear SVM) with a more sophisticated deep learning method by training a DNN to classify subjects into their respective phenotypes. Subsequently, it makes use of the concept of explainability to uncover the decision-making process of DNNs. It explains the classifier's decisions by applying LRP[16–18]. The method eventually utilizes the LRP relevance scores to determine the subset of most relevant features identifying the positions in the genome that are statistically associated with the trait under investigation. The statistical testing step is adopted unchanged from the original COMBI method. We found that the proposed deep learning techniques and explanation methods improved the performance of the combinatorial approach even further. DeepCOMBI was shown to perform better than the original COMBI method on both generated controlled datasets as well as on real GWAS datasets. The new method was also compared favorably to the baseline methods of RPVT and raw LRP relevance scores. In agreement with the proposed thesis of the dissertation, it was once again shown that the combination, not the individual components of the method, showed the highest performance increase. As a similar deep learning-based approach, we discussed the method proposed by Romagnoni *et al.*[193], who were the first to use XAI in the context of GWAS and apply PFI as an explanation method on a GWAS dataset. However, the corresponding NN could not outperform traditional ML-based tools in terms of prediction accuracy. The model-agnostic approach of PFI that left space for improved performances of more sophisticated explanation methods - like LRP - specifically tailored to NNs.

After having validated the proposed thesis of this dissertation in the field of GWAS and making two methodological contributions to the identification of important positions in the genome, the goal was now to elaborate on the thesis in another field of biological research. Beyond the identification of associations between phenotypes and genotypes, we aimed to contribute to a better understanding of how genetic information is translated into physical structures and cell function. When investigating the transformation of genetic information from DNA to mRNA to proteins, the genome is often interpreted in the context of cell types by examining which genes are active in certain cells. To this end, the technique of quantifying all mRNA molecules in single cells, called scRNA-Seq, has given rise to many important insights by clustering cells

according to their transcriptome. Challenges remain when high experimental efforts lead to small but high-dimensional datasets and both multitask learning approaches[292–304] and transfer learning approaches[308–313] have been proposed in this context. To overcome the issues of small datasets and in accordance with the thesis of this dissertation, we proposed to combine the ML concept of transfer learning with a widely used algorithm for clustering scRNA-Seq data. The novel method allows to utilize *a priori* knowledge from large, well-annotated reference datasets, including the corresponding clustering labels when available. Through the use of NMF[23–25] of the source dataset to reconstruct a modified version of the target dataset, the clustering result of a traditional downstream clustering algorithm was improved. We showed that we can achieve higher performance rates when clustering the modified dataset instead of the original target dataset or the concatenation of source and target dataset. This was investigated in a number of different settings, including simulated and real-world environments. Once again, it was shown that a combinatorial approach of existing tools and novel ML ideas improved the analysis of large-scale biological datasets. Most competitor transfer learning methods limit a clustering method to a reference set of cell types[308–311] and only a few allow the annotation of new target clusters. While one deep learning-based method, called SAVER-X[312], focuses on denoising the target dataset, another approach[313] aims to improve the clustering of a target dataset with the help of a source dataset that does not share any cell types with the target dataset. These methodological goals might be closely related but are not equal to the goal of the method proposed here, which aims to induce specific properties of the source dataset into the target dataset, which share a significant number of cell types.

To summarize and conclude, this dissertation contributes to one of the most important challenges in biology - understanding the genome and the processes around its translation into biological structures and functions. By proposing three combinatorial approaches for the analysis of large-scale biological datasets, each consisting of traditional methods on the one hand and state-of-the-art ML algorithms on the other, we showed that, in this context too, "*the whole is greater than the sum of its parts*"[26] (derived from Aristotle, 4th century BC). The two GWAS-related methods can determine locations on a genetic sequence which affect the probability of having a specific disease or trait and the scRN-Seq-related method enhances the grouping of cells into specific classes based on their transcriptomes. Hence, in the course of this dissertation and in full support of the proposed thesis, it was shown that combinatorial approaches of traditional methods and novel ML ideas for the analysis of large-scale biological datasets can be developed and successfully put into practice to better understand the translation of genetic code into phenotypes and biological function increasing the statistical power and accuracy of existing techniques. In future applications, the three proposed methods could be used consecutively to first identify SNPs that are associated with a phenotype and then to determine in what cells types the corresponding gene is activated to cause the phenotype to develop. Great potential lies

in the possibility of utilizing the findings of the proposed methods in practical applications, *e.g.* for predictive personal prognosis, data-driven diagnosis, the design of optimal treatment plans or the identification of potential drug targets.

To round off this dissertation, we want to mention the many possible future research options. The proposed approaches can be extended to explore numerous different directions by substituting one of the general alogrithm's substeps with other suitable procedures. Specific suggestions for future work regarding the proposed methods are described in the discussion sections of the corresponding chapters. Most of these ideas consider applying either other ML methods or different traditional data analysis methods that need performance boosting. Since deep learning and XAI have gained huge attention from the scientific community, the proposed methods could benefit from recently published approaches in this field. Besides classification, which forms the core of ML, and clustering, which is the traditional analysis in scRNA-Seq, it would be interesting to adjust and study the proposed methodologies for other learning tasks such as regression or detection. Future research could also focus on exploring combinatorial approaches in semi-supervised learning or online learning. Furthermore, the proposed combinatorial approaches can be applied to other areas of scientific research in biological and medical fields, *e.g.* clinical decision support in cancer diagnosis. Finally, future work could also aim at exploring applications in other fields where widely used traditional methods need improvement through the introduction of novel ML-based approaches.

Appendix

I. Parameter investigation on the conservativeness of the COMBI method on generated datasets

In this chapter, we discuss reasons for the conservativeness of the COMBI method when the permutation-based thresholding procedure (see **Algorithm 2** in **Chapter 3.2.2** in the main text) is applied. We investigate this phenomenon on the generated datasets from Mieth *et al.* 2016[10]. Please note that during this initial phase of the dissertation, a slightly different experimental setup for the generated datasets was employed and DeepCOMBI was not introduced yet. For example, 10,000 datasets were generated here instead of 1,000 in the DeepCOMBI publication[15], which - without loss of generality - causes the performance curves here and in the main text not to be identical. As a starting point, we present the ROC performance curves and the corresponding permutation-based thresholding result of the COMBI method and its competitor, RPVT, on the 10,000 datasets from 2016 in **Figure A.1**.

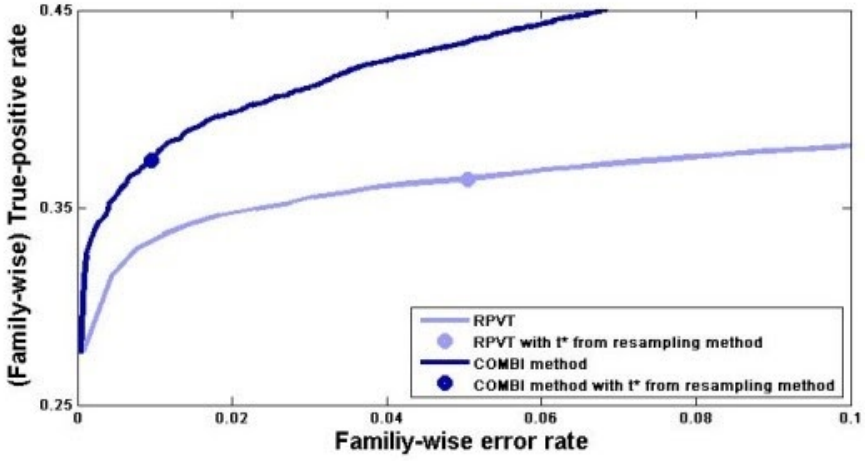

Figure A.1: Main results of both COMBI and RPVT methods applied to the semi-real generated datasets with controlled phenotypes: *TPR* averaged over 10,000 generated datasets as a function of the *FWER*. The dots mark measurements of the permutation-based calibration, where the corresponding thresholds are calibrated to guarantee a *FWER* of $\alpha \leq 0.05$.

It can be seen that in the controlled environment of generated datasets, the permutation-based threshold calibration yields a rather conservative error rate. The COMBI method does not exploit the full significance level but makes fewer errors than anticipated. Instead of the desired error rate of $\alpha \leq 0.05$, a *FWER* of only around 1% is achieved. Even though it is desirable to increase power by simultaneously making more mistakes, *i.e.* as many as anticipated, it is important to note that the COMBI method has lower error rates and higher power than the RPVT method in combination with the

same permutation-based calibration principle. Reasons for the conservativeness of the COMBI method are now to be investigated further.

An effect that makes the COMBI method conservative is the different number of uninformative (or noise) SNPs the threshold calibration is based on and eventually applied to. To illustrate this, assume that $k = 30$, indicating that the 30 SNPs with the highest SVM weights are selected for each replication in the permutation test and *p*-values are computed. The significance threshold is then determined based on these *p*-values only. As the threshold is calibrated on random labels, it is based on the *p*-values of 30 uninformative SNPs. However, when we train on real labels when applying the threshold, it is very likely that 20 informative SNPs are selected as part of the 30 highest ranked SNPs. There are thus only 10 spots left for the noise SNPs, which are rejected only if they exceed the threshold. Having 10 instead of 30 noise SNPs makes an erroneous rejection much more unlikely. Thus, the COMBI method makes fewer mistakes than anticipated and is rather conservative. To validate this hypothesis, we perform a number of experiments where different parameter values are altered to achieve the correct *FWER* of 5%. In the first experiment, *k* is increased in a way that the number of noise SNPs remains constant. Instead of only selecting *k* SNPs (noise or informative), we select the informative SNPs that are amongst the *k* best SVM weights; in addition to that, the best *k* noise SNPs, *i.e.* $k_i^* = k + n_{TP,i}$, where $n_{TP,i}$ is the number of true positives among the *k* best SNPs in the *i*-th replication of the experiment. This should eliminate the described effect and hence yield a *FWER* closer to that set prior to the permutation test. Applying this slightly modified COMBI method to the 10,000 generated semi-real datasets leads to a *FWER* of around 5% and thus supports this hypothesis. Note that these modifications can only be applied to datasets where the ground truth and thus $n_{TP,i}$, is known. To investigate this effect in more detail, we now perform an experiment where we increase the number of selected SNPs in order to check whether this also yields a less conservative error rate. **Figure A.2** shows that increasing *k* to a maximum of 1,000 yields a *FWER* of 5%. This is reasonable, as increasing *k* means increasing the fraction of noise SNPs in the set of SNPs that are picked via the permutation-based calibration procedure. For $k = 1000$, enough SNPs are selected and we yield a non-conservative *FWER* of 5%. The fewer SNPs we select, the better the curve we achieve and the more conservative the permutation-based calibration. The optimal curve but also the most conservative threshold calibration is reached for $k = 30$, which is the parameter chosen for all other applications of the COMBI method. Even though it does not exploit the full level of error, it yields the highest power using the permutation-based calibration. The higher *k*, the less optimal are the resulting performance curves, but also the closer is the corresponding ENFR to the anticipated 5%. For $k < 30$, the *ROC* curve of the COMBI method suffers a severe loss in power.

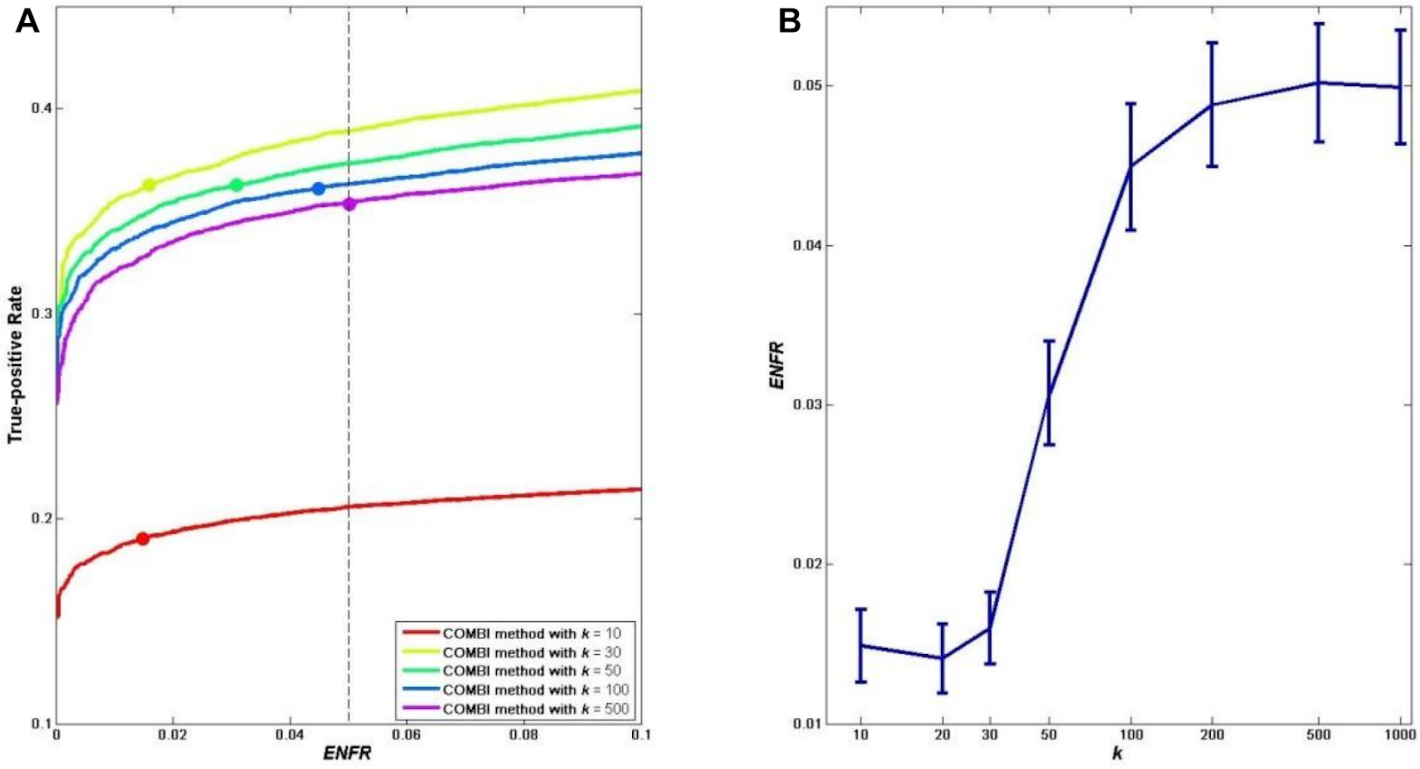

Figure A.2: Performance curves and ENFR of COMBI for different values of *k* on generated datasets from Mieth *et al.* (2016)[10]. **A** ROC curves and **B** *ENFR* of the COMBI method averaged over the 10,000 generated datasets are shown, both for increasing *k*, *i.e.* the number of SNPs to select in the screening step, from 10 to 500. The points in **A** represent the results of the COMBI method after applying the permutation-based significance threshold, which is calibrated to guarantee a *FWER* of $\alpha \leq 0.05$. Mean and standard deviation of *ENFR* are shown in **B** for different values of *k*.

The next parameter to be investigated is the effect size parameter γ, which is involved in the data generation process. We expect that for high levels of noise, *i.e.* when there is a small effect size and basically a complete lack of real association, the *FWER* of the COMBI method approaches the expected 5% because the 30 selected SNPs selected are all noise SNPs. Thus, the threshold is not only based on noise SNPs but also only applied to noise SNPs. Observe from **Figure A.3** that for a predefined $\alpha \leq 0.05$, the correct *FWER* is actually achieved for a minimal effect size of $\gamma = 0.25$, which is almost equivalent to having a maximum level of noise and therefore no informative SNPs associated with the disease. The curve is optimal for minimum noise / high effect size, where the identification of true associations is much easier, which supports the hypothesis from above.

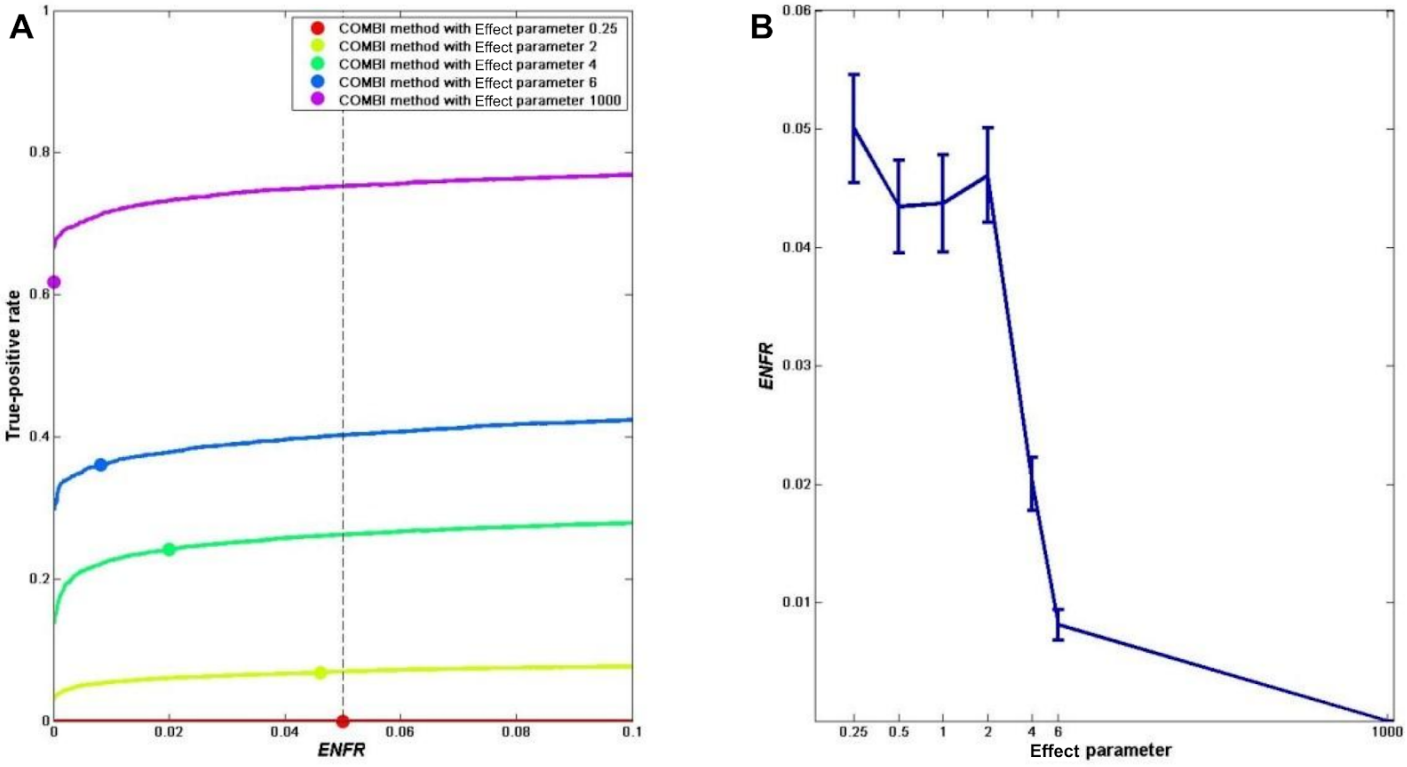

Figure A.3: Performance curves and ENFR of COMBI for different values of the effect size parameter γ on generated datasets from Mieth *et al.* (2016)[10]. **A** ROC curves and **B** *ENFR* of the COMBI method averaged over the 10,000 generated datasets are shown, both for increasing γ, *i.e.* the effect size parameter, from 0.25 to 1,000, where high values correspond to low noise and low values to high noise in the process of simulating the datasets. The points in **A** represent the results of the COMBI method after applying the permutation-based significance threshold, which is calibrated to guarantee a *FWER* of $\alpha \leq 0.05$. Mean and standard deviation of *ENFR* are shown in **B** for different values of γ.

Observe from **Figure A.4** that we achieve the error rate expected after setting $\alpha \leq 0.05$ in the permutation test only for a filter length l_{filter} of 1, which corresponds to applying no filter. The error rate decreases and power increases with increasing the filter length up to 35, where the curve is optimal. The method yields higher error rates and less power for greater filter lengths. This finding is in agreement with what Alexander and Lange[231] found. We, therefore, decide to use a filter length of 35, which yields optimal but conservative results.

We learn from these experiments that the proposed method may achieve lower error rates and higher *TPR* than anticipated.

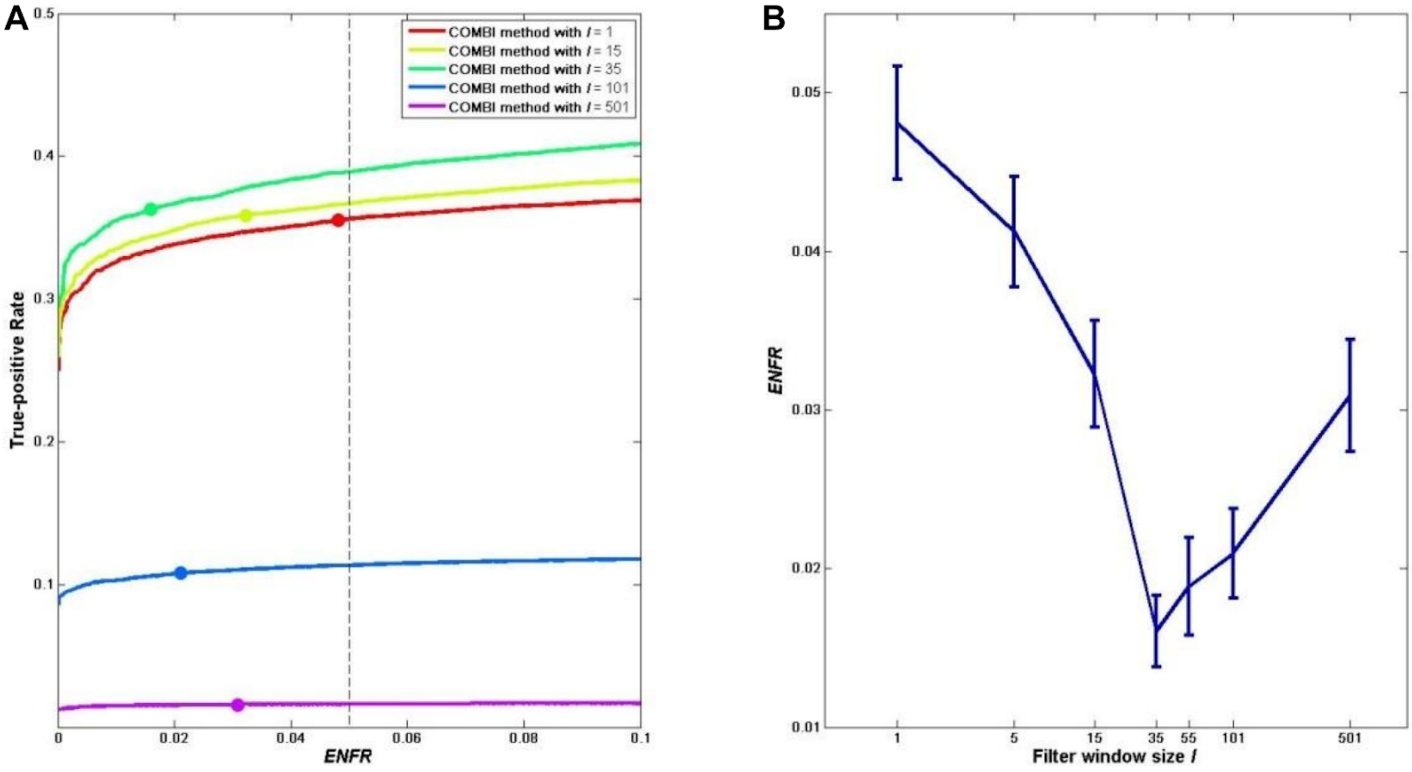

Figure A.4: **Performance curves and ENFR of COMBI for different values of the filter window size l_{filter} on generated datasets from Mieth *et al.* (2016)[10].** **A** ROC curves and **B** *ENFR* of the COMBI method averaged over the 10,000 generated datasets are shown, both for increasing l_{filter}, *i.e.* the filter window size, from 1 to 501, where the former corresponds to applying no filter at all and the latter to an extremely strong flattening filter. The points in **A** represent the results of the COMBI method after applying the permutation-based significance threshold, which is calibrated to guarantee a *FWER* of $\alpha \leq 0.05$. Mean and standard deviation of *ENFR* are shown in **B** for different values of l_{filter}.

II. Stability analysis of the COMBI method on WTCCC data

In order to establish an internal validation criterion of the COMBI method on WTCCC data, we analyze the stability of the reported associations, which indicates how well the results can be reproduced on another independent sample. Stability is a desirable property of any SNP-selection method: if a method is not stable, it could either indicate that too many locations are selected, meaning that the result contains a random subset of non-significant SNPs, or that not enough locations are selected so that the result contains only a random subset of the significant SNPs. To investigate stability, we proceed as follows: the original data is randomly split into two equally sized subsets (of individuals), *A* and *B,* ten times. The method under scrutiny, *i.e.* either the COMBI method or standard RPVT, is applied separately to the data from sets *A* and *B,* leading to sets *S(A)* and *S(B)* of reported SNPs, respectively. Using the Tanimoto Index[326] $T(S(A), S(B)) = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|}$, the similarity of these two sets and thus the stability of the used method is measured. Here $|S|$ denotes the cardinality of the set S. In this manner, the stability of the COMBI method can be compared to the stability of standard RPVT. Simulation results considering the internal stability of the two methods when applied to the WTCCC Crohn's Disease dataset are shown in **Figure A.5**.

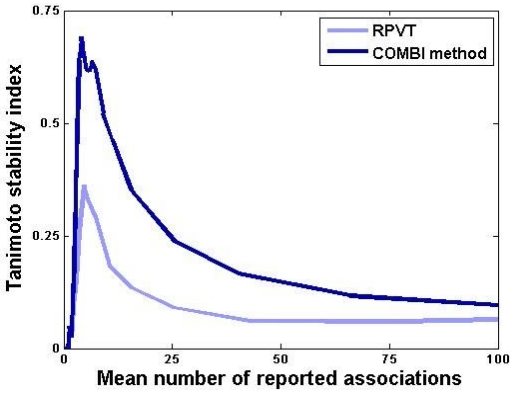

Figure A.5: Stability analysis of RPVT and COMBI on Crohn's Disease WTCCC dataset from Mieth *et al.* (2016)[10]. The averaged Tanimoto stability indices between the reported associations in two randomly selected subsets of the Crohn's disease dataset are shown for varying numbers of reported associations. Higher Tanimoto indices indicate higher stability of the method.

COMBI produces more stable results than RPVT. The Tanimoto stability index is plotted against the mean number of reported associations, *i.e.* $\frac{|S(A)+S(B)|}{2}$. When we repeatedly split the data into two parts and investigate how similar the results of the two methods are in the two subsets, we find that the results of the COMBI method are more similar and thus more stable than RPVT independently of the mean number of reported associations (varied via the significance threshold) for all levels of *ENFR*. This result holds true for all seven diseases and is robust with respect to the parameter *k* (number of SNPs selected in the screening step) (see **Figure A.6**).

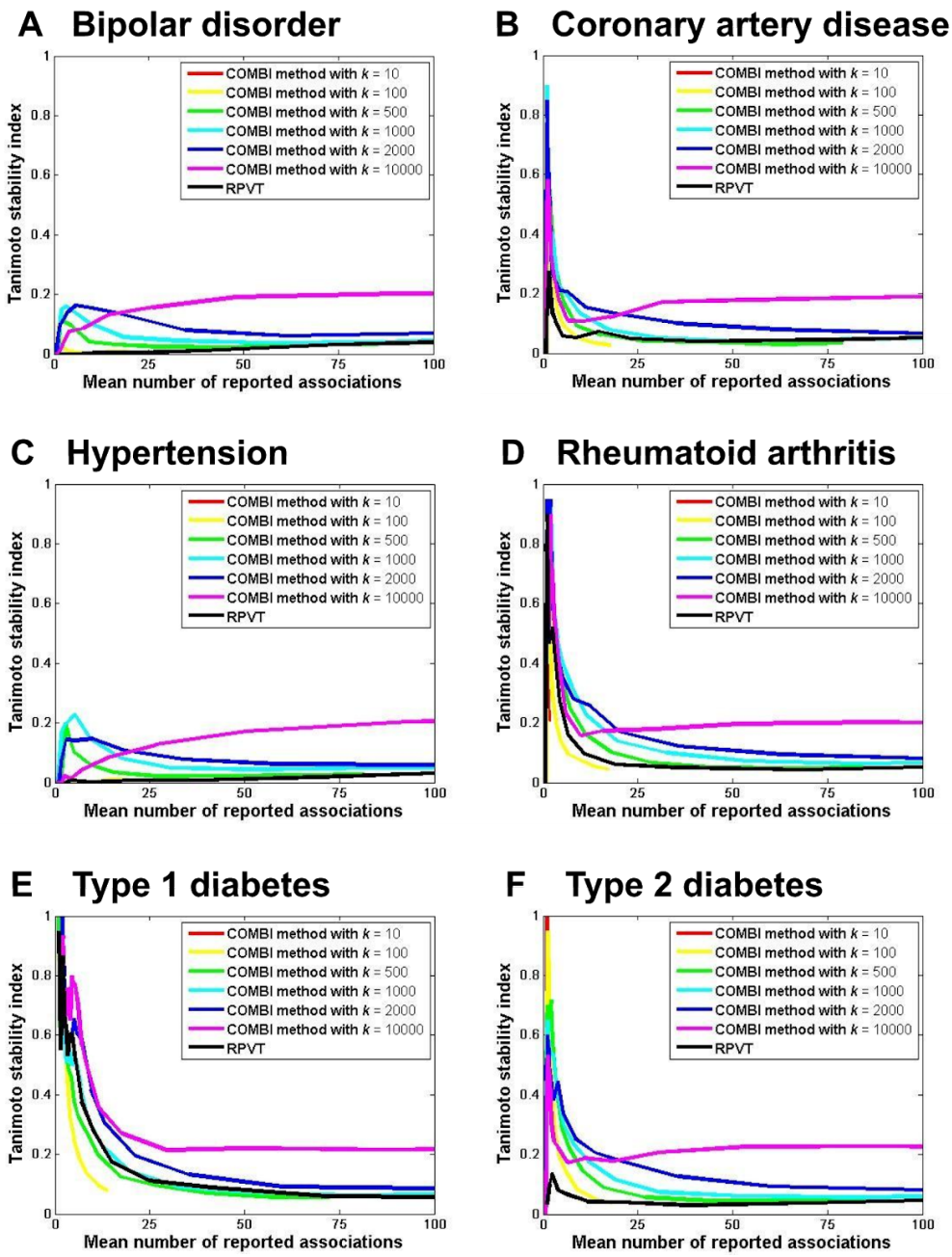

Figure A.6: Stability analysis of RPVT and COMBI for different values of *k* on six WTCCC datasets from Mieth *et al.* (2016)[10]. The averaged Tanimoto stability indices between the reported associations in two randomly selected subsets of the Crohn's disease dataset are shown for varying numbers of reported associations. Higher Tanimoto indices indicate higher stability of the method.

III. Extended clustering analysis on independent source and target datasets using various source data labels as *a priori* knowledge

In **Chapter 4.3.3** of the main text of this dissertation we investigate the performance of the proposed transfer learning method using level 3 labels from the original Usoskin *et al.* publication[82] as *a priori* knowledge about the source dataset. Here, we present additional analyses using different source cluster labels. As with the Tasic dataset[291], we first analyze the Hockley dataset[286] pretending no reliable source labels for the Usoskin dataset are available and generate them via NMF clustering. We assume a complete overlap between the cell types in source and target data and choose the number of clusters to be $k = 7$ for the source label generation. Afterwards, we use the source labels from the data-driven clustering of the original Usoskin *et al.* publication[82]. They provided labels in the form of a hierarchical clustering, which was cut off at three different levels resulting in three different sets of source labels with different numbers of clusters (4,8 and 11 cell types). Here, in addition to the results in the main text on level 3 labels, we present the results for NMF labels, level 1 and level 2 labels.

Figure A.7 shows the clustering results of all competitor methods on the Hockley dataset. TargetCluster uses only data from Hockley to assign clusters and ConcatenateCluster uses a concatenation of data from Hockley and Usoskin to assign clusters. TransferCluster uses the novel transfer learning approach with Hockley as target and Usoskin as source with four different sets of corresponding labels.

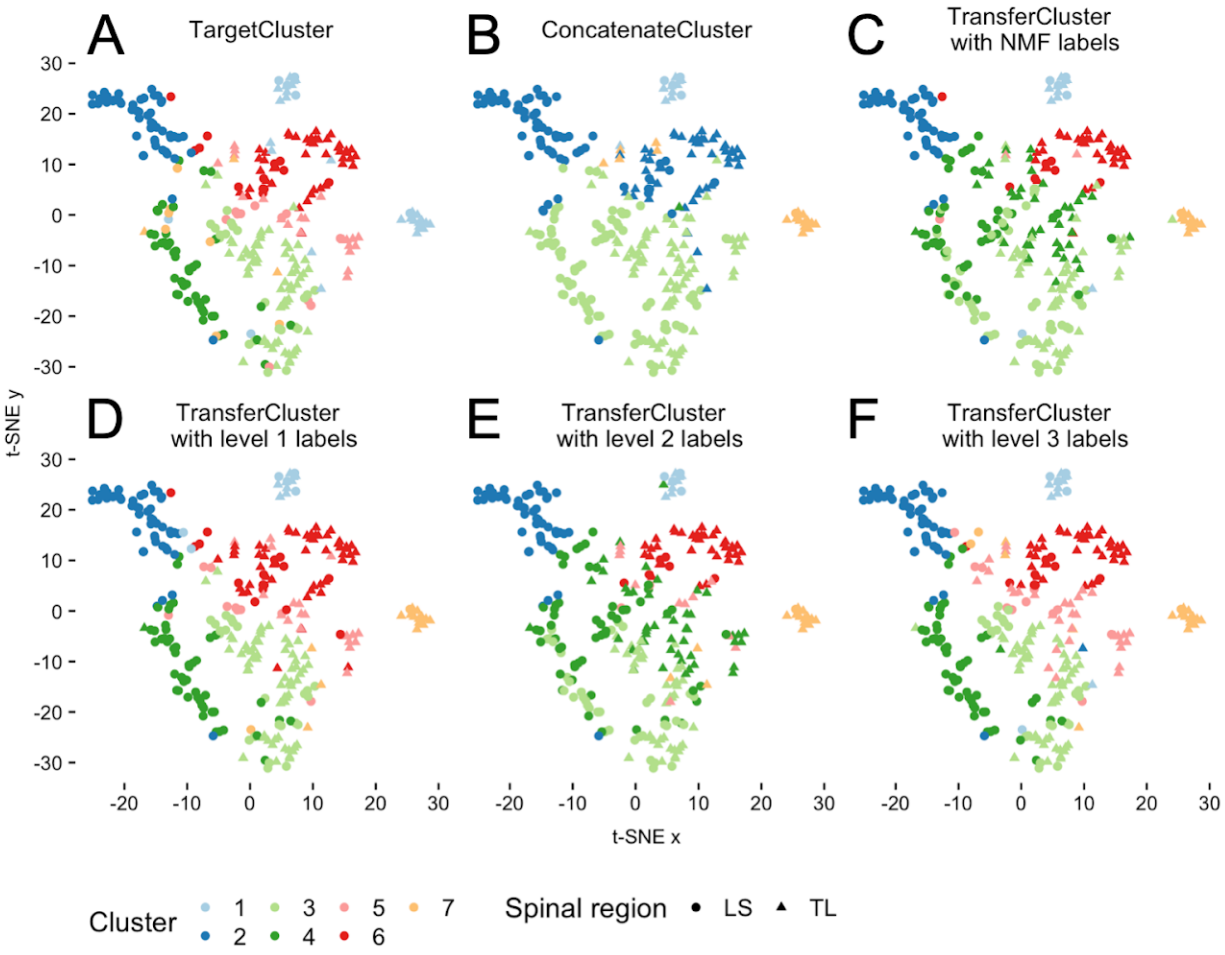

Figure A.7: Clustering results of independent source (Usoskin *et al.*[82]) and target (Hockley *et al.*[286]) datasets using different clustering memberships of the source dataset. t-SNE plots of the mouse colonic sensory neurons from the Hockley dataset are shown. **A** TargetCluster, using only data from Hockley *et al.* to assign clusters. **B** ConcatenateCluster, using a concatenation of data from Hockley *et al.* and Usoskin *et al.* (mouse sensory neurons) to assign clusters. **C** TransferCluster, using the novel transfer learning approach with Usoskin *et al.* as source and Hockley *et al.* as target with NMF labels for the source dataset. **D** TransferCluster with level 1 labels for the source dataset, **E** TransferCluster with level 2 labels for the source dataset. **F** TransferCluster with level 3 labels for the source dataset. Colors refer to the clusters derived from the different approaches. Shapes refer to the spinal segment from which the neuron was isolated (triangle, TL (thoracolumbar); circle, LS (lumbosacral)).

Since SC3 - the clustering method used for all approaches investigated in this dissertation (TargetCluster, ConcatenateCluster and TransferCluster) - is not deterministic and produces different results when solving the same clustering problem multiple times. We count the number of times some specific clusters of interest are separated correctly from each other by the three methods when repeating the procedure 1,000 times. Three pairs of clusters are identified to be of interest and **Table A.1** shows the number of times each of these pairs of cell types is separated correctly. Two biologically distinct groups of cells, named mNP and mNFa cells (**Main Figure 32 G** Cluster 1 and 7), are only separated 224 times when applying SC3 on the target dataset alone. Taking source information via the proposed transfer learning method

TransferCluster with NMF or level 1, 2 and 3 labels into account consistently increases this number (to 469, 300, 313 and 352, respectively). Concatenating source and target datasets and applying SC3 to the complete dataset (ConcatenateCluster) increases the number of times mNP and mNFa cells are correctly separated even further to 506. However, this comes with a loss of performance when looking at the other two pairs of cell types that are only poorly separated with ConcatenateCluster. pNF cells (**Main Figure 32 G** Cluster 2 vs. 6) are only separated 481 times and the pPEP cells (**Main Figure 32 G** Cluster 4 vs. 3) only 4 times. In contrast, TransferCluster is able to almost perfectly separate pNF clusters independent of what labels are used for the source dataset (999; 1,000; 1,000; 1,000 for NMF, level 1, 2 and 3 labels, respectively) and also has very high separation rates for the pPEP cell types (984, 703, 706 and 887 for NMF, level 1, 2 and 3 labels, respectively).

Table A1: Stability analysis of the three competitor methods on independent source (Usoskin *et al.*[82]) and target (Hockley *et al.*[286]) datasets using different clustering memberships of the source dataset. For each method, we present the number of times a specific cell type is identified correctly out of 1,000 replications. In addition to the numbers of TargetCluster and ConcatenateCluster, the numbers for TransferCluster with NMF labels for the source dataset, with level 1 labels for the source dataset, with level 2 labels for the source dataset and with level 3 labels for the source dataset are shown.

	mNP/mNFa cluster separation counts	**pNF cluster separation counts**	**pPEP cluster separation counts**
TargetCluster	224	999	984
ConcatenateCluster	506	481	4
TransferCluster with NMF labels	469	999	984
TransferCluster with level 1 labels	300	1,000	703
TransferCluster with level 2 labels	313	1,000	706
TransferCluster with level 3 labels	352	1,000	887

Author's publications and contributions

- Laszlo David, Sayed-Amir Marashi, Abdelhalim Larhlimi, **Bettina Mieth** & Alexander Bockmayr. FFCA: a feasibility-based method for flux coupling analysis of metabolic networks. *BMC Bioinformatics*, volume 12, article number 236 (**2011**)[327].

  Author contributions: L.D., S.-A.M., A.L. and A.B. designed and directed research; B.M. performed research and analyzed data; and L.D., S.-A.M. and A.L. wrote the paper.

- **Bettina Mieth**, Marius Kloft, Juan Antonio Rodríguez, Sören Sonnenburg, Robin Vobruba, Carlos Morcillo-Suárez, Xavier Farré, Urko M. Marigorta, Ernst Fehr, Thorsten Dickhaus, Gilles Blanchard, Daniel Schunk, Arcadi Navarro & Klaus-Robert Müller. Combining Multiple Hypothesis Testing with Machine Learning Increases the Statistical Power of Genome-wide Association Studies. *Scientific Reports*, volume 6, article number: 36671 (2016)[10].

  Author contributions: E.F., T.D., G.B., D.S., A.N. and K.-R.M. designed and directed research; B.M., M.K., J.A.R., S.S., R.V., C.M.-S., X.F., U.M.M. and D.S. performed research and analyzed data; and B.M., M.K., J.A.R., C.M.-S., E.F., T.D., G.B., D.S., A.N. and K.-R.M. wrote the paper.

- **Bettina Mieth**, James R.F. Hockley, Nico Görnitz, Marina M.-C. Höhne, Klaus-Robert Müller, Alex Gutteridge & Daniel Ziemek. Using transfer learning from prior reference knowledge to improve the clustering of single-cell RNA-Seq data. *Scientific Reports*, volume 9, article number 920353 (2019)[22].

  Author contributions: K.-R.M., A.G. and D.Z. designed and directed research; B.M., J.R.F.H., N.G. and M.M.C.V. performed research and analyzed data; and B.M., J.R.F.H., N.G., K.-R.M., A.G. and D.Z. wrote the paper.

- **Bettina Mieth**, Alexandre Rozier, Juan Antonio Rodriguez, Marina M.-C. Höhne, Nico Görnitz & Klaus-Robert Müller. DeepCOMBI: Explainable artificial intelligence for the analysis and discovery in genome-wide association studies. Currently under review at *NAR Genomics and Bioinformatics*. bioRxiv. doi.org/10.1101/2020.11.06.371542 (2020)[15].

  Author contributions: B.M., N.G. and K.-R.M. designed and directed research; B.M., A.R. and J.R.F.H. performed research and analyzed data; and B.M., J.R.F.H., M.M-C.H. and K.-R.M. wrote the paper.

Bibliography

1. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. & Walter, P. *Molecular biology of the cell. 5th ed.* (Garland Science, 2008).

2. Shendure, J. *et al.* DNA sequencing at 40: past, present and future. *Nature* **550**, 345–353 (2017).

3. Collins, F. S., Morgan, M. & Patrinos, A. The Human Genome Project: lessons from large-scale biology. *Science* **300**, 286–290 (2003).

4. Jasny, B. R. & Roberts, L. Building on the DNA revolution: introduction. *Science* **300**, 277–278 (2003).

5. Wetterstrand, K. A. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP). Available at www.genome.gov/sequencingcostsdata. Accessed April 2 2021.

6. Nature, Nature Genetics & Nature Reviews Genetics. Milestones in Genomic Sequencing. Available at https://www.nature.com/articles/d42859-020-00099-0. Accessed April 2 2021.

7. Buniello, A. *et al.* The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Res.* **47**, D1005–D1012 (2019).

8. Lee, S. H., Wray, N. R., Goddard, M. E. & Visscher, P. M. Estimating missing heritability for disease from genome-wide association studies. *Am. J. Hum. Genet.* **88**, 294–305 (2011).

9. Zuk, O., Hechter, E., Sunyaev, S. R. & Lander, E. S. The mystery of missing heritability: Genetic interactions create phantom heritability. *Proc. Natl. Acad. Sci. U. S. A.* **109**, 1193–1198 (2012).

10. Mieth, B. *et al.* Combining Multiple Hypothesis Testing with Machine Learning Increases the Statistical Power of Genome-wide Association Studies. *Sci. Rep.* **6**, 36671 (2016).

11. Müller, K. R., Mika, S., Rätsch, G., Tsuda, K. & Schölkopf, B. An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Netw.* **12**, 181–201 (2001).

12. Boser, B. E., Guyon, I. M. & Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* 144–152 (Association for Computing Machinery, 1992).

13. Cortes, C. & Vapnik, V. Support-vector networks. *Machine Learning* **20**, 273–297 (1995).

14. Montavon, G., Orr, G. & Müller, K.-R. *Neural Networks: Tricks of the Trade*. (Springer, 2012).

15. Mieth, B. *et al.* DeepCOMBI: Explainable artificial intelligence for the analysis and discovery in genome-wide association studies. *bioRxiv* doi:10.1101/2020.11.06.371542 (2020).

16. Bach, S. *et al.* On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS One* **10**, e0130140 (2015).

17. Montavon, G., Lapuschkin, S., Binder, A., Samek, W. & Müller, K.-R. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognit.* **65**, 211–222 (2017).

18. Montavon, G., Binder, A., Lapuschkin, S., Samek, W. & Müller, K.-R. Layer-Wise Relevance Propagation: An Overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (eds. Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K. & Müller, K.-R.) 193–209 (Springer International Publishing, 2019).

19. Tang, F. *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods* **6**, 377–382 (2009).

20. Hwang, B., Lee, J. H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.* **50**, 1–14 (2018).

21. Kiselev, V. Y., Andrews, T. S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat. Rev. Genet.* **20**, 273–282 (2019).

22. Mieth, B. *et al.* Using transfer learning from prior reference knowledge to improve the clustering of single-cell RNA-Seq data. *Sci. Rep.* **9**, 20353 (2019).

23. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401**, 788–791 (1999).

24. Lee, D. & Seung, H. Algorithms for Non-negative Matrix Factorization. In *Advances In Neural Information Processing Systems 13* (Morgan Kaufman Publishers, 2001).

25. Paatero, P. & Tapper, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* **5**, 111–126 (1994).

26. Aristotle. Metaphysics Book Z and H. In *Clarendon Aristotle Series* (ed. David Bostock) 17:1041b (Oxford University Press, 1994).

27. 1000 Genomes Project Consortium *et al.* A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).

28. Martín-Subero, J. I. How epigenomics brings phenotype into being. *Pediatr. Endocrinol. Rev.* **9**, 506–510 (2011).

29. Rave-Harel, N. *et al.* The molecular basis of partial penetrance of splicing mutations in cystic fibrosis. *Am. J. Hum. Genet.* **60**, 87–94 (1997).

30. O'brien, W. M., Li, C. C. & Taylor, F. H. Penetrance and the distribution of sib-pair types, exemplified by taste ability and rheumatoid arthritis. *J. Chronic Dis.* **18**, 675–680 (1965).

31. Ott, J. *Analysis of Human Genetic Linkage*. (JHU Press, 1999).

32. Kong, X. *et al.* A combined linkage-physical map of the human genome. *Am. J. Hum. Genet.* **75**, 1143–1148 (2004).

33. Slatkin, M. Linkage disequilibrium - understanding the evolutionary past and mapping the medical future. *Nat. Rev. Genet.* **9**, 477–485 (2008).

34. Harms, M. J. & Thornton, J. W. Evolutionary biochemistry: revealing the historical and physical causes of protein properties. *Nat. Rev. Genet.* **14**, 559–571 (2013).

35. Society for Experimental Biology (Great Britain). *The Biological Replication of Macromolecules*. (Company of Biologists, 1958).

36. Ahlquist, P. RNA-dependent RNA polymerases, viruses, and RNA silencing. *Science* **296**, 1270–1273 (2002).

37. Yao, N. Y. & O'Donnell, M. SnapShot: The replisome. *Cell* **141**, 1088 (2010).

38. Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* **447**, 661–678 (2007).

39. Ozaki, K. *et al.* Functional SNPs in the lymphotoxin-alpha gene that are associated with susceptibility to myocardial infarction. *Nat. Genet.* **32**, 650–654 (2002).

40. Ikegawa, S. A Short History of the Genome-Wide Association Study: Where We Were and Where We Are Going. *Genomics Inform.* **10**, 220 (2012).

41. Visscher, P. M. *et al.* 10 Years of GWAS Discovery: Biology, Function, and Translation. *Am. J. Hum. Genet.* **101**, 5–22 (2017).

42. Mills, M. C. & Rahal, C. A scientometric review of genome-wide association studies. *Commun Biol* **2**, 9 (2019).

43. Visscher, P. M., Brown, M. A., McCarthy, M. I. & Yang, J. Five years of GWAS discovery. *Am. J. Hum. Genet.* **90**, 7–24 (2012).

44. Ripke, S. *et al.* Genome-wide association analysis identifies 13 new risk loci for schizophrenia. *Nat. Genet.* **45**, 1150–1159 (2013).

45. Lee, J. J. *et al.* Gene discovery and polygenic prediction from a genome-wide association study of educational attainment in 1.1 million individuals. *Nat. Genet.* **50**, 1112–1121 (2018).

46. Jansen, P. R. *et al.* Genome-wide analysis of insomnia in 1,331,010 individuals identifies new risk loci and functional pathways. *Nat. Genet.* **51**, 394–403 (2019).

47. Chen, X., Teichmann, S. A. & Meyer, K. B. From Tissues to Cell Types and Back: Single-Cell Gene Expression Analysis of Tissue Architecture. *Annu. Rev. Biomed. Data Sci.* **1**, 29–51 (2018).

48. Hashimshony, T., Wagner, F., Sher, N. & Yanai, I. CEL-Seq: single-cell RNA-Seq by multiplexed linear amplification. *Cell Rep.* **2**, 666–673 (2012).

49. Levitin, H. M., Yuan, J. & Sims, P. A. Single-Cell Transcriptomic Analysis of Tumor Heterogeneity. *Trends Cancer Res.* **4**, 264–268 (2018).

50. Neu, K. E., Tang, Q., Wilson, P. C. & Khan, A. A. Single-Cell Genomics: Approaches and Utility in Immunology. *Trends Immunol.* **38**, 140–149 (2017).

51. Zafar, H., Lin, C. & Bar-Joseph, Z. Single-cell lineage tracing by integrating CRISPR-Cas9 mutations with transcriptomic data. *Nat. Commun.* **11**, 3055 (2020).

52. Barba, M., Czosnek, H. & Hadidi, A. Historical perspective, development and applications of next-generation sequencing in plant virology. *Viruses* **6**, 106–136 (2014).

53. Pennisi, E. Development cell by cell. *Science* **362**, 1344–1345 (2018).

54. Reinius, B. *et al.* Analysis of allelic expression patterns in clonal somatic cells by single-cell RNA-seq. *Nat. Genet.* **48**, 1430–1435 (2016).

55. Plasschaert, L. W. *et al.* A single-cell atlas of the airway epithelium reveals the CFTR-rich pulmonary ionocyte. *Nature* **560**, 377–381 (2018).

56. Montoro, D. T. *et al.* A revised airway epithelial hierarchy includes CFTR-expressing ionocytes. *Nature* **560**, 319–324 (2018).

57. Evans, D. M., Visscher, P. M. & Wray, N. R. Harnessing the information contained within genome-wide association studies to improve individual prediction of complex disease risk. *Hum. Mol. Genet.* **18**, 3525–3531 (2009).

58. Kooperberg, C., LeBlanc, M. & Obenchain, V. Risk prediction using genome-wide association studies. *Genet. Epidemiol.* **34**, 643–652 (2010).

59. Choi, S., Bae, S. & Park, T. Risk Prediction Using Genome-Wide Association Studies on Type 2 Diabetes. *Genomics Inform.* **14**, 138–148 (2016).

60. Wei, Z. *et al.* Large sample size, wide variant spectrum, and advanced machine-learning technique boost risk prediction for inflammatory bowel disease. *Am. J. Hum. Genet.* **92**, 1008–1012 (2013).

61. Okser, S. *et al.* Regularized machine learning in the genetic prediction of complex traits. *PLoS Genet.* **10**, e1004754 (2014).

62. Roshan, U., Chikkagoudar, S., Wei, Z., Wang, K. & Hakonarson, H. Ranking causal variants and associated regions in genome-wide association studies by the support vector machine and random forest. *Nucleic Acids Res.* **39**, e62 (2011).

63. Shi, G. *et al.* Mining gold dust under the genome wide significance level: a two-stage approach to analysis of GWAS. *Genet. Epidemiol.* **35**, 111–118 (2011).

64. Meinshausen, N. & Bühlmann, P. Stability selection. *J. R. Stat. Soc. Series B Stat. Methodol.* **72**, 417–473 (2010).

65. Fisher, C. K. & Mehta, P. Bayesian feature selection for high-dimensional linear regression via the Ising approximation with applications to genomics. *Bioinformatics* **31**, 1754–1761 (2015).

66. Zhou, H., Sehl, M. E., Sinsheimer, J. S. & Lange, K. Association screening of common and rare genetic variants by penalized regression. *Bioinformatics* **26**, 2375–2382 (2010).

67. Bush, W. S. & Moore, J. H. Chapter 11: Genome-wide association studies. *PLoS Comput. Biol.* **8**, e1002822 (2012).

68. Wray, N. R. *et al.* Pitfalls of predicting complex traits from SNPs. *Nat. Rev. Genet.* **14**, 507–515 (2013).

69. Edwards, S. L., Beesley, J., French, J. D. & Dunning, A. M. Beyond GWASs: illuminating the dark road from association to function. *Am. J. Hum. Genet.* **93**, 779–797 (2013).

70. Abdi, H. Bonferroni and Šidák corrections for multiple comparisons. In *Encyclopedia of measurement and statistics* (ed. Salkind, N. J.) **3**, 103–107 (SAGE Publications Inc., 2007).

71. Gibson, G. Hints of hidden heritability in GWAS. *Nat. Genet.* **42**, 558–560 (2010).

72. Bacher, R. *et al.* SCnorm: robust normalization of single-cell RNA-seq data. *Nat. Methods* **14**, 584–586 (2017).

73. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods* **14**, 565–571 (2017).

74. Brennecke, P. *et al.* Accounting for technical noise in single-cell RNA-seq experiments. *Nat. Methods* **10**, 1093–1095 (2013).

75. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **16**, 241 (2015).

76. Lun, A. T. L., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17**, 75 (2016).

77. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods* **13**, 845–848 (2016).

78. Rostom, R., Svensson, V., Teichmann, S. A. & Kar, G. Computational approaches for interpreting scRNA-seq data. *FEBS Lett.* **591**, 2213–2225 (2017).

79. Van der Maaten, L. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579-2605 (2008).

80. Yang, L., Liu, J., Lu, Q., Riggs, A. D. & Wu, X. SAIC: an iterative clustering approach for analysis of single cell RNA-seq data. *BMC Genomics* **18**, 689 (2017).

81. Guo, M., Wang, H., Potter, S. S., Whitsett, J. A. & Xu, Y. SINCERA: A Pipeline for Single-Cell RNA-Seq Profiling Analysis. *PLoS Comput. Biol.* **11**, e1004575 (2015).

82. Usoskin, D. *et al.* Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. *Nat. Neurosci.* **18**, 145–153 (2015).

83. Žurauskienė, J. & Yau, C. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics* **17**, 140 (2016).

84. Kiselev, V. Y. *et al.* SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods* **14**, 483–486 (2017).

85. Aibar, S. *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nat. Methods* **14**, 1083–1086 (2017).

86. Xu, C. & Su, Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **31**, 1974–1980 (2015).

87. Bach, K. *et al.* Differentiation dynamics of mammary epithelial cells revealed by single-cell RNA sequencing. *Nat. Commun.* **8**, 2128 (2017).

88. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* **10**, P10008 (2008).

89. Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. U. S. A.* **105**, 1118–1123 (2008).

90. Dulken, B. W., Leeman, D. S., Boutet, S. C., Hebestreit, K. & Brunet, A. Single-Cell Transcriptomic Analysis Defines Heterogeneity and Transcriptional Dynamics in the Adult Neural Stem Cell Lineage. *Cell Rep.* **18**, 777–790 (2017).

91. Angerer, P. *et al.* Single cells make big data: New challenges and opportunities in transcriptomics. *Curr. Opin. Syst. Biol.* **4**, 85–91 (2017).

92. Ding, J., Condon, A. & Shah, S. P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat. Commun.* **9**, 2002 (2018).

93. Wang, D. & Gu, J. VASC: Dimension Reduction and Visualization of Single-cell RNA-seq Data by Deep Variational Autoencoder. *Genomics Proteomics Bioinformatics* **16**, 320–331 (2018).

94. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).

95. Grønbech, C. H. *et al.* scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics* **36**, 4415–4422 (2020).

96. MacQueen, J. *et al.* Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (eds. Le Cam, L. M. & Neyman, J.) 281–297 (University of California, 1967).

97. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**, 129–137 (1982).

98. Weiss, N. A., Holmes, P. T. & Hardy, M. *A Course in Probability*. (Pearson College Division, 2006).

99. Hartigan, J. A. & Wong, M. A. A K-means clustering algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat.* **28**, 100–108 (1979).

100. Euclid & Barrow, I. *Euclide's Elements: The Whole Fifteen Books Compendiously Demonstrated. With Archimedes Theorems of the Sphere and Cylinder, Investigated by the Method of Indivisibles*. (W. Redmayne, 1714).

101. Pearson, K. & Galton, F. VII. Note on regression and inheritance in the case of two parents. *Proc. R. Soc. Lond.* **58**, 240–242 (1895).

102. Spearman, C. The proof and measurement of association between two things. *Am. J. Psychol.* **15**, 7–101 (1904).

103. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Philos. Mag. J. Science* **2**, 559–572 (1901).

104. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **24** 498–520 (1933).

105. Belkin, M. & Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**, 1373–1396 (2003).

106. Saeed, F., Salim, N., Abdo, A. & Hentabli, H. Combining Multiple Individual Clusterings of Chemical Structures Using Cluster-Based Similarity Partitioning Algorithm. In *Advanced Machine Learning Technologies and Applications* (eds. Hassanien, A. E., Salem, A.-B, M., Ramadan, R. & Kim, T.-H.) 276–284 (Springer Berlin Heidelberg, 2012).

107. Johnson, S. C. Hierarchical clustering schemes. *Psychometrika* **32**, 241–254 (1967).

108. Samuel, A. L. Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Dev.* **3**, 210–229 (1959).

109. Bahdanau, D., Cho, K. & Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* doi:1409.0473 (2014).

110. Povey, D. *et al.* The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding* (IEEE Signal Processing Society, 2011).

111. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 13890 (2017).

112. Montavon, G., Rupp, M. & Gobre, V. Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.* **15**, 095003 (2013).

113. Kononenko, I. Machine learning for medical diagnosis: history, state of the art and perspective. *Artif. Intell. Med.* **23**, 89–109 (2001).

114. Awoyemi, J. O., Adetunmbi, A. O. & Oluwadare, S. A. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics* (eds.Misra, S., Matthews, V. O. & Adewumi, A.) 1–9 (IEEE, 2017).

115. Crawford, M., Khoshgoftaar, T. M., Prusa, J., Richter, A. N. & Al Najada, H. Survey of review spam detection using machine learning techniques. *J. Big Data* **2**, 23 (2015).

116. Porshnev, A., Redkin, I. & Shevchenko, A. Machine Learning in Prediction of Stock Market Indicators Based on Historical Data and Data from Twitter Sentiment Analysis. In *2013 IEEE 13th International Conference on Data Mining Workshops* 440–444 (IEEE Computer Society, 2013).

117. Dogan, Ü., Edelbrunner, J. & Iossifidis, I. Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior. In *2011 IEEE International Conference on Robotics and Biomimetics* 1837–1843 (IEEE, 2011).

118. Nyhoff, J. Algorithms To Live By: The Computer Science of Human Decisions. *Perspect. Sci. Christ. Faith* **69**, 127+ (2017).

119. Barlow, H. B. Unsupervised learning. *Neural Comput.* **1**, 295–311 (1989).

120. Caruana, R. & Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* 161–168 (Association for Computing Machinery, 2006).

121. Schölkopf, B., Smola, A. & Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**, 1299–1319 (1998).

122. Bishop, C. M. *Pattern Recognition and Machine Learning*. (Springer, 2016).

123. Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S. & Sundararajan, S. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th international conference on Machine learning* 408–415 (Association for Computing Machinery, 2008).

124. Müller, K. R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J. & Vapnik, V. Predicting time series with support vector machines. In *Proceedings of ICANN 1997* 999-1004 (Spring LNCS, 1997).

125. Caudill, M. Neural networks primer, part I. *AI Expert* **2**, 46–52 (1987).

126. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

127. Li, Q. *et al.* Medical image classification with convolutional neural network. In *2014 13th International Conference on Control Automation Robotics Vision (ICARV)* 844–848 (IEEE, 2014).

128. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).

129. Collobert, R. & Weston, J. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* 160–167 (Association for Computing Machinery, 2008).

130. Chan, W., Jaitly, N., Le, Q. & Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4960–4964 (IEEE, 2016).

131. McCulloch, W. S. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biol.* **52**, 115-133 (1943).

132. Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65**, 386–408 (1958).

133. Minsky, M., Papert, S. A. & Bottou, L. *Perceptrons: An Introduction to Computational Geometry*. (MIT Press, 2017).

134. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning internal representations by error propagation. *California Univ. San Diego La Jolla Inst. Cog. Science* Technical Report 1:49 (1985).

135. Weng, J., Ahuja, N. & Huang, T. S. Cresceptron: a self-organizing neural network which grows adaptively. In *Proceedings IJCNN International Joint Conference on Neural Networks* 576–581 (IEEE, 1992).

136. Caruana, R. Multitask Learning. *Mach. Learn.* **28**, 41–75 (1997).

137. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (MIT Press, 2016).

138. Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M. & Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence* (AAAI Press, 2011).

139. Graves, A. *et al.* A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 855–868 (2009).

140. Amodei, D. *et al.* Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin. In *Proceedings of The 33rd International Conference on Machine Learning* (eds. Balcan, M. F. & Weinberger, K. Q.) **48,** 173–182 (PMLR, 2016).

141. Kuo, P.-H., Lin, S.-T. & Hu, J. DNAE-GAN: Noise-free acoustic signal generator by integrating autoencoder and generative adversarial network. *Int. J. Distrib. Sens. Netw.* **16** (2020).

142. Glorot, X., Bordes, A. & Bengio, Y. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (eds. Gordon, G., Dunson, D. & Dudík, M.) **15**, 315–323 (PMLR, 2011).

143. Hochreiter, S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertainty Fuzziness Knowledge Based Syst.* **6**, 107–116 (1998).

144. Maas, A. L., Hannun, A. Y. & Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML* **30** 3 (Citeseer, 2013).

145. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).

146. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).

147. Fukushima, K. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**, 193–202 (1980).

148. Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).

149. Duchi, J., Hazan, E. & Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **12**, 2121-2159 (2011).

150. LeCun, Y., Bottou, L., Orr, G. B. & Müller, K.-R. Efficient BackProp. In *Lecture Notes in Computer Science* 9–50 (Springer Science & Business Media 1998).

151. LeCun, Y. *et al.* Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* **1**, 541–551 (1989).

152. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Computer Science & Data Analysis* 17–25 (Chapman and Hall / CRC, 2011).

153. Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. (MIT Press, 2012).

154. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Series B Stat. Methodol.* **67**, 301–320 (2005).

155. Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K. & Müller, K.-R. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. (Springer Nature, 2019).

156. Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J. & Müller, K.-R. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proc. IEEE* **109**, 247–278 (2021).

157. Montavon, G., Samek, W. & Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **73**, 1–15 (2018).

158. Alber, M. *et al.* iNNvestigate neural networks! *J. Mach. Learn. Res.* **20**, 1–8 (2019).

159. Fisher, A., Rudin, C. & Dominici, F. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.* **20**, 1–81 (2019).

160. Ribeiro, M. T., Singh, S. & Guestrin, C. ´Why should i trust you?' Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.* 1135–1144 (ACM, 2016).

161. Hastie, T. J. *Generalized Additive Models*. (Routledge, 2017).

162. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside Convolutional Networks: Visualising image classification models and saliency maps. *arXiv* doi:*1312.6034* (2013).

163. Sturm, I., Lapuschkin, S., Samek, W. & Müller, K.-R. Interpretable deep neural networks for single-trial EEG classification. *J. Neurosci. Methods* **274**, 141–145 (2016).

164. Schütt, K. T., Gastegger, M., Tkatchenko, A. & Müller, K.-R. Quantum-Chemical Insights from Interpretable Atomistic Neural Networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (eds. Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K. & Müller, K.-R.) 311–330 (Springer International Publishing, 2019).

165. Kim, D., Sra, S. & Dhillon, I. S. Fast Newton-type Methods for the Least Squares Nonnegative Matrix Approximation Problem. In *Proceedings of the 2007 SIAM International Conference on Data Mining* (eds. Apte, C., Liu, B., Parthasarathy, S. & Skillicorn, D.) (Society for Industrial and Applied Mathematics, 2007).

166. Ren, B. *et al.* Using Data Imputation for Signal Separation in High-contrast Imaging. *Astrophys. J.* **892**, 74 (2020).

167. Aghdam, M. H., Analoui, M. & Kabiri, P. Application of nonnegative matrix factorization in recommender systems. In *6th International Symposium on Telecommunications* (IEEE, 2012).

168. Murrell, B. *et al.* Non-Negative Matrix Factorization for Learning Alignment-Specific Models of Protein Evolution. *PLoS ONE* **6**, e28898 (2011).

169. Kim, H. & Park, H. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics* **23**, 1495–1502 (2007).

170. Stein-O'Brien, G. L. *et al.* Enter the Matrix: Factorization Uncovers Knowledge from Omics. *Trends Genet.* **34**, 790–805 (2018).

171. Taslaman, L. & Nilsson, B. A Framework for Regularized Non-Negative Matrix Factorization, with Application to the Analysis of Gene Expression Data. *PLoS ONE* **7**, e46331 (2012).

172. Wright, S. J. Coordinate descent algorithms. *Math. Program.* **151**, 3–34 (2015).

173. Pan, S. J. & Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **22**, 1345–1359 (2010).

174. Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R. & Serrano L¢pez, A. J. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Technique*. (IGI Global, 2009).

175. Bozinovski, S. Reminder of the First Paper on Transfer Learning in Neural Networks, 1976. *Informatica* **44**, 291-302 (2020).

176. Thrun, S. Is learning the n-th thing any easier than learning the first? In *Advances In Neural Information Processing Systems* 640–646 (Morgan Kaufman Publishers, 1996).

177. Sugiyama, M., Krauledat, M. & Müller, K.-R. Covariate shift adaptation by importance weighted cross validation. *J. Mach. Learn. Res.* **8**, 9851005 (2007).

178. Pratt, L. Reuse of neural networks through transfer. *Conn. Sci.* **8**, 133 (1996).

179. Lin, Y.-P. & Jung, T.-P. Improving EEG-Based Emotion Classification Using Conditional Transfer Learning. *Front. Hum. Neurosci.* **11**, 334 (2017).

180. Chapelle, O. *et al.* Multi-task learning for boosting with application to web search ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM, 2010).

181. Attenberg, J., Weinberger, K., Dasgupta, A., Smola, A. & Zinkevich, M. Collaborative Email-Spam Filtering with the Hashing Trick. In *CEAS 2009 - Sixth Conference on Email and Anti-Spam* (2009).

182. Maitra, D. S., Bhattacharya, U. & Parui, S. K. CNN based common approach to handwritten character recognition of multiple scripts. In *13th International Conference on Document Analysis and Recognition* (IEEE Computer Society, 2015).

183. Hajiramezanali, E., Dadaneh, S. Z., Karbalayghareh, A., Zhou, M. & Qian, X. Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data. In 3*2nd Conference on Neural Information Processing Systems* (eds. Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K. & Cesa-Bianchi, N.) (ACM Curran Associates Inc., 2018).

184. Mihalkova, L., Huynh, T. & Mooney, R. J. Mapping and Revising Markov Logic Networks for Transfer Learning. In *Proceedings of the 22nd National Conference on Artificial Intelligence* **1**, 608–614 (AAAI Press, 2007).

185. Evgeniou, T., Micchelli, C. A. & Pontil, M. Learning Multiple Tasks with Kernel Methods. *J. Mach. Learn. Res.* **6**, 615–637 (2005).

186. Szegedy, C. *et al.* Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2015).

187. Rozier, A. *Harnessing Explainable Neural Networks to Increase the Statistical Power of Genome-wide Association Studies.* (Technical University of Berlin, 2019).

188. Lippert, C. *et al.* An exhaustive epistatic SNP association analysis on expanded Wellcome Trust data. *Sci. Rep.* **3**, 1099 (2013).

189. Geer, S. van de, van de Geer, S., Bühlmann, P., Ritov, Y. & Dezeure, R. On asymptotically optimal confidence regions and tests for high-dimensional models. *Ann. Stat.* **42**, 1166–1202 (2014).

190. Cordell, H. J. Detecting gene–gene interactions that underlie human diseases. *Nature Rev. Genet.* **10**, 392–404 (2009).

191. Van Lishout, F. *et al.* An efficient algorithm to perform multiple testing in epistasis screening. *BMC Bioinformatics* **14**, 138 (2013).

192. Lippert, C. *et al.* FaST linear mixed models for genome-wide association studies. *Nat. Methods*, **8**, 833–835 (2011).

193. Romagnoni, A. *et al.* Comparative performances of machine learning methods for classifying Crohn Disease patients using genome-wide genotyping data. *Sci. Rep.* **9**, 10351 (2019).

194. Abraham, G., Kowalczyk, A., Zobel, J. & Inouye, M. Performance and robustness of penalized and unpenalized methods for genetic prediction of complex human disease. *Genet. Epidemiol.* **37**, 184–195 (2013).

195. Chen, G.-B. *et al.* Performance of risk prediction for inflammatory bowel disease based on genotyping platform and genomic risk score method. *BMC Med. Genet.* **18**, 94 (2017).

196. Botta, V., Louppe, G., Geurts, P. & Wehenkel, L. Exploiting SNP correlations within random forest for genome-wide association studies. *PLoS One* **9**, e93379 (2014).

197. Chen, X. & Ishwaran, H. Random forests for genomic data analysis. *Genomics* **99**, 323–329 (2012).

198. Libbrecht, M. W. & Noble, W. S. Machine learning applications in genetics and genomics. *Nat. Rev. Genet.* **16**, 321–332 (2015).

199. Ching, T. *et al.* Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15**, 141 (2018).

200. Waldmann, P. Approximate Bayesian neural networks in genomic prediction. *Genet. Sel. Evol.* **50**, 70 (2018).

201. Montaez, C. A. C. *et al.* Deep Learning Classification of Polygenic Obesity using Genome Wide Association Study SNPs. In *2018 International Joint Conference on Neural Networks* (IEEE, 2018).

202. Wang, X. *New Nonlinear Machine Learning Algorithms With Applications to Biomedical Data Science.* (University of Pittsburgh, 2019).

203. Uppu, S., Krishna, A. & Gopalan, R. P. A deep learning approach to detect SNP interactions. *J. Softw.* **11**, 965–975 (2016).

204. Kindermans, P.-J. *et al.* Learning how to explain neural networks: PatternNet and PatternAttribution. *arXiv* doi:*1705.05598* (2017).

205. Agresti, A. *Categorical Data Analysis*. (John Wiley & Sons, 2013).

206. Moskvina, V. & Schmidt, K. M. On multiple-testing correction in genome-wide association studies. *Genet. Epidemiol.* **32**, 567–573 (2008).

207. Dickhaus, T. & Stange, J. Multiple Point Hypothesis Test Problems and Effective Numbers of Tests for Control of the Family-Wise Error Rate. *Calcutta Stat. Assoc. Bull.* **65**, 123–144 (2013).

208. Dickhaus, T. *Simultaneous Statistical Inference: With Applications in the Life Sciences*. (Springer Science & Business Media, 2014).

209. Guyon, I. & Elisseeff, A. An Introduction to Feature Extraction. In *Feature Extraction: Foundations and Applications.* 1–25 (Springer, 2006).

210. Wasserman, L. & Roeder, K. High dimensional variable selection. *Ann. Stat.* **37**, 2178–2201 (2009).

211. Meinshausen, N., Maathuis, M. H. & Bühlmann, P. Asymptotic optimality of the Westfall–Young permutation procedure for multiple testing under dependence. *Ann Stat.* **39**, 3369–3391 (2011).

212. Westfall, P. H. & Stanley Young, S. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. (John Wiley & Sons, 1993).

213. Meinshausen, N., Maathuis, M. H. & Bühlmann, P. Asymptotic optimality of the Westfall–Young permutation procedure for multiple testing under dependence. *Ann. Stat.* **39**, 3369–3391 (2011).

214. Dudoit, S. & van der Laan, M. J. *Multiple Testing Procedures with Applications to Genomics*. (Springer Science & Business Media, 2007).

215. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. & Lin, C.-J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. res.* **9**, 1871–1874 (2008).

216. Chang, C.-C. & Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 1–27 (2011).

217. The Apache Software Foundation. Commons Math: The apache commons mathematics library. Available at commons.apache.org/proper/commons-math/ (2016).

218. Helleputte, T. LiblineaR: Linear predictive models based on the LIBLINEAR C/C++ library. Available at rdrr.io/cran/LiblineaR/ (2015).

219. Turner, S. D. qqman: an R package for visualizing GWAS results using Q-Q and manhattan plots. *bioRxiv* doi: 10.1101/005165 (2014).

220. Dowle, M. *et al.* data. table: Extension of 'data. frame'. Available at cran.r-project.org/web/packages/data.table/index.html (2018).

221. Warnes, G. R., Bolker, B. & Lumley, T. gtools: Various R Programming Tools. Available at cran.ma.imperial.ac.uk/web/packages/gdata/ (2015).

222. Clayton, D. snpStats: SnpMatrix and XSnpMatrix classes and methods. Available at rdrr.io/bioc/snpStats/ (2015).

223. Chollet, F. Keras: The Python Deep Learning library. *Astrophysics Source Code Library* ascl:1806.022 (2018).

224. Salanti, G. *et al.* Underlying genetic models of inheritance in established type 2 diabetes associations. *Am. J. Epidemiol.* **170**, 537–545 (2009).

225. Clarke, G. M. *et al.* Basic statistical analysis in genetic case-control studies. *Nat. Protoc.* **6**, 121–133 (2011).

226. Marigorta, U. M. & Navarro, A. High trans-ethnic replicability of GWAS results implies common causal variants. *PLoS Genet.* **9**, e1003566 (2013).

227. Marigota, U. M., Rodriguez, J. A. & Navarro, A. GWAS: a milestone in the road from genotypes to phenotypes. In *Genome-Wide Association Studies: From Polymorphism to Personalized Medicine* 12 (Cambridge University Press, 2016).

228. Gibson, G. Rare and common variants: twenty arguments. *Nat. Rev. Genet.* **13**, 135–145 (2012).

229. Purcell, S. *et al.* PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **81**, 559–575 (2007).

230. Manolio, T. A. *et al.* Finding the missing heritability of complex diseases. *Nature* **461**, 747–753 (2009).

231. Alexander, D. H. & Lange, K. Stability selection for genome-wide association. *Genet. Epidemiol.* **35**, 722–728 (2011).

232. Consortium, I. H. & Others. A haplotype map of the human genome. *Nature* **437**, 1299 (2005).

233. Pearson, K. On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that it Can be Reasonably Supposed to have Arisen from Random Sampling. In *Breakthrough in Statistics* **2**, 11–28 (Springer Series in Statistics, 1992).

234. Preuss, C., Riemenschneider, M., Wiedmann, D. & Stoll, M. Evolutionary dynamics of co-segregating gene clusters associated with complex diseases. *PLoS One* **7**, e36205 (2012).

235. Kim, H. *et al.* The new obesity-associated protein, neuronal growth regulator 1 (NEGR1), is implicated in Niemann-Pick disease Type C (NPC2)-mediated cholesterol trafficking. *Biochem. Biophys. Res. Commun.* **482**, 1367–1374 (2017).

236. Boender, A. J., van Gestel, M. A., Garner, K. M., Luijendijk, M. C. M. & Adan, R. A. H. The obesity-associated gene Negr1 regulates aspects of energy balance in rat hypothalamic areas. *Physiol. Rep.* **2**, e12083 (2014).

237. Winkler, T. W. *et al.* The Influence of Age and Sex on Genetic Associations with Adult Body Size and Shape: A Large-Scale Genome-Wide Interaction Study. *PLoS Genet.* **11**, e1005378 (2015).

238. Machiela, M. J. & Chanock, S. J. LDlink: a web-based application for exploring population-specific haplotype structure and linking correlated alleles of possible functional variants. *Bioinformatics* **31**, 3555–3557 (2015).

239. GTEx Consortium. The GTEx Consortium atlas of genetic regulatory effects across human tissues. *Science* **369**, 1318–1330 (2020).

240. Mittag, F. *et al.* Use of support vector machines for disease risk prediction in genome-wide association studies: concerns and opportunities. *Hum. Mutat.* **33**, 1708–1718 (2012).

241. Quevedo, J. R., Bahamonde, A., Pérez-Enciso, M. & Luaces, O. Disease liability prediction from large scale genotyping data using classifiers with a reject option. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **9**, 88–97 (2012).

242. Wu, Q., Ye, Y., Liu, Y. & Ng, M. K. SNP selection and classification of genome-wide SNP data using stratified sampling random forests. *IEEE Trans. Nanobioscience* **11**, 216–227 (2012).

243. Minnier, J., Yuan, M., Liu, J. S. & Cai, T. Risk Classification with an Adaptive Naive Bayes Kernel Machine Model. *J. Am. Stat. Assoc.* **110**, 393–404 (2015).

244. Nguyen, T.-T., Huang, J., Wu, Q., Nguyen, T. & Li, M. Genome-wide association data classification and SNPs selection using two-stage quality-based Random Forests. *BMC Genom.* **16**, 5 (2015).

245. Manor, O. & Segal, E. Predicting disease risk using bootstrap ranking and classification algorithms. *PLoS Comput. Biol.* **9**, e1003200 (2013).

246. Hoffman, G. E., Logsdon, B. A. & Mezey, J. G. PUMA: a unified framework for penalized multiple regression analysis of GWAS data. *PLoS Comput. Biol.* **9**, e1003101 (2013).

247. Breiman, L. Random Forests. *Mach. Learn.* **45**, 5–32 (2001).

248. Pahikkala, T., Okser, S., Airola, A., Salakoski, T. & Aittokallio, T. Wrapper-based selection of genetic features in genome-wide association studies through fast matrix operations. *Algorithms Mol. Biol.* **7**, 11 (2012).

249. He, Q. & Lin, D.-Y. A variable selection method for genome-wide association studies. *Bioinformatics* **27**, 1–8 (2011).

250. Fan, J. & Lv, J. Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc.: Series B Stat. Methodol.* **70**, 849–911 (2008).

251. Li, J., Zhong, W., Li, R. & Wu, R. A fast algorithm for detecting gene-gene interactions in genome-wide association studies. *Ann. Appl. Stat.* **8**, 2292–2318 (2014).

252. Mimno, D., Blei, D. M. & Engelhardt, B. E. Posterior predictive checks to quantify lack-of-fit in admixture models of latent population structure. *Proc. Natl. Acad. Sci. U. S. A.* **112**, E3441–50 (2015).

253. Rakitsch, B., Lippert, C., Stegle, O. & Borgwardt, K. A Lasso multi-marker mixed model for association mapping with population structure correction. *Bioinformatics* **29**, 206–214 (2013).

254. Loh, P.-R. *et al.* Efficient Bayesian mixed-model analysis increases association power in large cohorts. *Nat. Genet.* **47**, 284–290 (2015).

255. Song, M., Hao, W. & Storey, J. D. Testing for genetic associations in arbitrarily structured populations. *Nat. Genet.* **47**, 550–554 (2015).

256. Zhou, X. & Stephens, M. Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nat. Methods* **11**, 407–409 (2014).

257. Kang, H. M. *et al.* Efficient control of population structure in model organism association mapping. *Genetics* **178**, 1709–1723 (2008).

258. Goodfellow, I., Bengio, Y. & Courville, A. Chapter 9. Convolutional networks. In *Deep Learning*. (MIT Press, 2016).

259. Lapuschkin, S. *et al.* Unmasking Clever Hans predictors and assessing what machines really learn. *Nat. Commun.* **10**, 1096 (2019).

260. Haeusermann, T. *et al.* Open sharing of genomic data: Who does it and why? *PLoS One* **12**, e0177158 (2017).

261. Bengio, Y., Courville, A. & Vincent, P. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013).

262. Goodfellow, I. J. *et al.* Challenges in Representation Learning: A Report on Three Machine Learning Contests. In *Neural Information Processing* 117–124 (Springer Berlin Heidelberg, 2013).

263. Nosofsky, R. M., Sanders, C. A., Meagher, B. J. & Douglas, B. J. Toward the development of a feature-space representation for a complex natural category domain. *Behav. Res. Methods* **50**, 530–556 (2018).

264. Ruder, S. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv* doi: 1706.05098 (2017).

265. Koutsoukas, A., Monaghan, K. J., Li, X. & Huan, J. Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *J. Cheminform.* **9**, 42 (2017).

266. Domhan, T., Springenberg, J. T. & Hutter, F. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-fourth international joint conference on artificial intelligence* (AIII Press, 2015).

267. Wasserstein, R. L. & Lazar, N. A. The ASA Statement on p-Values: Context, Process, and Purpose. *Am. Stat.* **70**, 129–133 (2016).

268. Bostrom, N. & Yudkowsky, E. The ethics of artificial intelligence. In *The Cambridge handbook of artificial intelligence* **1**, 316–334 (Cambridge University Press, 2014).

269. Hellwege, J. N. *et al.* Population Stratification in Genetic Association Studies. *Curr. Protoc. Hum. Genet.* **95**, 1.22.1–1.22.23 (2017).

270. Chen, L., Li, C., Miller, S. & Schenkel, F. Multi-population genomic prediction using a multi-task Bayesian learning model. *BMC Genetics* **15**, 53 (2014).

271. Eisen, M. B., Spellman, P. T., Brown, P. O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. U. S. A.* **95**, 14863–14868 (1998).

272. Inamura, K. *et al.* Two subclasses of lung squamous cell carcinoma with different gene expression profiles and prognosis identified by hierarchical clustering and non-negative matrix factorization. *Oncogene* **24**, 7105–7113 (2005).

273. Chi, K. R. Singled out for sequencing. *Nat. Methods* **11**, 13–17 (2014).

274. Nawy, T. Single-cell sequencing. *Nat. Methods* **11**, 18 (2014).

275. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).

276. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161**, 1202–1214 (2015).

277. Bendall, S. C. *et al.* Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell* **157**, 714–725 (2014).

278. Kim, J. K. & Marioni, J. C. Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. *Genome Biol.* **14**, R7 (2013).

279. Grün, D. *et al.* Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature* **525**, 251–255 (2015).

280. Buettner, F. *et al.* Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol.* **33**, 155–160 (2015).

281. Keren-Shaul, H. *et al.* A Unique Microglia Type Associated with Restricting Development of Alzheimer's Disease. *Cell* **169**, 1276–1290.e17 (2017).

282. Regev, A. *et al.* The Human Cell Atlas. *Elife* **6**, e27041 (2017).

283. Trapnell, C. Defining cell types and states with single-cell genomics. *Genome Res.* **25**, 1491–1498 (2015).

284. Crow, M. & Gillis, J. Co-expression in Single-Cell Analysis: Saving Grace or Original Sin? *Trends Genet.* **34**, 823–831 (2018).

285. Poulin, J.-F., Tasic, B., Hjerling-Leffler, J., Trimarchi, J. M. & Awatramani, R. Disentangling neural cell diversity using single-cell transcriptomics. *Nat. Neurosci.* **19**, 1131–1141 (2016).

286. Hockley, J. R. F. *et al.* Single-cell RNAseq reveals seven classes of colonic sensory neuron. *Gut* **68**, 633–644 (2019).

287. Zeisel, A. *et al.* Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**, 1138–1142 (2015).

288. Zeisel, A. *et al.* Molecular Architecture of the Mouse Nervous System. *Cell* **174**, 999–1014.e22 (2018).

289. Darmanis, S. *et al.* A survey of human brain transcriptome diversity at the single cell level. *Proc. Natl. Acad. Sci. U. S. A.* **112**, 7285–7290 (2015).

290. Häring, M. *et al.* Neuronal atlas of the dorsal horn defines its architecture and links sensory input to transcriptional cell types. *Nat. Neurosci.* **21**, 869–880 (2018).

291. Tasic, B. *et al.* Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nat. Neurosci.* **19**, 335–346 (2016).

292. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).

293. Polański, K. *et al.* BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics* **36**, 964-965 (2019).

294. Barkas, N. *et al.* Wiring together large single-cell RNA-seq sample collections. *bioRxiv* doi:10.1101/460246 (2018).

295. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat. Biotechnol.* **37**, 685–691 (2019).

296. Burkhardt, D. B. *et al.* Quantifying the effect of experimental perturbations at single-cell resolution. *Nat. Biotechnol.* doi:10.1038/s41587-020-00803-5 (2021).

297. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.* **36**, 411–420 (2018).

298. Zhang, H. *et al.* A multitask clustering approach for single-cell RNA-seq analysis in Recessive Dystrophic Epidermolysis Bullosa. *PLoS Comput. Biol.* **14**, e1006053 (2018).

299. Forrow, A. *et al.* Statistical Optimal Transport via Factored Couplings. In *Proceedings of Machine Learning Research* (eds. Chaudhuri, K. & Sugiyama, M.) 2454–2465 (PMLR, 2019).

300. Johansen, N. & Quon, G. scAlign: a tool for alignment, integration, and rare cell identification from scRNA-seq data. *Genome Biol.* **20**, 166 (2019).

301. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).

302. Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).

303. Li, X. *et al.* Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nat. Commun.* **11**, 2338 (2020).

304. Johnson, T. S. *et al.* LAmbDA: label ambiguous domain adaptation dataset integration reduces batch effects and improves subtype detection. *Bioinformatics* **35**, 4696–4706 (2019).

305. Gao, X., Hu, D., Gogol, M. & Li, H. ClusterMap: compare multiple single cell RNA-Seq datasets across different experimental conditions. *Bioinformatics* **35**, 3038–3045 (2019).

306. Crow, M., Paul, A., Ballouz, S., Huang, Z. J. & Gillis, J. Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor. *Nat. Commun.* **9**, 884 (2018).

307. Mereu, E. *et al.* matchSCore: Matching Single-Cell Phenotypes Across Tools and Experiments. *bioRxiv* doi:10.1101/314831 (2018).

308. Aran, D. *et al.* Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat. Immunol.* **20**, 163–172 (2019).

309. Srivastava, D., Iyer, A., Kumar, V. & Sengupta, D. CellAtlasSearch: a scalable search engine for single cells. *Nucleic Acids Res.* **46**, W141–W147 (2018).

310. Kiselev, V. Y., Yiu, A. & Hemberg, M. scmap: projection of single-cell RNA-seq data across data sets. *Nat. Methods* **15**, 359–362 (2018).

311. Lieberman, Y., Rokach, L. & Shay, T. CaSTLe - Classification of single cells by transfer learning: Harnessing the power of publicly available single cell RNA sequencing experiments to annotate new experiments. *PLoS One* **13**, e0205499 (2018).

312. Wang, J. *et al.* Data denoising with transfer learning in single-cell transcriptomics. *Nat. Methods* **16**, 875–878 (2019).

313. Lin, C., Jain, S., Kim, H. & Bar-Joseph, Z. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Res.* **45**, e156 (2017).

314. Cristianini, N., Kandola, J., Elisseeff, A. & Shawe-Taylor, J. On Kernel Target Alignment. In *Innovations in Machine Learning: Theory and Applications* (eds. Holmes, D. E. & Jain, L. C.) 205–256 (Springer Berlin Heidelberg, 2006).

315. Van Dijk, D. *et al.* Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell* **174**, 716–729.e27 (2018).

316. Hubert, L. & Arabie, P. Comparing partitions. *J. Classification* **2**, 193–218 (1985).

317. Zylka, M. J., Rice, F. L. & Anderson, D. J. Topographically distinct epidermal nociceptive circuits revealed by axonal tracers targeted to Mrgprd. *Neuron* **45**, 17–25 (2005).

318. Li, C.-L. *et al.* Somatosensory neuron types identified by high-coverage single-cell RNA-sequencing and functional heterogeneity. *Cell Res.* **26**, 967 (2016).

319. Picelli, S. *et al.* Full-length RNA-seq from single cells using Smart-seq2. *Nat. Protoc.* **9**, 171–181 (2014).

320. Chennupati, G., Vangara, R., Skau, E., Djidjev, H. & Alexandrov, B. Distributed non-negative matrix factorization with determination of the number of latent features. *J. Supercomput.* 76, 7458–7488 (2020).

321. Yin, J., Gao, L. & Zhang, Z. Scalable Nonnegative Matrix Factorization with Block-wise Updates. In *Machine Learning and Knowledge Discovery in Databases* 337–352 (Springer Berlin Heidelberg, 2014).

322. Hegedűs, I., Jelasity, M., Kocsis, L. & Benczúr, A. A. Fully distributed robust singular value decomposition. In *14-th IEEE International Conference on Peer-to-Peer Computing* 1–9 (IEEE, 2014).

323. Wang, D., Vipperla, R., Evans, N. & Zheng, T. F. Online Non-Negative Convolutive Pattern Learning for Speech Signals. *IEEE Trans. Signal Process.* **61**, 44–56 (2013).

324. He, X., Kan, M.-Y., Xie, P. & Chen, X. Comment-based multi-view clustering of web 2.0 items. In *Proceedings of the 23rd international conference on World wide web* 771–782 (Association for Computing Machinery, 2014).

325. Liu, J., Wang, C., Gao, J. & Han, J. Multi-View Clustering via Joint Nonnegative Matrix Factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining* 252–260 (Society for Industrial and Applied Mathematics, 2013).

326. Rogers, D. J. & Tanimoto, T. T. A Computer Program for Classifying Plants. *Science* **132**, 1115–1118 (1960).

327. David, L., Marashi, S.-A., Larhlimi, A., Mieth, B. & Bockmayr, A. FFCA: a feasibility-based method for flux coupling analysis of metabolic networks. *BMC Bioinformatics* **12**, 236 (2011).