PREDICTING THE EFFECTS OF GENETIC VARIANTS IN ALS PATIENTS

NOVEMBER 2019

Utrecht University

UMC Utrecht

Michelle de Groot (6312128)
UTRECHT UNIVERSITY | UMC UTRECHT
SUPERVISORS: VERÓNICA BURRIEL, MATTHIEU BRINKHUIS
EXTERNAL SUPERVISOR: KEVIN KENNA (UMC)

Abstract

Amyotrophic Lateral Sclerosis is a neurodegenerative and lethal disease that causes death 3-5 years after diagnosis. A cure has not been developed yet. Researchers require more knowledge on the genetic architecture of the disease in order to develop a treatment. Up until now, variants in the DNA have been identified as a cause for ALS. To use more opportunities that lie in the field of genetics, DNA data of many patients and healthy controls has been gathered. An initiative that addresses this challenge is Project MinE, that aims to bring researchers, patients and other stakeholders together worldwide. They have created a database with many DNA profiles that could be used for ALS research. In the last decade, it became clear that the focus must be on the whole DNA sequence, instead of only protein coding genes. This other part has a regulatory function, which means that it has a major influence on the activity of protein coding genes. Next to this, not only variants that are common in a certain population, but also variants that are rare (but have a damaging effect) must be studied. A technique that can help to make sense of these topics, is machine learning. This Business Informatics thesis aims to compare the two machine learning frameworks “CADD” and “ExPecto” on their ability to predict gene expression effects from variants in regulatory DNA sequences. It is shown that the tools do not perform well on validation data of the GTEx and MPRA projects. Furthermore, the tools do not give any significant predictions for MinE data, when variants of patients and controls are compared. However, it is shown that the ExPecto framework, which was introduced in 2018, outperforms the state-of-the-art technique CADD in the validation phase.

Preface

Before the first period of my thesis, I followed the course 'Bioinformatics in Neuroscience' as a preparation to my time at the ALS centre. In a very short time, all sorts of new terms, tools, data and visualisations were introduced to me. ALS was one of the central subjects of this course, so various researchers of this field came to give guest lectures. Also the data we worked with, was derived from ALS patients. The course was a valuable, but intensive, preparation to the thesis.

ALS was a rather specific choice of interest for my research project. I first heard of this disease via the international Ice Bucket Challenge. In the last few years, the disease seemed to pop up via all sorts of ways. There were acquaintances who worked on ALS research and told about this, but I also saw the destructive effects from up-close.

The fact that there is still a lot to discover, since there is no known cause or medicine, made me want to look into it more. On top of that, I discovered that the research group of the UMC was interested in using machine learning algorithms for finding genetic variants of interest. Since I follow the profile 'applied data science' and because of my interest in machine learning, this seemed like the perfect way to broaden my knowledge. Furthermore, the MinE project was introduced to me, which is a remarkable initiative to bring different stakeholders together to gather genome sequences from patients and controls for data analysis purposes.

This research could be of great value for my career in different ways. My current idea is to end up in the field of data science. What kind of role I will be playing in terms of programming (or not) is not yet clear to me, but I know that it is valuable to touch the technical side either way. It is also a great way to see my capability in learning a new domain in a rather short time. In the consultancy practice, it happens a lot that you get to work with different people and projects in various domains. A certain amount of flexibility is needed.

I would like to thank Kevin Kenna for his inspiring ideas and his feedback during my time at the UMC Utrecht. His supervision and knowledge about genetic research have been essential for gaining new insights for my thesis. He has consistently kept me on the right track and he has given me the opportunity to learn a lot more about genetics, machine learning and statistical methods, which is valuable for my further career in the field of data science.

Furthermore, I would like to thank my University supervisors Verónica Burriel and Matthieu Brinkhuis for their time and effort to read my thesis products and give their feedback. They have inspired me in showing how to do research in the Business Informatics field and how to set up such a thesis study in the right way.

Table of contents

1. Introduction

In the summer of 2014, there arose a rather special hype for a good cause that gained popularity very fast. All over the world, people threw buckets of ice water on themselves. Everybody seemed to be involved, even former president Obama joined in august. This phenomenon was called the Ice Bucket Challenge. It was a clever way to raise awareness for ALS and to encourage people to donate. According to an article of July 2016 in the New York Times, a huge amount of 115 million dollars was raised. This was the start of the funding for many research projects, with among others the "project MinE".

The amount of money was satisfying an urgent need, because ALS is not solved yet. This means that the causes are not known and there is no cure, while it is a horrible disease. Symptoms that are seen often, are weakness of the muscles, difficulty with talking (dysarthria), difficulty with swallowing (dysphagia) and spasticity. Patients often have 3-5 years to live after the diagnosis. Eventually, the muscles of an ALS patient will not work anymore, due to the death of motor neurons.

A collaboration between researchers and doctors worldwide was build up to gain new insights on the disease, which goes by the name "Project MinE". The idea was to gather DNA data from 15.000 patients and 7.500 controls, which makes it the largest ALS data project in history. Until now, approximately 50% of this goal is accomplished. A lot of money is needed to gather this data, because the technique "Whole Genome Sequencing (WGS)" is used. This is a way to sequence the whole human DNA string, so every little piece can be taken into account in disease research. Sequencing DNA was an expensive job up until a few years ago, when it was still thousands of dollars. Luckily, technical developments made the price drop to €100.

With the data of Project MinE, researchers are able to find specific parts of the DNA sequence that are representative for having ALS. More specifically, the patients have certain genetic variants that cause the disease. For instance, a disruptive variant could appear in a gene. This causes a dysregulation in the production of proteins. In the year 2000, two influential people spoke about the promising possibilities of finding genetic risk factors. The director of the National Human Genome Research Institute said that diagnosis based on the DNA sequence could be realised in 10 years. Furthermore, U.S. president Clinton predicted that it would "revolutionise the diagnosis, prevention and treatment of most, if not all, human diseases" (1). Until now, these statements are not completely fulfilled yet, but genetics has certainly broadened the understanding of the architecture of diseases. Additionally, it serves as an inspiration for new personalised medication.

Research on ALS is necessary, because the exact causes have not been found yet. There is also no medicine that can cure the patients. However, if the relevant variants and genes could be identified, then it helps to develop personalised medicine in the future. The technique that could be used for this, is called "gene therapy", which targets a gene that possesses the disruptive effect.

Until a few years back, research was mainly focussed on variants in a small part of the DNA sequence: the protein coding genes. However, the other 99% of the sequence, also called regulatory or non-coding DNA, contains a lot of disease risks. The influences of variants in this part are much harder to interpret, since they have an indirect effect on the production of proteins (2). In (3), which is a paper from 2006, the need for more knowledge on the non-coding regions is stated. Especially the underlying mechanisms that are responsible for a certain expression change. Researchers have been aiming to address this problem by developing several data analytics techniques, that are able to identify disruptive variants in regulatory regions. A technique that has been gaining popularity in many areas in the last years, is machine learning. This is also something which is used in the medical area more and more. It goes hand in hand with the increasing amount of data that can be gathered

and stored, which makes it interesting to invest in complex data analytics techniques. On top of that, the developments in computing power have helped the use of complex algorithms to analyse data.

In ALS research, there is still much to learn about the genetic architecture and the effects of variants. While approximately 30 ALS genes have been found, it is not yet enough. These discoveries only explain a small percentage of all patients. Opportunities lay in rare variants in the regulatory regions. These variants could have a damaging effect on the expression of protein coding genes. They could alter it in such a way, that they contribute in the development of ALS. To explore this area more, various data analytics techniques will be used.

This Business Informatics thesis study aims to use machine learning algorithms to identify common and rare variants in regulatory regions in ALS patients. The opportunity will be taken to apply and compare machine learning techniques to a real world case, with the goal of finding the most suitable option for ALS research. In section 1.1, the topic and motivation are further elaborated upon. The aim of the research is explained in 1.2. The document outline can be found in 1.3.

1.1 Topic and motivation

Up until a few years ago, research on the genetic causes of ALS have been mainly focused on 1% of the DNA: the protein coding regions. This was simply because the rest of it was seen as irrelevant. In 2013 researchers found out that the 99% of the human genome that was considered 'junk DNA', actually consists of very relevant regulatory regions. They do not code for proteins themselves, but they are able to turn genes on or off. They could actually have an indirect effect on the transcriptional process. Because of this quite recent genomic discovery and the developments in genome sequencing, the effects of regulatory variants have not been studied extensively for ALS.

On top of that, due to budget constraints of research groups, cheaper techniques like Genome Wide Association studies (GWAS) are used. They only take a part of the genomic variation into account. GWAS do provide less detailed information compared to using Whole Genome Sequencing (WGS). GWAS will look at a certain region that could be responsible for a SNP, while WGS will look at the whole genome. If one would combine this with machine learning, certain variants could be found that are responsible for activities in a gene and the resulting phenotype. Luckily, the price of WGS for one person has decreased significantly. Project MinE is focused on gathering DNA profiles with WGS as well.

The UMC Utrecht contains a research department on Neurogenetics that, among others, specialises in ALS. Project MinE is an important initiative of this group. The UMC researchers have been using various machine learning techniques and other data analysis solutions to identify ALS genes. This department has some successful collaborations in finding ALS genes already, which is seen as a major accomplishment. Unfortunately, it is not yet enough to save people. New machine learning techniques are developed all over the world for the area of genetics and some could be of great impact to ALS research. This is the reason why ALS research groups need all hands on deck. More specifically, people from different areas are needed, like medical doctors, researchers, IT people and so on. A student with an Informatics background is able to understand data science techniques and apply them. However, a thorough understanding of the genetics area is required.

1.2 Research objective

In the field of genetics, it is a challenging task to map the expression effects of variants. Especially because a large part of the DNA sequence is non-coding. However, there are tools that are able to predict these effects to some extent.

This thesis aims to find the machine learning techniques that serve the goal of discovering disruptive effects of regulatory variants in ALS patients. Insights on the differences and similarities between tools will declare which tool serves the goal best. It will be a chance to learn from machine learning tools for this complex genetic problem. Finding new disruptive variants will serve as input for further ALS research.

It is now known that ALS is caused by a combination of common and rare variants. Since rare variants are more difficult to identify than common variants, it is valuable to use tools that can detect both types. By studying the differences between patients and controls, the changes in gene expression effects can be found. This is not only an opportunity to find variants to keep an eye on in further research, but also to find the exact disruptive effects in gene expression.

The aim is to bridge a gap that exists in ALS research. Until now, there is no literature on expression changes from variants in regulatory regions of ALS patients, while this opportunity is recognised by many scientists and described in a plethora of published papers, as is described further in this proposal.

1.3 Document outline

This thesis document has the goal to find the machine learning algorithm(s) that are able to find disruptive regulatory variants in ALS patients. This is done by documenting the discovery and comparison of the algorithms. Empirical research and the corresponding empirical cycle with its phases are explained in chapter 2. The empirical cycle is used as the method for this thesis.

The theoretical background is given in the Research Problem Analysis in chapter 3. Chapter 3.1 is about Amyotrophic Lateral Sclerosis (ALS). First the disease itself is explained in 3.1.1 and then the research until now is explained in 3.1.2. Furthermore, important topics in genetics are explained in chapter 3.2. It begins with DNA in 3.2.1, then the transcription process in 3.2.2, then variants in 3.2.3, regulatory DNA regions in 3.2.4 and GWAS and WGA are discussed in 3.2.5. Chapter 3.3 elaborates on the machine learning tools that are available and the comparison between them (3.3.1). Then it moves on to the tools that have been chosen for this thesis: ExPecto in 3.3.2 and CADD in 3.3.2. The last part of the theoretical background is given in chapter 4 about scientific relevance, which gives a review on related works.

Chapter 5 pays attention to the actual use of the two tools and the data that is required. Especially to the steps that must be taken to come to a result. 5.1 is about the setup of ExPecto and chapter 5.2 is about the setup of CADD.

The validation of the two machine learning tools is described in chapter 6. First the GTEx analysis with ExPecto and CADD is presented in 6.1 and then the MPRA analysis with the same two tools in chapter 6.2.

Next, the pre-processing steps for the actual MinE analysis is described in chapter 7, which is about the research execution. The results and insights are presented in the data analysis in chapter 8.

Finally, chapter 9 provides the conclusions, chapter 10 the discussion and chapter 11 the future work.

2. Method

The method of the thesis project will be described in this chapter. The aim of this research is to find the most suitable machine learning technique that gives insights on how variants disrupt the gene expression of ALS patients. Gene expression could also be called “activity”, since it encompasses the turning on-and-off-(or up and down) of a gene. Regulatory DNA sequences will be studied in particular. Machine learning techniques are able to give insights on this. Combining these topics together, resulted in the following main research question:

How can machine learning algorithms predict the way that genetic variants disrupt the activity of regulatory DNA sequences in ALS patients?

This chapter will describe empirical research and the corresponding empirical cycle with its phases.

2.1 Empirical research

For this research, certain guidelines are needed to structure the thesis project. In this case, the empirical cycle will be used, which is a method that is suited for information science research projects. It is described by Wieringa in (4). The empirical cycle is a problem-solving method, that is part of design science. This cycle is preferred, because this research does not involve a design problem. Algorithms that are already designed, are studied in a certain context. There are two major aspects of design science that must be clarified: the artefact(s) and the context. An artefact could be many things, like a method that is studied or an algorithm that is used. The context is what the artefact is designed for, like development or use of software.

In this thesis project, the central artefacts are the tools that are studied. These tools are the ExPecto framework and the Combined Annotation-Dependent Depletion (CADD). They are tools that are designed for complex issues in bioinformatics. The problem context here is showing the disruptive effects of variants in regulatory DNA sequences of ALS patients.

The empirical cycle aims to answer knowledge questions about an artefact in a certain context. An empirical research starts with defining the problem context according to a few questions. After defining the context, an empirical cycle can be initiated.

2.1.1 Context

The first three contextual questions of the checklist are about knowledge goals, improvement goals and the current knowledge. Knowledge goals will specify the problem statement and improvement goals will find possible solutions to the problem. The problem and possible solution are further elaborated on in the introduction of this proposal. The current knowledge will mainly come from published literature. This phase will clarify important topics for the thesis and it will discuss the need for new research. In this third part of the context design, an overview of the current knowledge is provided. The topics to be described will be the foundation for the sub questions. The sub questions that can be answered after this phase are:

SQ1. How are genetic variants able to disrupt the activity of regulatory DNA sequences in ALS patients?

a. What are genetic variants?
b. What is gene activity?
c. What are regulatory DNA sequences?
d. What is ALS?

SQ2. What scientific research has been done on predicting effects of genetic variants in regulatory regions?

a. What research has been done in ALS?
b. How can the effects of genetic variants in regulatory regions be predicted?
c. What are the obstacles in predicting effects of genetic variants in regulatory regions?

SQ3. What ML tools are available to address this problem (predicting the expression effects of variants in regulatory regions) and how could they be compared?

a. Why is machine learning important in identifying risk factors in ALS patients?
b. What are the requirements for the ML tools?
c. What existing ML tools fit the requirements?

The answers to these knowledge questions are part of three major subjects: Amyotrophic Lateral Sclerosis (ALS), genetics and the machine learning tools. This can be found in the research problem analysis in chapter 3.

2.2 Empirical cycle

The empirical cycle (figure 1) consists of five phases, that can be executed subsequently until the knowledge goals are achieved. The five phases are: research problem analysis, research & inference design, validation, research execution, and data analysis.

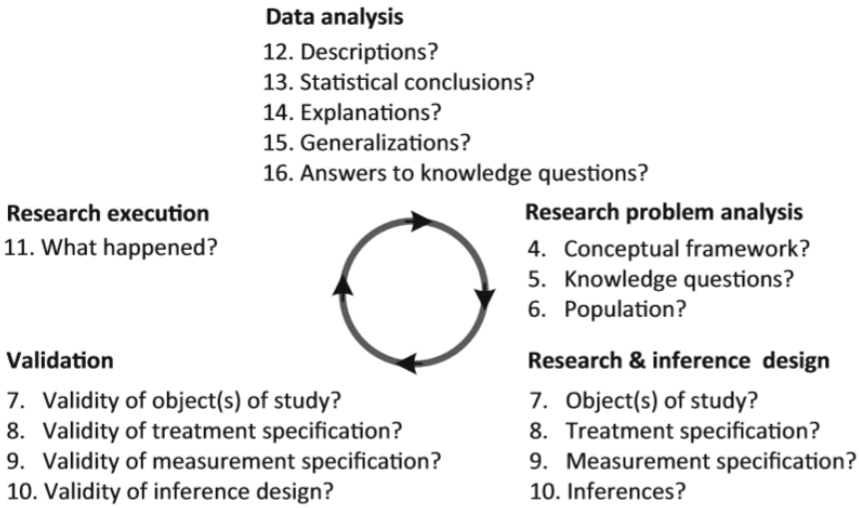


Figure 1. Empirical cycle (4).

2.2.1 Research problem analysis

The research problem analysis is answered by the knowledge questions 1, 2 and 3. It uses a combination of literature and the expertise of different ALS researchers of the neurogenetics UMC group. The problem analysis can mainly be found within these knowledge questions in chapter 3 and in the introduction chapter of this proposal.

2.2.2 Research and inference design

The research design is about the suitable setup. In this project, there will be made use of a sample-based research. Generalization to a population happens when the researcher applies objects of study to samples of the population. Main concepts in this phase are the objects of study, the population, the treatment and the measurement.

The treatment encompasses the way in which the objects of study (machine learning tools) are applied to the population samples. Important features to study here are the data, the pre-processing steps that are needed and other requirements of the tools that will be used. Before starting with the machine learning tools, some skills need to be improved, like working with different genetic file formats and working with languages like Python and the Bash (for the terminal), which are not fully known to the student yet. These are necessary to implement the ExPecto framework. Accordingly, the following sub-question will be answered in this phase:

SQ4. What kind of data and pre-processing steps are required by the tools?

ExPecto has not been used before on ALS data and might provide new insights on the disease. The data for this tool has to be gathered, pre-processed and analysed from scratch. CADD has already been used before on ALS data, but it will be used again, since it can be run with the same version of the data that will be available for ExPecto. A tool might also have a complementary function, but this is not known beforehand. This happens for instance, if both tools have the same quality and they highlight significant disruptive variants in ALS patients.

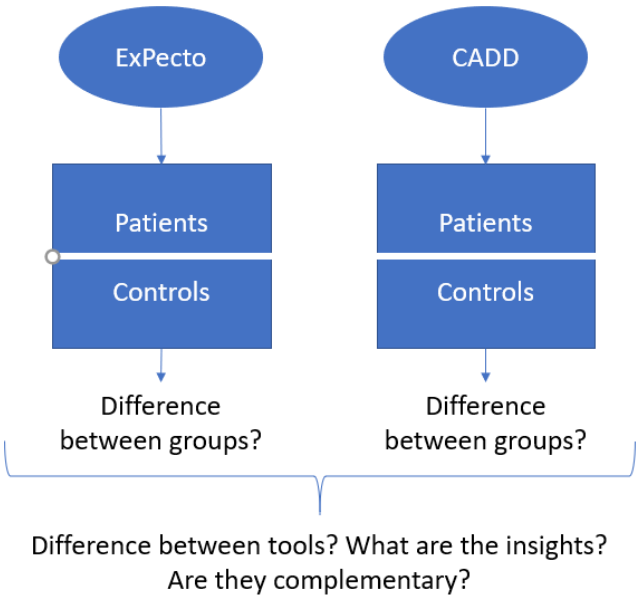


Figure 2. The treatment design for the objects of study. Both tools are used on data of patients and controls. After getting results, the outcomes are compared to each other.

Comparing the outcomes is made possible by the measurement design. The machine learning tools must provide an outcome that can be compared to one and other. They provide a score or a value that gives an indication about the effects of the variants. More specifically, a change in gene expression they might cause. The outcomes of tools will be compared with a statistical test. This is visualised in figure 2.

The inference design is about drawing conclusions from the data analysis. This will clarify how the outcomes of the machine learning tools can be interpreted. The way in which the tools will gain information on variant effects in regulatory regions in ALS patients, needs to be checked.

2.2.3 Validation

For this research, it is of high importance that the answers to the knowledge questions are reliable. This could be accomplished by validating the research design. Literature and expert knowledge will help to design the research in a correct way to provide a sufficient background. In order to check if

the algorithms produce a reliable outcome, the tools (objects of study) must be validated. One of the main obstacles in algorithms that discover effects of regulatory variants, is the validation. Since this is a relatively new research field, there are not many datasets available to validate with. A way to do this, is testing them on variants that are known to have an expression effect. For this purpose, data from the GTEx project and data from experiments with Massively Parallel Reporter Assay (MPRA) can be used. These are initiatives to map expression effects of variants. The GTEx datasets and the MPRA datasets will not provide variants that are associated to ALS, because research in that field has not been published yet. However, there are other already known variants with their effects in these datasets, that will help the validation of the machine learning tools (5).

The GTEx dataset and the MPRA dataset that will be used in the analysis, serve as labelled data. They both contain variants with the known expression effects (labels). ExPecto and CADD will give a prediction for every variant. These predictions are then compared to the actual expression effects. The quality is measured by the use of Receiver Operating Characteristic (ROC) curves. They result in an Area Under the Curve (AUC) and eventually a confusion matrix. These terms are further explained in the Validation chapter 6.

Another way of validation that will be used, is testing the Objects of Study on samples of the larger data files first. Subsequently, the outcome data is checked on unexpected output that doesn't fit the manuals of the tools. For instance, empty rows, columns and fields are checked or the number of rows that it had to produce.

2.2.4 Research execution

In this phase, the research is executed according to the design given in phase 2 of the empirical cycle. The data is gathered, the pre-processing steps are followed and the machine learning tools are used on 46 million variants of the MinE project to gain new insights about ALS patients. In this dataset, there are variants from patients, as well as controls. The information about what variant belongs to who, is not necessary until the Data Analysis phase. In chapter 5, an explanation is given on how to use the two machine learning tools.

2.2.5 Data Analysis

Conclusions could be drawn after the data analysis phase. The outcomes will be ordered, selected and visualised in plots to be able to interpret the data. Statistical tests are used to compare patients and controls within groups. They are also used to compare the outcomes of the two tools. In this last phase, it is very important to be able to interpret the outcome and to see if there are any insights of value for the research question. The analysis method used in this phase, is a gene burden analysis, which is able to give a statistical score to a gene, to see if it is differently expressed in ALS patients. Outcomes of these analyses were plotted in Manhattan plots.

SQ5. What predictions on expression effects are made by the machine learning tools?

a. What was the validation performance of each tool?
b. What variants in patients were significantly different from variants in controls in terms of expression effects?
c. What are the conclusions when comparing the outcomes of the tools?

3. Research problem analysis

3.1 Amyotrophic Lateral Sclerosis

Amyotrophic Lateral Sclerosis (ALS) is a very destructive disease that, unfortunately, affects 1 to 2 newly diagnosed people per population of 100.000 a year. ALS is lethal for all patients, because the right medication has not been developed. Prevention and curing therapy are not yet in sight, mainly because the causes remain largely unknown. After diagnosis, patients have only 3 to 5 years to live (6). Only 20% of the individuals affected with the disease, has familial ALS (fALS). Heritability is the cause of fALS, which indicates that it is manifested in more than one member of a family. The remaining percentage has a sporadic version of the disease (7).

Research that tries to map the development of ALS cases worldwide, has predicted a relatively large increase of 69% from 222.801 in 2015 to 376.674 patients in 2040. This is mainly due to the ageing of population. In EU countries specifically, the number of patients will increase from 29.208 in 2015 to 35.024 in 2040. The overall number could be seen as an underestimate, due to expected positive developments in research in the upcoming years (8).

The developments in ALS research grow exponentially because of new techniques and insights. Hardiman et al. (2017) manage to give a proper review on the disease in terms of epidemiology and discoveries. This review is developed relatively recent and will be used to explain the area of ALS along with other relevant studies.

3.1.1 Disease and phenotypes

ALS patients develop several primary symptoms, like weakness of the muscles, difficulty with talking (dysarthria), difficulty with swallowing (dysphagia) and spasticity. ALS is not solely seen as a disease that causes motor dysfunction, but it is also responsible for cognitive/behavioural impairment. For instance, this impairment is reflected into a form of dementia. Frontotemporal dementia (FVD) has been linked to a mutation in gene C9orf72, which will be further elaborated on in chapter 3.1.2.

When looking at the symptoms, ALS usually could be categorized in bulbar-onset or spinal-onset disease. The former version is expressed in the neck and head. It results in difficulty with speech and swallowing. In 25% of the cases, patients are diagnosed with bulbar-onset disease. Dropping things, falling and other signs of weakness of the muscles in limbs are an expression of spinal-onset disease. The latter is diagnosed in the majority of the patients.

Muscle weakness, the symptom that ALS is most known for, occurs due to the death of upper and lower motor neurons. This happens in three highly influential places of the body: the brainstem, the motor cortex (part of the brain) and the spinal cord (6). Motor neurons are crucial cells for the use of muscles, since they deliver message from the central nervous system to the targeted muscle. People need them for every movement they make. Another characteristic is that there are different kinds of motor neurons. Every kind targets their own muscle type. A certain body movement involves the collaboration of several types of motor neurons to give the right commands (9).

The exact causes of these symptoms remain yet unknown. However, research has shown the importance of genetics, environment and lifestyle for the development of the different phenotypical aspects of the disease. Genome Wide Association Studies (GWAS) have mainly been used to study the genetic architecture. An important outcome was that next to common variants, rare variants were responsible for ALS. This means that the occurred variants individually have a relatively small impact. They could be responsible for a certain phenotypic trait, and together they cause the disease (10). Genome-Wide Association Studies are actually more applicable for diseases with common variants, like Schizophrenia. This is why making use of Whole Genome Sequencing (WGS) is

important for ALS, which is a technique used in the MinE project. The topics in this paragraph will be explained further in chapter 3.2.

Next to genetics, the environment and lifestyle might play a causal role, but not much has been proved about this aspect. Case-control studies have been showing insights for this. One observation is that a significant number of athletes within populations is affected. Another risk-increasing factor seems to be smoking (7). On top of that, other factors seem to be exposure to heavy metals and pesticides. Additionally, trauma is a risk factor, like head injury and bone fracture (11).

3.1.2 Research on ALS

Because of the major technical developments, like machine learning techniques and the increasing computer power, many new insights were gained in the medical area in the past ten years. Additionally, techniques to analyse DNA data also evolved. Whole Genome Sequencing (WGS), machine learning algorithms and supercomputers to analyse the data had a great impact on the discovery of ALS genes. Figure 3 shows the increase of discovered genes in the last ten years, comparing to the first years of research. Approximately 30 genes have been discovered thus far. They all have a certain impact in the development of the disease. In familial ALS, 4 genes account for 70% of the patients: SOD1, TARDBP, FUS and C9ORF72. The search for new responsible genes is essential, because they could be targeted in clinical trials. Otherwise, finding medication is like looking for a needle in a haystack. Familial and sporadic cases were showing an increasing overlap in genetic mutations over the years. This means that some ALS genes in sporadic cases, are also targets in familial cases (7,9).

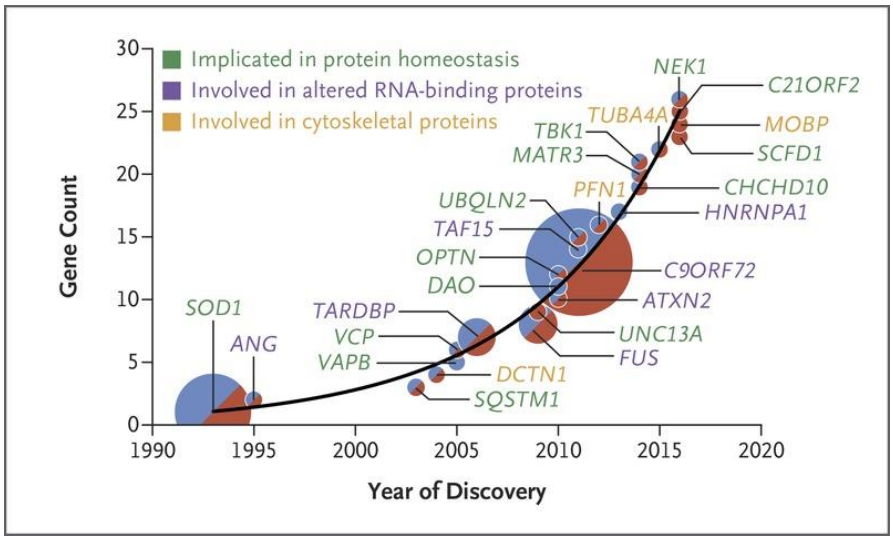


Figure 3: Discovery of ALS genes (11).

When looking at the methods and techniques that provided insights in new ALS genes, it is clear that a wide range was used to make it possible. For instance in finding the gene TUBA4A, a technique called exome-wide rare variant analysis is used, where 1% of the human genome (protein coding part) is sequenced. This provided insight in a combination of rare variants in the gene TUBA4A of familial ALS patients versus controls (12).

Another method is used to find the gene KIF5A, that has been identified quite recently in 2018. A Genome Wide Association Study (GWAS) is carried out to compare patients and controls. On top of that, a rare variant burden (RVB) analysis for patients and control is used. The GWAS resulted in finding a variant (rs113247976) in the KIF5A gene that causes a coding change (13).

Furthermore, gene C9orf72 was a very important discovery in 2011, since it addresses between 25% and 40% of the familial ALS cases and also a small percentage of the sporadic ALS cases. The genetic variant that causes disruptions in the gene, is located in a non-coding regulatory region (that is further elaborated upon in 3.2.4). The main technique that was used to find this variant, is Polymerase Chain Reaction (PCR). As pointed out in 3.1.1, C9orf72 is mentioned as a cause for frontotemporal dementia (FVD) (6,14).

Two other genes that are involved in regulatory regions, are TARDBP and FUS. They are especially involved in the creation of non-coding RNA's. TARDBP and FUS variants can cause a reduced expression of these RNA's. Their downregulation has also been associated with motor neuron cells, which indicates that these cells are extra sensitive to these variants. Researchers have noticed the altered expression of non-coding RNA's, which makes research in regulatory regions of ALS patients increasingly important (15).

Now the question remains why the discovery of disruptive DNA regions is so important. Understanding the genetic basis of ALS will help to find a fitting therapy for patients. Pharmacists have to know where to start, in order to develop medication. That information comes from research groups. The genetic defects could serve as a roadmap for developing new therapies. Promising developments in gene therapy allow specialists to discover the effects of gene silencing and gene editing after targeting the disruptive genetic regions. This field is also known as personalized medicine or treatment (15).

3.2 Genetics

The differences between species are made possible through inheritance; the traits inherited from parents to offspring. For instance, this separates humans from a mouse in terms of characteristics. Differences between individuals of a single species are due to variants that are caused by genetics, gene expression and the environment. Genes are the components of heredity. A gene is part of the genomic sequence and can be located as a region in it. These regions could have various functions, like regulation or transcription or other sequences with a certain function (16). More about gene regions and their mutations will be explained in this chapter.

Gregor Johann Mendel was the first person in history to run an experiment on heredity and write about it. He was an Austrian monk who experimented with pea plants in his garden from 1856 to 1863. His interest was the inheritance of certain traits of the plants, like height and colour, to their offspring. He found out about the principle of inheritance through studying different generations of offspring. Mendel is therefore recognised as the founder of the field of genetics (16).

Not all historic events in the field of genetics will be mentioned in this literature review, but there must be mentioned another major development that happened in 2003: The completion of the human genome sequence by the Human Genome Project. It is seen as a scientific breakthrough, because of the enormous amounts of research opportunities it made possible. The reference genome is a DNA sequence that does not belong to one person, but it can be seen as an average of the vast majority of the human population. It was realised by a large number of research groups all over the world (17).

In the year 2007, one of the first DNA string of an individual was completely sequenced and compared with the reference genome. The individual was one of the people that discovered the structure of DNA in the 50's: James Watson (18). In that time, this study was seen as a great accomplishment, while the costs were almost a million dollar and it took a few months to finish. Fortunately, the sequencing techniques are much more developed and cheap nowadays.

3.2.1 DNA

The discovery of Deoxyribonucleic acid (DNA) could not simply be allocated to one or two scientists. It actually consisted of several scientific events and discoveries by different people in a period of a hundred years. One could see it as pieces of a puzzle, that was finalised by James Watson and Francis Crick in the 1950s. The first identification was done in the 1860s by Friedrich Miescher, who studied white blood cells (19).

DNA is wrapped up in a cell in the structure of a double helix. The individual building blocks of DNA are called nucleotides and consist of four types: Adenine (A), Guanine (G), Thymine (T) and Cytosine (C). The double helix consists of two strands that are complementary in their sequence. Opposite of an A, there is a T and the C matches up with a T. The combination of the two letters is called a "base pair" (20). The whole human genome consists of more than 3 billion of these base pairs. This long sequence is unique to every human, due to genetic variants, which will be explained in sub-chapter 3.2.3 (21).

A closer look at the structure of DNA and how it is wrapped up, is provided in figure 4. The DNA string is hold together by bead-shaped proteins, called histones. Separate groups of eight histones have a part of the DNA string wrapped around it. These groups are called nucleosomes and they form the organisation of the chromatin. The chromatin is the collection of nucleosomes and the DNA between them. Mechanisms play the role in opening and closing the chromatin structure, to make gene expression possible, which is the major role of our DNA. Gene expression is explained further in this chapter. A long string of chromatin is the basis for a chromosome (22,23).

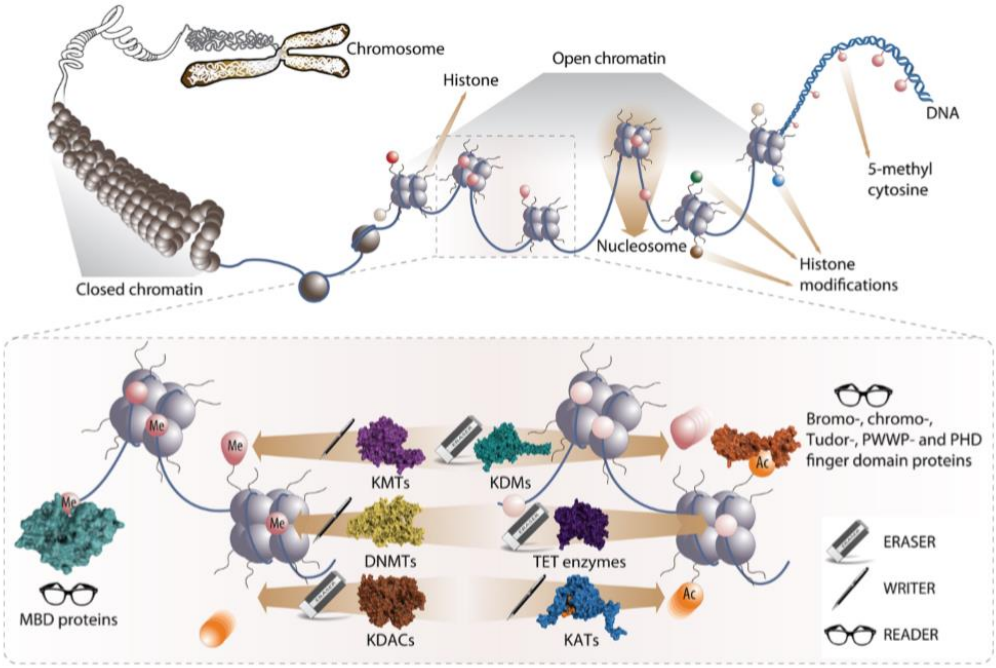


Figure 4. Structure of DNA (21).

The DNA in a cell is divided into 46 chromosomes that bind in pairs. At first, there are pairs of chromosomes 1 to 22 and there are two sex chromosomes X and Y. For a female, the sex chromosome pair consists of two times the X, while a male owns one X and one Y. Some of the chromosomes are longer than others, which allows them to contain more genetic information. Genes can be found on these chromosome strings. They are regions of nucleotides that have a specific function. They can be seen as a manual with tasks to be completed by a cell, in order to survive and

reproduce (21). The process of a gene which is translated into a useful product, is explained in the following sub-chapter.

3.2.2 The transcription and translation process

For a human body to be able to function and interact with its environment, the DNA transcription and translation process is essential (figure 5). This is a process that can be found in every cell and has proteins as outcome product. Proteins are the 'builders' of a cell and take care of the development and identity. The part of the genome that codes for proteins, is only 1% of all DNA, which revolves around approximately 20.000 genes (22). The other 99%, which was called "junk DNA" until recently, will be further elaborated on in part 3.2.4.

The setting of the initial step of the transcription process is the nucleus, which is the core of the cell and can be seen as the command centre. Genes are specific parts in the DNA string and they consist of nucleotides. The gene body has two tails: 5'-end (start) and 3'-end (end). The region that belongs to a gene, can be found by its chromosome and its start- and end-position on this chromosome, for instance gene BRCA2 can be found on chromosome 13 (position 32.315.086 – 32.400.266). These positions indicate the sequence of nucleotides. Expression of a certain gene can only take place if the Transcription Start Site (TSS) is accessible and can be recognised by transcriptome factors. A TSS is located at the 5' tail. The tails will provide information about where to start and where to stop coding. Coding information is relevant to come to the next product: pre-mRNA, which consists of exons and introns. These are developed after transcribing the whole gene body. Next, pre-mRNA is processed into mRNA through splicing, which removes the introns. mRNA only consists of exons that include the 5'UTR and the 3'UTR. For the next step, the mRNA is transported from the nucleus to the cytoplasm of the cell, where it is translated to proteins by an organelle called the 'ribosome'. (22).

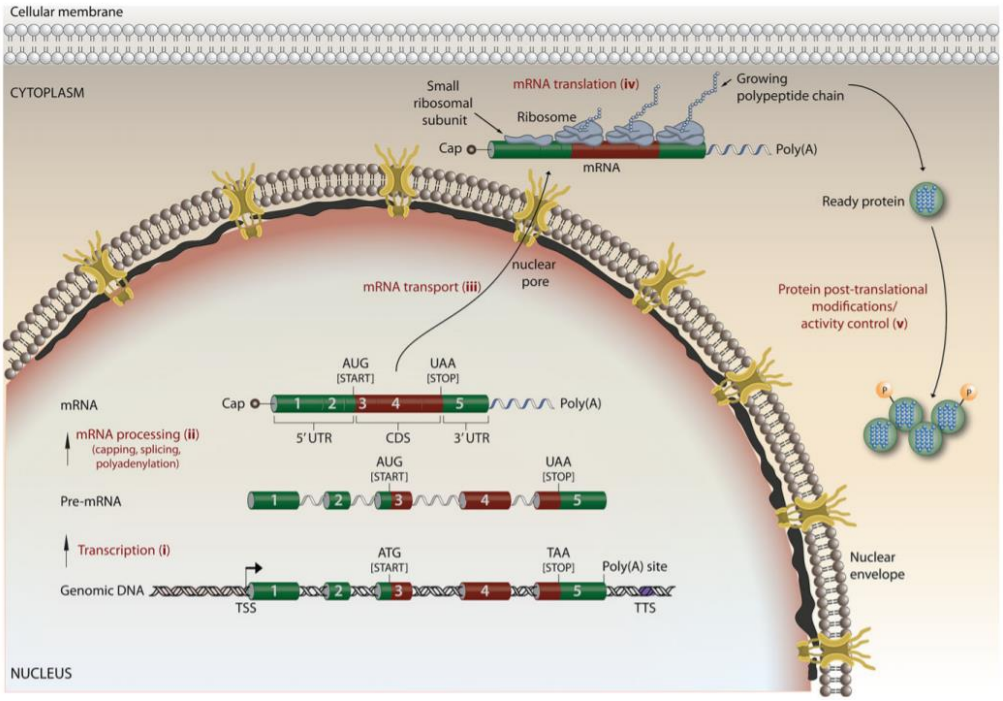


Figure 5. The transcription and translation process (22).

Gene expression is very important for a cell to function in the right way, since it influences the phenotypical, functional and developmental state. Examples of the responsibility of a gene, are the eye color and the skin color. If the gene expression is changed in a bad way and the function or phenotype of the cell is affected, this could result in conditions like cancer or an infection (22).

3.2.3 Variants

The DNA sequence is unique to every human, this is due to variants, which are minor changes in the string of A's, T's, G's and C's. For instance, this explains why someone has blue eyes and another person has brown eyes. Variants (or mutations) could come in different forms, like a change (instead of an A, there is a G), a gap or a duplication. Most DNA disruptions will stay unnoticed or result in changes that are not dangerous or unpleasant. However, other disruptions could be disastrous and eventually cause diseases. The most likely reason for this is erroneous encoding of a protein. If the gene does not encode for the right building blocks (amino acids), the protein won't work or it will work incorrectly (20).

With thousands of genetic variants per person, it is likely to think that a part of them might be harmful and disrupt the transcription process. Diseases could be caused by common or rare variants. These are terms that say something about the frequency of the occurrence within a population. Common variants require large studies to reach statistical significance and are mostly found by Genome Wide Association Studies (GWAS). In the GWAS, they are called Single Nucleotide Polymorphisms (SNPs), which means a change in a single nucleotide (A, C, T, G) in a certain location on the chromosome. A SNP could have multiple alleles. These are the different nucleotides that were found for a SNP in populations. When a disease is caused by a single or multiple common variants, the SNPs are found in a large part of the population. In that case, one SNP is an important contributor to the disease. Statistical significance can be found when patient groups and control groups are compared (10).

Rare variants are characterised by their relatively low frequency in a population. They are not able to be detected by population-based GWAS. The role of causal rare variants, can be seen as a group of variants that all have a small individual contribution to the development of the disease. They can be found by looking into a certain region of the genome. This region, that consists of one gene or several genes, is affected by the variants in terms of function disruption (10).

3.2.4 Regulatory DNA regions

Since finalizing the sequence of the reference genome, researchers have thought that a large part of the human DNA is just "junk DNA". It was 2013 when this was proven to be otherwise. The junk DNA actually had a huge impact on protein coding genes, because it consists of regulatory regions. Genes can only be expressed when they are turned "on". Expression means that a transcription and translation process is initiated. The regulatory regions have the power to upregulate or downregulate genes. Until a few years ago, research was mainly focused on the protein coding parts of the genome. Now regulatory regions have also gained interest, because variants in these regions could be large risk factors for a plethora of diseases (5).

In cell-types or tissues, genes are expressed differently. Cells need to know what their tasks are and how to reproduce themselves. The regulatory sequences of the DNA turn the genes that are necessary for a specific cell-type or tissue on or off. It is a complex coordination function. This is the reason why research in different cell-types and tissues is needed when people want to see the whole spectrum of regulatory variants. A variant in a regulatory region can have a distinct effect in cells in a specific place in the body compared to another place. It is possible that one regulatory sequence is responsible for an expression effect in gene X in a tissue, while in another tissue, this same regulatory sequence is responsible for an expression effect in gene Y. An example is a variant in the regulatory sequence that has a lowered expression effect in gene X in blood cells, while this variant has an increased expression effect in the same gene in liver cells. Once again: it is important to do research on cell-types and tissues of interest if you want to find reliable insights (24).

The term “regulatory region” has been mentioned often in this report. This actually is a category for several regions that all have their own name and function. In (5), they are called promoters, enhancers, silencers and insulators. Their functions are illustrated in figure 6. Promoters are located just in front of the gene in the 5’ region. They activate the transcription process of a gene via the mechanisms Transcription Factors (TF) and RNA Polymerase II (RNAPII). A step in this process is the binding of RNAPII to the TATA-box, which is part of the promoter region. Enhancers can also have an influence on the transcription process through interacting with RNAPII. In contrast to promoters, enhancers could be thousands of base pairs away from the gene. Another characteristic is that they increase gene expression from a distance with the help of proteins called “activators”. These proteins bind the enhancer to the promoter region of a gene. This is possible because of the flexibility of the DNA string. Silencers are regions that decrease gene expression and could also be many base pairs away from their target. Proteins called “repressors” bind to the silencer region and cause this effect. Both activators and repressors are types of Transcription Factors. The fourth regulatory region is the insulator, which could form a barrier between different parts of the chromatin. For instance the link between the enhancer and promotor could be disrupted, so they can no longer interact with each other.

The aforementioned mechanism “transcription factor” plays a central role in gene expression. They work together with proteins to turn genes on or off. Mutations in TF’s could have a major impact in developing diseases, because of the disruption of regulating gene expression. Approximately 30% of the human disease spectrum is caused by mutations in transcription factors (25).

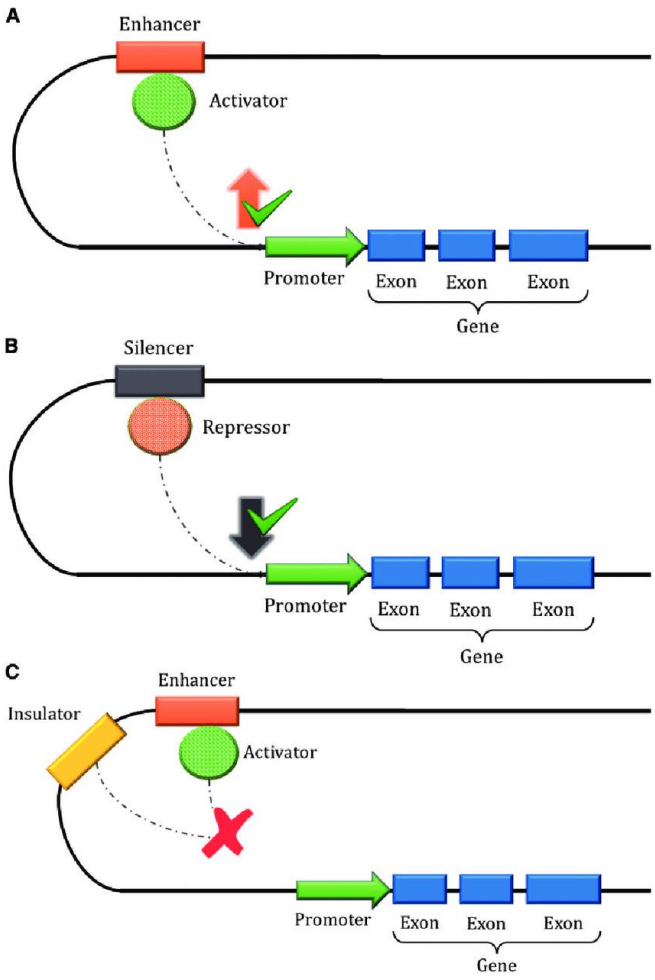


Figure 6. The functions of enhancers, silencers, enhancers and promoters (5)

In (26), the importance of the mechanisms of gene regulation for ALS is highlighted. Especially the role of the non-coding micro-RNA's (miRNA) is referred to. In chapter 3.1.2, several ALS genes with mutations in their non-coding regions have been mentioned. The RNA molecules play their role in how these genes are expressed by binding to DNA sequences that are important to a certain gene. An estimated amount of 60% of protein-coding genes is regulated by miRNA's. The outcome of different researches indicated that they are involved and downregulated in various neurodegenerative diseases, like Parkinson's, Alzheimer's and Huntington's. Thus it is not surprising to see that dysregulation of miRNA's also plays a role in Amyotrophic Lateral Sclerosis. The downregulation and upregulation of these RNA's has already been proven to exist in human patients and in animal models with ALS.

In addition to miRNA's, the role of DNA methylation is also mentioned to be a disrupted regulatory factor in ALS patients. Methylation processes are key in changing the activity of DNA sequences. They interact with transcription factors and can influence the transcription process. In some of the promoter sequences of genes associated to the disease, changing methylation statuses have been reported in patients. It turns out to be majorly involved in transcriptional silencing of a gene. Next to methylation, researchers have shown that histone marks (also known as histone modifications) play their part in ALS gene regulation. Histones are able to activate or silence transcription. They have a small tail to which different molecules are bound, these are called the histone marks. They can regulate DNA accessibility chemical tags. When the tags are "flipped", they can unwind DNA for the

transcription process of genes (20). With this ability, they can control if a gene is turned on or off and they regulate to what extent a gene is expressed, since this differs per cell type (27,28). In ALS mouse models, disrupted histone behaviour was found in the SOD1 gene (one of the ALS genes) in motor neuron cells (26).

3.2.5 Sequencing the whole genome

Since 2002, a technique called Genome Wide Association Studies (GWAS) has been the state-of-the-art in finding genetic causes of traits and diseases. There are drawbacks in using this technique, especially because a lot of valuable information is left out. Whole Genome Sequencing (WGS) is a sequencing technique that is the foundation for more detailed studies, which will be further elaborated upon in this sub-chapter. It is also the technique that was used for retrieving data for this thesis study. The aim is to point out that using WGS goes a step deeper than using GWAS.

GWAS is used to identify regions in the genome that are associated with a specific trait you are interested in, for instance height. Such a region is called a Quantitative Trait Locus (QTL). This is realised by identifying Single Nucleotide Polymorphisms (SNP). A SNP is a genetic variant that occurs in more than 1% of the population. At certain locations in the genome, there could be an A instead of a C, which makes it a SNP. Frequency of SNPs play a large role in GWAS, because the number of SNPs must vary enough between patients and healthy controls, so that they can be picked up. When a SNP pops up that is significantly different between patients and healthy controls, it is not ensured that this is the actual causal variant that is responsible for the trait of interest. This SNP could be related to a QTL (29,30).

This is one of the main drawbacks in using a GWAS: it finds a location that is associated with the trait of interest, but not the exact causal variant. On top of that, it does not show what the expression effects of the variants are for a gene, when it is found in a regulatory region (24).

Another option for finding disease risks is using the technique WGS. Instead of only sequencing at a million SNP positions, the whole human genome is sequenced. This makes it more possible to find rare variants, instead of finding only associations to risk variants. The key is to look at regions or groups of variants that cause a specific trait. It could also be that there are several variants found in a gene (31).

The main differences between GWAS and WGS are being mentioned in (32). The first one is that WGS can take a broader range of variants into account than other sequencing techniques. This means that it can take common and rare variants into account, while GWAS is specialised in finding common variants. WGS is actually able to discover variants as disease-risks that are missed by alternative options. GWAS is only able to give an indication where the disruptive variant could be located. It does not give the precise information about the variant of interest and its effects. Furthermore, by using WGS, one could gain more details from non-coding regulatory regions.

3.3 Algorithms and tools

Evolvement of new technologies to sequence and analyse DNA is responsible for the discovery of ALS genes described in 3.1.2 (11). In 3.2.5, the difference between Genome Wide Association Studies (GWAS) and Whole Genome Sequencing (WGS) is explained. Project MinE addresses the WGS method, which makes more detailed analyses possible, since the whole genome is taken into account. There are a few important factors to be studied that could lead to interesting new insights. They are all in the field of genetics, because the genetic basis of ALS has to be understood better. The data of whole genomes of patients and controls makes it more achievable to find rare variants in the DNA. More specifically, a better understanding of variants in regulatory regions of ALS patients could

be promising and helpful in developing future treatments (15). During this thesis period, the focus will be on the disruptive expression effects of common and rare variants in regulatory regions in DNA sequences.

Finding disruptive variants and predicting the effects of variants in the genome is difficult. Diseases could be caused by variants with a high frequency (common variants) and/or variants with a low frequency (rare variants). Quantitative genetics methods, like GWAS, focus mainly on variants with a high frequency. On top of that, it is difficult to predict transcriptional effects from noncoding parts of the genome. Since 2013, there is an increased interest in developing tools that predict the effects of variants in regulatory regions. In that year, researchers came to the conclusion that the largest part of the human DNA sequence did have an effect on the remaining part: the protein coding genes.

A characteristic for regulatory sequences, is that they regulate gene expression. In chapter 3.2.4, the relevance of studying different cell-types and tissues of interest for finding reliable insights is pointed out. Regulatory regions could change (increase or decrease) gene expression in a specific cell-type (24). For instance, it is known that motor neuron cells are an important factor in ALS, so this could be a field of interest to address with tools.

In this introduction of chapter 3.3, the requirements for finding variants with machine learning tools have been mentioned. At first, the tool has to take the whole DNA sequence into account, this means also the regulatory regions. Secondly, is must not discriminate between variants with a low and a high frequency. Accordingly, this is about common as well as rare variants. The next factor to look at, is the actual effect of the variant in terms of gene expression. Subsequently, studying different cell-types and tissues is relevant for researching regulatory effects. On top of that, the input for a tool (retrieved from patients and controls within project MinE) must be variants with their positions and alleles.

3.3.1 Comparison of tools

With these requirements in mind, appropriate machine learning tools can be looked for. In (5), a set of techniques and methods is defined that are able to identify disease-associated variants in regulatory regions and show their effects. The prediction algorithms use information of the location of the specific variant. If there is an overlap between a variant and a regulatory region, there could be associated to a disease. They are all trained on a large amount of data and then tested to make a reliable classification.

The first tool they mention is CADD, which uses logistic regression and was first introduced in 2014. An important characteristic of CADD is the creation of a “C-score” that gives an indication for the effect of the variant. The advantage is that this score can be utilised by other tools for the prioritisation of variants. The disadvantage is that it is not suitable for non-linear relationships, due to the limitations of the Support Vector Machine. CADD is often used as comparison for newer tools, since it has been the state of the art for a few years, just like in (33) and (34). Another tool that uses a Support Vector Machine, is DeltaSVM. Other supervised algorithms that are mentioned, are GWAVA with its random forest, FATHMM-XF, LINSIGHT with linear and probabilistic models, PRVCS with its composite statistics model, ARVIN with its random forest and DIVAN with its decision tree. Next to the supervised options, there are also some unsupervised alternatives, like Eigen (that uses unsupervised spectral learning) and GenoCanyon. Unsupervised tools that are able to detect non-linear relationships are also mentioned. They use a technique called “deep learning”. For instance, the tools DANN and DeepSEA use neural networks.

A part of the tools (like GWAVA, FATHMM, GenoCanyon and CADD) must be accessed via a website. This means that nothing needs to be done via the command line. The scripts do not need to be run manually, which could be seen as a benefit, especially because no programming or tool knowledge is necessary. A drawback is that parameters cannot be altered and the user is not able to see what happens in the backend. It would be interesting to use tools from a website as well as from the command line. In the last option, there is a possibility to see and understand the process and code in much more detail.

Furthermore, it would be interesting to see the prediction difference between supervised and unsupervised methods. For this study, CADD will be used as a comparison tool, since it has been a commonly utilised prediction method, it meets the requirements and it produces a score for every variant, which makes it convenient material for comparison. However, CADD uses a web entry form, where you must download your variants file in order to get results. Scripts cannot be accessed or modified by the user. It has been the state of the art for years, so it will be interesting to see what effect scores will be added to the variants, compared to other tools. Furthermore, when looking into the publications around CADD, it is made clear that a second version was released in 2018, explained in (35). They released the CADD framework with a whole new method and dropped the idea of the support vector machine, which was introduced in 2014 in (36). Currently, it uses a logistic regression model to score variants.

The field of discovering expression effects of regulatory variants is rapidly evolving (5). Novel knowledge is gained with a fast pace through a plethora of researches. The gathering of new data is also helping to improve the developments in developing and training machine learning algorithms. This is why another quite new tool is introduced in this thesis study, named ExPecto. In (5), this machine learning framework is not mentioned, but its predecessor DeepSEA is. ExPecto has outperformed DeepSEA in finding variants that lead to a significant change in gene expression. It uses a combination of deep learning and linear models, to be able to pick up non-linear relationships (37).

A repertoire of three machine learning tools is mentioned in (2), that address the effects of non-coding variants: DeepSEA, CADD and gkm-SVM. The latter makes use of a support vector machine, just like CADD. It was introduced in 2015, but the framework was updated in 2016 (38). It can be used as input by the previously mentioned deltaSVM, which calculates the effect of sequence variants. Interested people can use the framework by installing the R package. The algorithm finds short sequences in the DNA that are responsible for the activity of enhancers. The gkm-SVM takes tissue specificity into account. However, the input that is available in Project MinE, is data in the form of variants that have been found in patients and controls. A certain transformation of the data would be necessary to use this tool, which is not in the scope of this thesis.

The machine learning tools that have been mentioned, are included in table 1. Cadd and ExPecto have been selected for this thesis study and they are highlighted in yellow.

Model	Year	Includes rare variants	Cell type specific	ML technique	VCF file as input	Code available
CADD	2014 (update: 2018)	Yes	No	Logistic regression	Yes	No
GWAVA	2014	Yes	No	Random Forest	No	No
FATHMM-XF	2017	No	No	Linear regression	Yes	No
LINSIGHT	2017	Yes	Yes	Probabilistic model	No	Yes
PRVCS	2016	No	No	Composite statistics model	Yes	Yes
ARVIN	2017	No	Yes	Random Forest	No	Yes
DIVAN	2016	No	No	Decision Tree	No	Yes
Eigen	2016	Yes	No	Unsupervised spectral learning	No	Yes
GenoCanyon	2015	Yes	Yes	Unsupervised statistical learning	No	No (website also not available)
DANN	2015	Yes	No	Deep neural network	No	Yes
Gkm-SVM	2015 (update: 2016)	Yes	Yes	Support vector machine	No	Yes
DeepSEA	2015	Yes	Yes	Deep neural network	Yes	Yes
ExPecto	2018	Yes	Yes	1. Convolutional Neural Network 2. Regularized linear models	Yes	Yes

Table 1. Machine learning tools and their characteristics.

3.3.2 Deep Learning Framework ExPecto

In 2018 an interesting machine learning framework based on the technique of 'deep learning' was introduced by Zhou et al. The framework is called ExPecto and takes common and rare variants into account to predict the tissue and cell type specific effects. Frequency of variants will not cause bias in terms of finding only high frequency ones. The tool is able to predict effects from regulatory regions (that are 99% of the human genome), that could lead to new insights. For instance, these insights could be parts of the sequence that activate expression of certain genes in the brain tissue. This framework is validated by Zhou et al. by the use of results from GWAS studies.

The necessary input for ExPecto is DNA-variant data in a VCF file. From there, the transcriptional effects of variants can be predicted. Frequency information about variants is not needed beforehand, because then the tool can take common and rare variants into account or even variants that have not been observed yet. Prediction of transcriptional effects is made possible because of the knowledge about transferred information from sequence to transcription. The DNA sequence actually encodes for a certain transcription outcome.

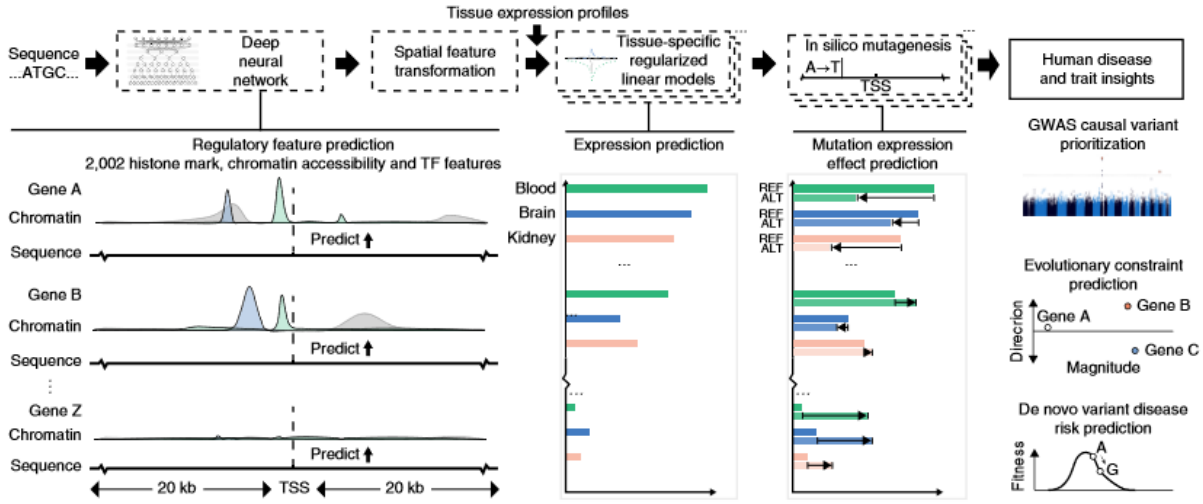


Figure 7. The ExPecto framework: CNN, spatial feature transformation, tissue-specific regularized linear models, in silico mutagenesis, human disease and trait insights (37).

The creators of ExPecto have taken a few main steps to come to this revolutionary framework (figure 7). They first used the technique of deep learning to find connections between regulatory regions, variants and their effects. This resulted in providing more information about a DNA sequence. Deep learning has been gaining popularity the last years, because of the opportunity to gather, store and analyse data faster and more data than before. Furthermore, it is a technique to use on raw, unstructured data like images, sounds or video. A deep learning model is trained on this data and tries to make sense of it by using different layers. There are three types of layers (figure 8) that can be found in a deep neural network: input layers, hidden layers and output layers. In every layer, there is an increased understanding of the data. This is achieved by the work of neurons, which are the circles in the figure. All neurons of a layer are connected to the neurons of the previous layer, just like in the brain and they can be seen as nodes for computations. A neuron or a group of neurons can represent a feature. In every layer, there are neurons that are more active in some parts of the data. This is the way to decide what is important for the desired outcome of the model. Weights are given to neurons/features that explain the data best. When features tend to a higher error, it is given a lower weight than a feature who decreases the error. Eventually the model has a meta understanding of the data and is able to classify new data (39).

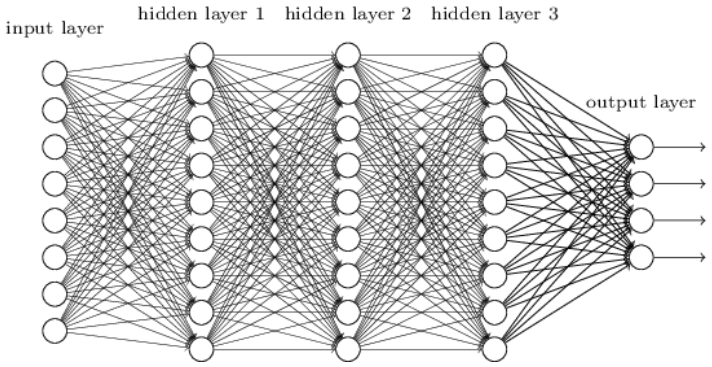


Figure 8. Deep learning neural network with an input layer, hidden layers and an output layer.

The deep learning technique that is used by ExPecto, is a convolutional neural network (CNN). This is a popular algorithm, often applied in image recognition or speech recognition because of its ability to analyse data with multiple arrays. A CNN has many layers and tries to learn something about the features of the input data in every layer. These layers are called convolutional layers and pooling

layers. The aim is to find features and look for their similarities, so they can be “pooled” together by the pooling layer. This all contributes to an overall understanding of the data (39).

To get an understanding of the DNA sequences, the framework needed information on transcription factors, histone marks and DNA accessibility for over 200 different cell types and tissues. More information on these three subjects can be found in chapter 3.2.4 about regulatory DNA regions. Transcription factors, histone marks and DNA accessibility will help the neural network to predict epigenomic effects. Epigenomics is the study of factors that change the way that genes are expressed. The DNA sequence itself is not changed, but the physical properties are altered. These alterations have an impact on whether a gene is more likely to be expressed or not (20).

After the data has been learned by the neural network, a feature transformation method was used for dimensionality reduction. This resulted in spatial feature transformations. The approach was to give weights to regions with respect to their distance to the Transcription Start Site (TSS). More information on the TSS can be found in chapter 3.2.2. The ExPecto framework takes regions of 40 kb around a TSS into account, which can be seen on the left in figure 6. The region of 40 kb means a total of 40.000 base pairs (bp) and can be divided in 20.000 bp on the left of the TSS and 20.000 bp on the right of the TSS. A base pair means one letter (A, C, T, G) in the sequence in this case. If 40 kb is mentioned in this document, it is about the whole region around the TSS, while 20 kb is about one of the sides. ExPecto predicts the effects of variants in this so-called “promoter-proximal” region. This is a regulatory part of the DNA sequence, which has a large influence on a gene, it is also located just in front of the gene.

With the transformed features, a tissue-specific prediction could be made for the expression level of every gene. For this last part, linear regression models are applied with a L2-regularization method. Regularization is used to give penalties to features that don’t add enough information to the desired outcome of a prediction model. Applying this method is of great significance in prediction problems with a large amount of features. The penalties help to prevent overfitting on the training data and to remove the unnecessary complexity, so the model gains efficiency. L1 (Lasso) and L2 (Ridge) are two popular regularization methods to use in predictive modelling. They compute the penalties in a different manner. L1 adds the sum of weights to the function parameters and L2 adds the squared weights to them (40). The outcome of the linear model is the predicted gene expression in specific tissues. During the training phase, sequences of chromosome 8 were kept out of the dataset. They were used to test the model accuracy after the training. The ExPecto model resulted in a 0.821 Spearman correlation.

Until now, measuring the effects of variants by ExPecto have not been mentioned extensively in published papers yet. The goal was to measure the gene expression level in specific tissues. A specific variant has a major impact on the outcome of this model, since it can change gene expression. The genetic code of an individual is modified and this can have an effect on the actual transcription process of a gene. In the model, the difference between the reference and alternative allele is taken into account to predict expression effects. The value that will be measured, is the log(fold change). This is a standard value to measure gene expression effects for two different alleles. It looks at RNA sequencing data for a specific allele. The log2 fold change is the one that is used often. This gives an indication how many times the original expression is increased or decreased by a disruptive variant. For instance, a doubling of the original fold change can be written as log2 fold change of 1 (41).

3.3.4 Combined Annotation-Dependent Depletion (CADD)

The second tool that will be used during this research is Combined Annotation-Dependent Depletion (CADD). This is a framework that went through major developments since 2014, when it was first

introduced in (36). At that time, CADD was still using a support vector machine as the central machine learning technique. In 2018, the next edition was presented in (35). This edition uses a logistic regression model.

The framework has the goal to discover disease-risk variants in the whole DNA sequence (coding and non-coding). It has the power to add scores to every variant, that indicate the destructive effects for a person. To realise this, 60 annotations were taken into account while training the model. Machine learning was used to make a distinction between “new” and “old” variants, which is an idea based on evolution. The old variants, cover the ones that have survived in the human body since millions of years. To be more specific: since the split between chimpanzees and humans. The fact that they have survived for this long, makes them more pure than the novel variants. They have not been removed during the process of natural selection. In the CADD framework, such old variants are being called *proxy-neutral*. Generally, more recently arisen variation in the human genome, contains more deleteriousness, thus higher risk for disease. However, a large part of these new variants could very well be neutral. This second of variants are called *proxy-deleterious* (35).

The CADD framework is visualised in figure 10. It consists of two main phases: The model-fitting in A and the variant scoring in B. In phase A, the logistic regression model is trained to be able to distinguish between proxy-neutral (old) and proxy-deleterious (new) variants. Most users will only go through phase B, which makes use of the trained model to add scores to their own data.

Phase A needed evolutionairy genomic information to define proxy-neutral variants. These variants are the ones that appear in 95% - 100% of the human DNA sequences, but they do not occur in the sequence of the human-ape ancestor. The proxy-neutral variants are used to create a simulated set of proxy-deletorious variants. All variants are annotaded, which means that extra genomic information is added to separate the two sets of variants. This is used as input for training the logistic regression model and eventually to score variants.

The CADD score is the final outcome for every variant, which is a combination of many different features. For instance, these features could be the context surrounding a certain sequence or the evolutionary constraint. The information comes from 60 annotations from several databases and projects, like ENCODE (42) and the UCSC genome browser (43). From these annotations, hundreds of numerical features were derived to realise a classification method.

The CADD framework uses a logistic regression, which is a popular classifier. In this case, it is a dichotomous classification problem, which means that there are 2 classes (proxy-neutral and proxy-deleterious). The goal of the model is to assign a data point to one of the two classes. This can be realised by calculating the probability that a data point is a member of a class. The probability of 0.5 is the decision boundary between the two classes. Parameters determine the outcome of the model. The maximum likelihood estimation is the method that is used to give the parameters the right values. These values must benefit the prediction accuracy (44).

After training the logistic regression model, a prediction was made for approximately 9 billion possible variants of the human reference genome. A first edition of the CADD score is created. This gives an indication if a variant has the probability of being proxy-neutral or proxy-deleterious. If the latter is the case, a variant has a larger risk to be harmful. Positive values are proxy-deleterious and negative values proxy-neutral. The higher the value, the more risk a variant has to be deleterious. However, this raw score is not suitable for comparison to other tools that have the aim to score variants. For the sake of interpretability, a PHRED scaling method will be used for transformation. This scale involves normalising the scores for all the 9 billion variants. A scaled CADD score from 0 to 10 indicates the bottom 90% of the raw scores. These are scores where the probability of a

deleterious effect is low. The scaled CADD score from 10 to 20 indicates the 90% to 99% of the raw scores. Lastly, a scaled CADD score of 20 and higher belongs to the top 1% of the raw scores. These variants have the highest change to be deleterious and thus have a higher disease risk.

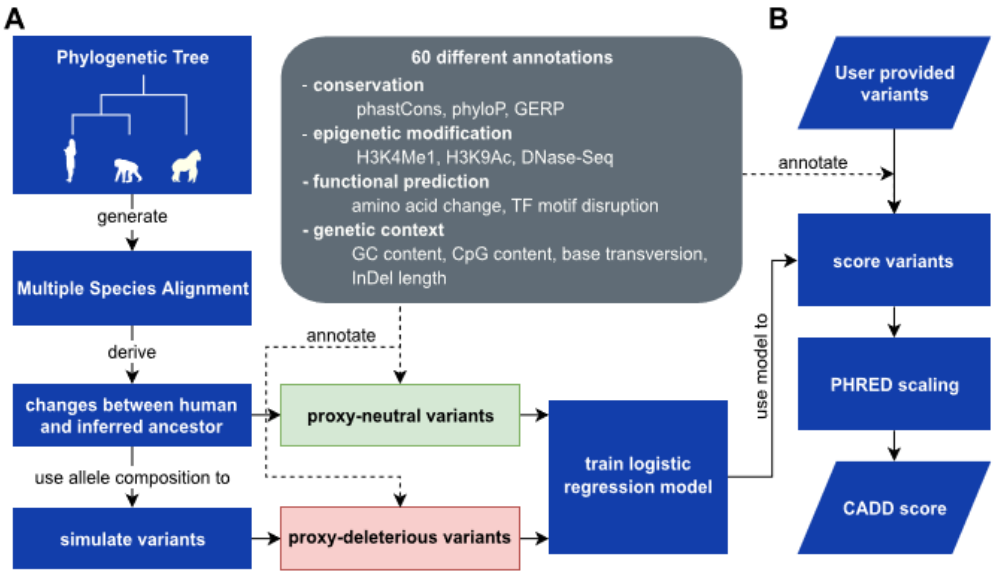


Figure 9. The CADD framework with the model-fitting in phase A and the variant scoring in phase B (35).

3.3.5 Conclusion of ExPecto and CADD

For this thesis study, the two tools CADD and ExPecto have been selected to highlight disruptive variants in regulatory regions. As shown in chapter 3.3.1, there were several candidates that were able to do this. However, many of them did not meet the set of requirements. CADD is a state-of-the-art option that is often used to make comparisons with newer tools. ExPecto is a machine learning framework that has been introduced recently and it is also able to differentiate expression effects between tissues and cell-types. It is a challenge to compare the two tools, since their prediction output for variants is different. CADD gives a PHRED-scaled score that could begin at 0, but it might also be higher than 20. The higher the score, the more risk the variant has to have a disruptive effect on human. ExPecto gives an indication on the log fold change of a variant on gene expression, which is mostly centred around zero. The scores will most likely be between -2 and 2, which is another range than the score that CADD provides. The lower or higher the ExPecto score, the more disruptive the variant is. Scores that are around zero, will not cause a large change in gene expression. For instance, variants with -0.6 or 0.6 are more likely to be disruptive than variants with scores of -0.1 or 0.1. Another difference is that ExPecto provides negative and positive scores and CADD only predicts positive scores.

Despite the difference in output of the models, the idea is the same: predict disruptive variants in regulatory regions. To tackle the challenge of the difference in predictions in the validation phase (chapter 6), the outputs of both models are separated into two classes: non-disruptive (0) and disruptive (1), since that is the only information that counts. The validation phase is necessary to show the quality of the models and to compare them. However, they have already been trained and tested by the developers of the algorithms. The validation in this thesis study can be seen as an extra step and as a way to check the reliability.

4. Scientific relevance

The relevance of epigenetics for ALS is explained in (45), which is a recently published paper from February 2019. The field of epigenetics covers the mechanisms (like DNA methylation, histone modifications and microRNAs) that play a role in gene expression. Evidence for the link between epigenetics and ALS is presented. However, there is a high need for more elaboration on the role of epigenetics in ALS. More specifically, the changes in gene expression need to be highlighted in ALS patients, since a thorough understanding remains yet undefined.

Gene expression is studied in a lot of diseases and disorders. Next to the insights on variant effects for pathogenesis, there are also advantages for medicine research. For instance in (46), a group of researchers did an experiment on patients with bipolar disorder versus healthy controls to see the effects of medication. They found out that the medicine Lithium caused significant expression changes in 236 genes of patients comparing to healthy controls. Statistical differences between the groups were measured with Shapiro-Wilkinson and Mann-Whitney tests.

Furthermore, the importance of studying the effects of rare and non-coding variants is recognised by various researchers. In (47), new low-frequency non-coding variants were identified that were related to Bone Mineral Density (BMD). The technique Whole Genome Sequencing is used to get a broader perspective of the possible variants.

In this thesis study, tools are compared on predicted expression scores. This method is also seen in (34), where their tool GenoNet is compared to eleven other tools. Their interest was to find out if GenoNet performed well on predicting expression effects on tissue/cell-type level and on organism level. The tools were validated using test data and then compared on the metrics Area Under Receiver Operating Characteristic (AUROC), Area Under the Precision Recall curve (AUPR) and a Pearson correlation.

Machine learning tools have been gaining popularity in the medical field in the last 10 years. A more complex technique, deep learning, is increasingly used to make sense of the regulatory parts of the DNA. For instance in (48) and (49), deep convolutional neural networks are presented as a solution. They aim to predict the expression changes that a specific variant causes. The expectation is that machine learning will have many more opportunities in the upcoming years, especially with the increasing computer power, increasing amount of data and new scientific discoveries. Insights on variants in non-coding areas of the DNA will help humanity to figure out the architecture of many diseases.

5. Set up

This chapter will describe the setup of the machine learning models that are used in this thesis: ExPecto and CADD . This encompasses the environment where the tools are run and the process that belongs to it. A High Performance Computing (HPC) environment of the UMC Utrecht is used. When a task is simply too computationally intensive to run on a laptop, a HPC is an option to operate in. The student has her own private space on this environment and can store and run data and scripts on a supercomputer with Central Processing Units (CPU's). There is also the opportunity to work with Graphics Processing Units (GPU's). Currently, not many UMC researchers are working with GPU's, so this environment is quite new and still in the experimentation phase. However, it does not withhold people from using them in their research.

A Linux command line is used to communicate with the HPC. Scripts could be send from there and they have the opportunity to run for hours, days or even weeks. In the HPC environment where the data is stored and run, there is also the setup for Miniconda, which allows for the user to gather programming languages and libraries. The languages Python and R are used from here.

5.1 ExPecto

All the required scripts, data and pre-trained models for ExPecto, can be found on their GitHub. The whole folder was cloned in the HPC, because this thesis project is executed in there. The general pipeline of the ExPecto framework can be found in figure 10. This pipeline is based on the three main steps, that use three scripts: train.py, chromatin.py and predict.py.

1. The linear models of ExPecto can be used to analyse new gene expression profiles for a specific tissue or cell type. This will result in an expression model. It also uses the learned information from the convolutional neural network. Expression level data is an appropriate dataset for this. There is data available from the MINE project that was used and trained. This dataset consists of gene expression levels for the motor neuron cells. Motor neuron cells are the main cell-type that is affected in the disease. Gene expression level data that was used as input, must belong to a person who doesn't have ALS. The ExPecto model must train on "healthy" data, because then it can detect abnormalities in patient data later in the process. The script that is used for this, is **train.py** from the ExPecto Github folder. This first step is optional, since ExPecto is already trained on more than 200 cell types and tissues. However, if a user of ExPecto wants to add another relevant cell-type (like motor neurons), then the model can be trained on the user's gene expression level dataset.
2. The next step is to run the **chromatin.py** script on a VCF file. Variants of patients are gathered in a VCF file with their alternative allele and the reference allele. This is a preferred format of the script. The whole genome could be used for this, but that requires a lot of memory. First, the script was tested on the shortest human chromosome 22. There is a VCF file with variants of patients and controls combined. It contains approximately 2 million variants. Such a number of variants could still take a lot of time and memory to run. This is why the file was split up in smaller chunks. Test files are used to determine the optimal number of variants in such a chunk. The test files contain 10, 100, 1.000, 10.000 and 100.000 variables. The time and memory that it took the HPC to run the files, can be found by using the command "qacct".
3. The output of ExPecto is an output file with the same columns of the VCF file, but then also the predicted expression effects. These effects are calculated by the log 2 fold change. This is a common measurement in the genetic area. This last step is done with the script **predict.py**. The input for this script are the VCF file, the expression models for every tissue (train.py), the

models from the convolutional neural network (chromatin.py) and a closest gene file. The latter is a file with the nearest gene to every variant.

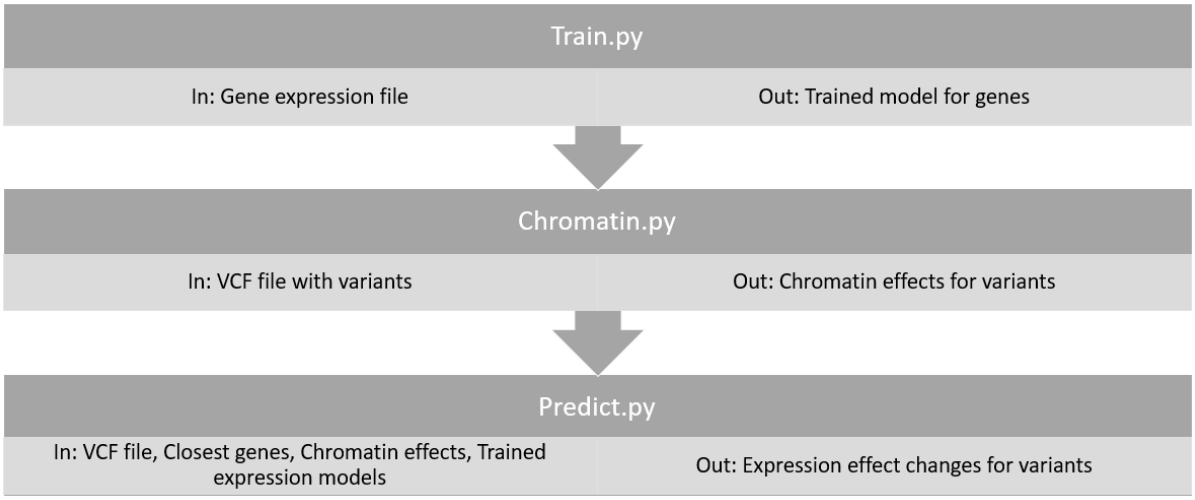


Figure 10. Pipeline of ExPecto Framework.

5.2 CADD

Compared to ExPecto, the CADD framework does not require programming knowledge, since it can be used by uploading a VCF file to their website https://cadd.gs.washington.edu/score. A file of at most 100.000 variants can be uploaded. The columns of the dataset that will be taken into account, are the CHROM, POS, REF and ALT. The outcome of the framework will be the raw CADD score and the PHRED-scaled CADD score for every variant. An extra possibility is to add the annotations to see how a certain score is generated.

The ALS research group has the CADD scores for Single Nucleotide Variants of the Project MinE data already stored in the HPC. In the research execution, these SNV's can be retrieved from the HPC for further analysis. However, the scores for indels are not yet calculated for project MinE data. Indels are a type of variant and they can consist of one or more letter(s) that are deleted or added at a certain place in the DNA. For instance, the reference allele is "TCTAA", but a deletion resulted in an alternative allele of only "T". A SNV is only one nucleotide that is changed, for instance a "T" to a "G" at a specific place in the genomic sequence (50,51).

6. Validation

Validation in this thesis project is necessary to see if the tools are set up in the right way. On top of that, it gives an indication for the comparison that will be made. After this phase, a conclusion could be drawn about reliability of the tools in terms of accuracy on predicting gene expression effects. This conclusion will be taken into account in the actual MinE data analysis.

For the validation phase, two datasets are used to show the quality of the models ExPecto and CADD: GTEx data and MPRA data. Receiver Operating Characteristic (ROC) curves are used as a quality metric in the GTEx analysis, as well as in the MPRA analysis. It is a way to compare observations with predictions of the models. The ROC curve goes hand in hand with the Area Under the Curve (AUC) and eventually a final confusion matrix. As stated before, the gene expression change is a continuous variable. In order to create a ROC curve, the values of the observations and predictions need to be translated to two classes: low expression change and high expression change. The classes are denoted by 0 and 1.

ROC curves and confusion matrices will be used in both analyses. First, the predictions from ExPecto and CADD for variants in the GTEx and MPRA datasets will be received. The expression changes from the GTEx and MPRA datasets can be seen as the observations. The ExPecto output, the CADD scores, the GTEx slope and the MPRA expression changes all indicate a slightly different perspective of gene expression changes. ROC curves are used to give insights about the quality of the models individually, but also to compare them to each other.

Per dataset, the observations are divided into 2 classes: high effect variants (1) and low effect variants (0). There is no known threshold to divide the classes, so a threshold must be retrieved from the distribution of the data. After dividing the observations into classes, the ROC curves plot the True Positive Rate (sensitivity) and the False Positive Rate (1-specificity) of the observations and predictions against each other. The Area Under the Curve is a quality metric for the models. The most optimal threshold for the predictions column (of CADD or ExPecto) can be retrieved after creating the ROC curve. This threshold will divide the predictions also in two classes: high effect variants (1) and low effect variants (0). Finally, a confusion matrix can give more insights on the total true positives, true negatives, false positives and false negatives.

6.1 GTEx

Validating the tools can be done by using labelled data, where the expression effect of a variant is already given. One of the projects that provide this kind of data is Genotype-Tissue Expression (GTEx). Most of the data can easily be accessed through their website https://gtexportal.org/home/datasets. The main file to use for this thesis project is the GTEx_Analysis_v7_eQTL.tar.gz. In this file, there are two files for every tissue/cell-type: an *.egenes.txt.gz file and a *.signif_variant_gene_pairs.txt.gz file. Only the file with the significant variants with a p-value smaller than 0.05 is used. An important note is that these variants are expression Quantitative Trait Loci's (eQTL's). This means that they do not necessarily have to be the disruptive variants, but they might as well be connected to the variant of interest.

The dataset will be pre-processed by using the dplyr package in R. The algorithms will be trained on the specific variants and produce an expression effect. These must be compared with the labels that are in the column "slope", that were given by the GTEx project. In this way, the researcher is able to make a statement about the prioritization of the variants, given by different tools. This can be used for the reliability of the final results in this thesis study.

6.1.1 ExPecto and GTEx

In the ExPecto paper (37), the use of GTEx data is mentioned as well. It is used to evaluate the ExPecto model on tissue specificity, by looking at the direction of expression change. A decreased or increased expression change prediction of ExPecto should match with the slope of GTEx expression. The tool is said to predict the right direction in 92% of the 500 GTEx variants with the biggest effect change. Furthermore, tissues and cell-types from other projects that were already tested individually, are brain cells, primary immune cells and blood cells. In this thesis project, Lymphocytes will be used to validate the ExPecto model. They are a sub-type of white blood cells and dysregulation could for instance lead to cancer. Lymphocytes are also related to the immune system. In the GTEx data, the file with the significant variants of this cell-type is Cells_EBV-transformed_lymphocytes.signifpairs. In the ExPecto model-file for tissues and cell-types, there is also one column/model that is called "Cells.EBV.Transformed.Lymphocytes", which will be used as comparison to the GTEx data.

There are 287.278 variants to be analysed. Since CPU's were used, the file had to be divided in chunks of 10.000 variants. By dividing the large file in smaller chunks, it will run faster. The chromatin representations were given with the script chromatin.py, which made nine models for different distances from the TSS. Furthermore, the closest gene file was created with BEDOPS. This tool can be used in the command line and it has several genomics oriented uses for research, among others the ability to assign variants to the closest gene (52). After comparing the closest BEDOPS genes to genes that were given to a variant by GTEx, it was discovered that there was a disagreement in this step. GTEx assigns genes to variants that are often more than 40 kb away from the TSS. The ExPecto model only looks at the closest genes. For the sake of comparability, a new gene file had to be created, that includes the given genes of GTEx. A file that contains genes with their TSS and additional information was extracted from the ENSEMBL website: ftp://ftp.ensembl.org/pub/grch37/current/gtf/homo_sapiens/. The new closest gene file was created in Rstudio with the following relevant columns: chromosome, variant position -1, variant position, reference allele, alternative allele, strand, ENSEMBL gene id and the distance to the TSS. A part of the closest gene file is visualised in figure 11.

	chr	H	id	ref	alt	strand	gene_id	tss_distance
1	4	17559212	17559213	T	C	+	ENSG00000002549	-19602
2	4	17565142	17565143	T	C	+	ENSG00000002549	-13672
3	4	17567354	17567355	A	C	+	ENSG00000002549	-11460
4	4	17567512	17567513	C	T	+	ENSG00000002549	-11302
5	4	17577328	17577329	A	G	+	ENSG00000002549	-1486
6	4	17578252	17578253	G	A	+	ENSG00000002549	-562
7	4	17578468	17578469	A	G	+	ENSG00000002549	-346
8	4	17579414	17579415	T	C	+	ENSG00000002549	600
9	4	17581642	17581643	CAG	C	+	ENSG00000002549	2828
10	4	17582050	17582051	G	A	+	ENSG00000002549	3236
11	4	17582441	17582442	T	C	+	ENSG00000002549	3627

Figure 11. Part of Closest Gene File for ExPecto's predict.py.

The last step of ExPecto is the execution of script predict.py. As an input, it uses the Lymphocytes variant file, the nine models that were created by chromatin.py, the closest gene file and the modellist file for all tissues. The output is all the csv files for the jobs that were submitted, since the VCF file was split in parts of 10.000 variants. Eventually, the csv files are merged to make the analysis

in Rstudio more interpretable. In R, the output csv file of ExPecto is merged with the GTEx file. This was done with the R-library sqldf, where data frames can be merged on the given columns.

The measurement of similarity between the ExPecto effect sizes and the slope of the GTEx predictions, is done by comparing the columns “Cells.EBV.Transformed.Lymphocytes” and “slope”. The slope is the change in expression effect for a specific allele. After simply plotting the two columns individually, it gets clear that a large number of duplicates is present in the data. Only 198.373 variants of 287.278 are unique. The dataset with unique variants will be used for the further analysis. Descriptive statistics of the GTEx slope and the ExPecto prediction in this dataset can be found in table 2. From these values, it gets clear that the range of the slope is more than four times larger than the ExPecto prediction. Next to that, there is a variation in the mean, since the latter is smaller and closer to zero.

	Min.	**1st Quartile**	**Median**	**Mean**	**3rd Quartile**	**Max.**
GTEx slope	-4.69935	-0.60806	0.35067	0.06545	0.66964	4.25463
ExPecto	-1.1966295	-0.0000823	0	0.0000015	0.0000771	1.1419185

Table 2. Summary data of the GTEx slope and the ExPecto predictions.

First, the method of Zhou et al. is followed, which means that the direction of the GTEx slope is compared to the direction of the ExPecto prediction for every variant. This direction of expression change can be negative or positive. To see if both columns agree, a multiplication is used. The results are saved in a new column. Then, a transformation to TRUE and FALSE is made to make the results more interpretable and structured. When they don’t disagree, a “FALSE” is given in the new column. This means that the multiplied values were both negative or both positive. When there is a disagreement, a “TRUE” is given in the new column to the corresponding row. This means that one of the multiplied values was negative and one was positive. The following code is used to make this column:

```
compare_posneg <- data.frame(AB =
x4$slope*x6$Cells.EBV.Transformed.Lymphocytes)

compare_posneg <- ifelse(compare_posneg > 0, FALSE, TRUE)

x4 <- cbind(x4, compare_posneg)
```

A summary (figure 12) of the column with FALSE and TRUE, shows that for only 40% of the variants, there is an agreement between GTEx and ExPecto on the direction of gene expression. This is without any filter, thus for all 198.373 variants. It is known that ExPecto is designed for variants that are in a range of 40.000 base pairs around a TSS. This means 20.000 base pairs on the left of the TSS and 20.000 base pairs on the right of the TSS. This was further explained in chapter 3.3.2. Adding a filter (with R library dplyr) for an absolute distance above this threshold, shows an agreement in 37% of the variants. For instance, this could be a variant that is located 27.000 base pairs upstream or downstream (on the left or right) of the TSS. However, for variants within this range, the data shows an agreement of 50% (figure 13). For instance, this could be a variant that is located 1.400 base pairs upstream or downstream of the TSS. From here, the decision was made to analyse variants within the range of 40.000 base pairs, since this results in a higher accuracy.

```
> x4 %>% summarise(mean(1-AB), n())
  mean(1 - AB)     n()
1    0.4047779 198373
```

Figure 12. GTEx and ExPecto agree on the direction of expression change in 40% of the total number of variants.

```
> x4 %>% filter(abs(dist)>20000) %>% summarise(mean(1-AB), n())
  mean(1 - AB)     n()
1    0.3717433 147427
> x4 %>% filter(abs(dist)<20000) %>% summarise(mean(1-AB), n())
  mean(1 - AB)   n()
1     0.500373 50944
```

Figure 13. GTEx and ExPecto agree on the direction in 37% of the cases (>20.000 base pairs from TSS) and in 50% of the cases (<20.000 base pairs from the TSS).

Furthermore, another filter on the ExPecto predictions was added during this GTEx analysis, while still comparing the direction of the expression. It is interesting to see the difference in agreement in low or high predictions. The graph in figure 14 shows the increasing accuracy for higher predictions. For instance, in the filter of ExPecto predictions above 0.1 (443 variant in total), there is an agreement on the direction of expression effects in 63% of the cases. In predictions above 0.4 (29 variants in total), there is an agreement on the direction in 79% of the cases. However, a note has to be made that the group sizes differ. The code that has been used for calculating the agreement is the following:

```
X4 %>% filter(abs(dist)<20000) %>%
group_by(abs(Cells.EBV.Transformed.Lymphocytes)>(0.4)) %>%
summarise(mean(1-AB), n()
```

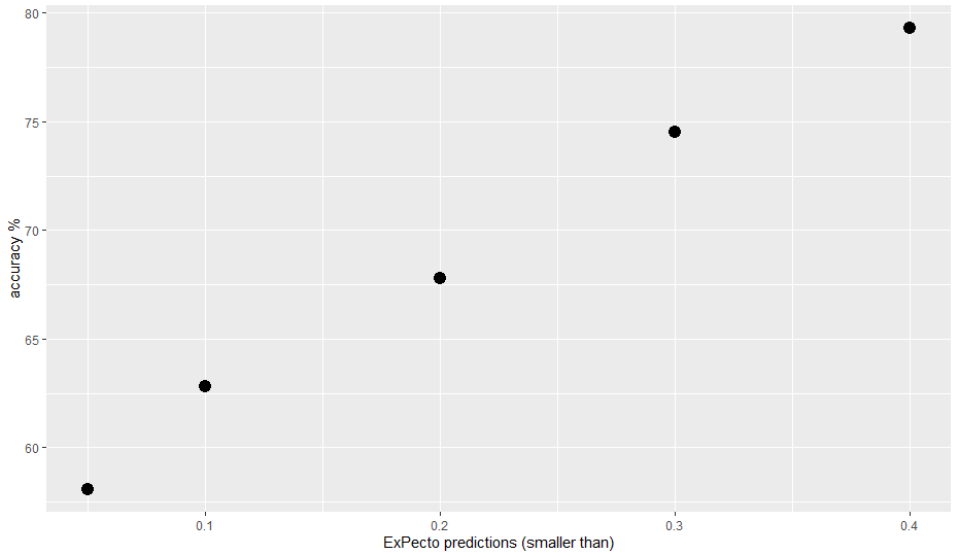


Figure 14. ExPecto predictions on x-axis, with their responding accuracies on the distance on the y-axis.

Another way to compare the ExPecto predictions for variants to the GTEx slopes, is to divide the values in two groups: non-disruptive variants (0) and disruptive variants (1). This creates a chance to see if a variant is assigned to the same class by the observed and the predictions datasets. The

drawback, is that there is no exact threshold for the determination of a disruptive variant. If there is no clear partition in the data, the “mean” is another option to divide the set. The distribution of the absolute GTEx slope is visualised in Appendix 1. The absolute value is used, because there are also slopes with a minus number. Large positive numbers and low negative numbers of the slope column, both indicate a disruptive effect of the genetic variant.

In the histogram of Appendix 1, the data is right skewed. This means that most data is located on the left of the distribution, but there is a long tail on the right with some high values. A clear partition in the data (for non-disruptive and disruptive variants) does not exist. A summary of the data column of the absolute GTEx slope can be found in table 3. The mean of the absolute value of the slope is 0.6966, which will be used for creating an ROC curve.

Min.	**1st Quartile**	**Median**	**Mean**	**3rd Quartile**	**Max.**
0.1422	0.5051	0.6357	0.6966	0.8246	4.6993

Table 3. Summary data of the absolute GTEx slope column.

A ROC curve can give insights on how the observations fit the predictions. A detailed explanation is given in chapter 6. The curve is created in R (code can be found in Appendix 2), with the mean of the absolute value of the GTEx slope as a threshold. It results in a plot of the specificity and the sensitivity against each other. The Area Under the Curve is a metric that gives insight on how precise the model is, when compared with the observation data. An AUC of 0.5 resembles a random model, which is the same as a tossing-coins experiment. Furthermore, the “pROC-package” in R, has the ability to choose the best threshold for the ExPecto predictions. This threshold will also divide the ExPecto column in two classes. From there, a confusion matrix can be calculated, which results in an accuracy metric of the model.

The ROC curve for the ExPecto data in the GTEx analysis, can be found in figure 15. According to the curve, the ExPecto model has a very bad performance, when the observations are compared to the predictions. This could be concluded by the visual representation, as the curve is very close to the diagonal line. It is supported by an Area Under the Curve of 0.522, while an AUC of 0.5 resembles a random model.

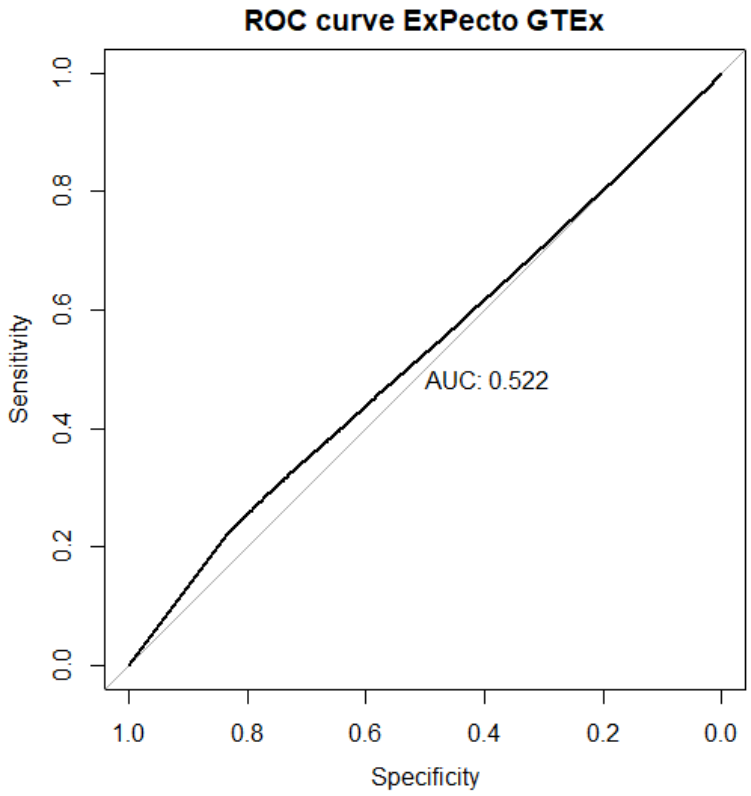


Figure 15. ROC curve of ExPecto data in the GTEx analysis.

The best absolute threshold for the ExPecto Lymphocytes column is 5.960464e-08. After finding this threshold, everything above this value, is assigned to the disruptive class (1). Everything below is assigned to the non-disruptive class (0). The confusion matrix can be found in table 4, which resulted in an accuracy of 41.06%.

	0	***1***
0	20097	17767
1	99149	61360

Table 4. Confusion matrix of the ExPecto GTEx analysis, with a predictions as rows and observations as columns.

A possible explanation for the lack of similarity between the predictions and observations, is that the initial file with the closest genes was changed to other target genes. ExPecto takes variants into account that are relatively close to a Transcription Start Site, while the GTEx variants could be much further away from the gene. For the sake of comparability, the closest gene file that is required by the prediction phase of ExPecto, was changed into genes that were given by GTEx. This means that ExPecto might not have been able to predict the similar expression effects, since it was intended for variants within 20.000 base pairs on the left or 20.000 base pairs on the right from a TSS. However, it does give a prediction for variants outside this range, but they are more likely to be random guesses. Neural networks are not known for their ability to extrapolate, so predictions outside of the range of the training set, are less accurate.

Another explanation is that the GTEx data consists of eQTLs. These do not necessarily have to be causal variants, but they could be connected to disruptive ones. ExPecto has a focus on the actual disruptive variants.

For the MinE data analysis, only variants within a range of 40.000 base pairs around a TSS of a gene will be used. This reduces the chances of ExPecto making random guesses.

6.1.2 CADD and GTEx

Next to ExPecto, the GTEx Lymphocytes data is also used for the validation of CADD. In the CADD-GTEx analysis, it is not possible to compare the direction of the expression effects. CADD only produces a PHRED-score, which gives an indication about the deleteriousness of a variant. Low PHRED scores tend to be more neutral and high PHRED scores tend to be more disruptive.

Table 5 shows summary data of the PHRED score. A transformation to absolute numbers is not necessary, since the scores don't have negative values. The minimum value is 0.001 and the maximum value is 40. The data is right skewed, since the median of 2.733 is smaller than the mean of 3.885.

Min.	1st Quartile	Median	Mean	3rd Quartile	Max.
0.001	1.036	2.733	3.885	5.682	40

Table 5. Summary data of the CADD PHRED score column.

To give an idea about the performance of the predictions of CADD on the GTEx data, a ROC curve is made with the corresponding Area Under the Curve. The same mean of the slope is used: 0.6966, as with the ExPecto analysis. The ROC curve can be found in figure 16, where the specificity is plotted against the sensitivity. The performance of the model is remarkably close to an AUC of 0.5, which means that it is close to a random classifier.

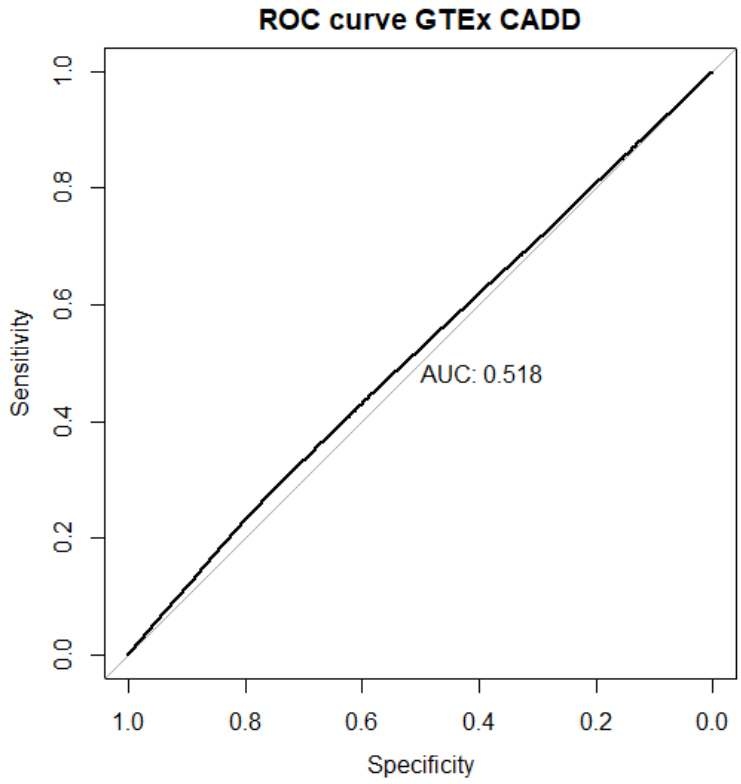


Figure 16. ROC curve of CADD scores in the GTEx analysis.

The best threshold for the CADD PHRED score, assigned by the function of the pROC library, is 4.8995. Every score below this, is assigned to class 0 and everything above this, is assigned to class 1. Class 0 indicates non-disruptive variants and class 1 indicates disruptive variants. The confusion matrix can be found in table 6, which resulted in an accuracy of 55.15%.

	0	***1***
0	83109	55607
1	33348	26288

Table 6. Confusion matrix of the CADD GTEx analysis, with a predictions as rows and observations as columns.

A reason for the bad performance of the CADD framework on GTEx data, could be the same as was already mentioned with the ExPecto GTEx analysis. The GTEx data consists of eQTLs, while the CADD scores are directly given to deleterious variants.

Another reason is that the GTEx data is derived from the Lymphocytes set, which is a specific cell-type. CADD does not give separate scores for cell-types or tissues, but it gives a general score of deleteriousness.

The use of the mean of the GTEx slope as threshold for deciding to what class a variant belongs to, is debatable. Especially when it gets clear that the AUC improves when the threshold is scaled up. For instance with CADD, the AUC goes from 0.518 to 0.530 when the threshold of the GTEx slope is set to 2, instead of the threshold 0.6966.

6.2 Validating the ExPecto Set up

Data to validate your own set up of the ExPecto framework is provided on their website. It is a file with 2.443.754 variants and their predictions for every tissue/cell-type. For this thesis research, the first 10.000 rows are extracted and put into a new file, called website_variants.vcf. Thereafter, the labels (predictions) were removed until only relevant information was left. This was given as input to the ExPecto model on the HPC, without training anything. Only predict.py was used for this validation step. The hypothesis is that the predictions of the ExPecto researchers are in line with the ones from the master student. A Pearson correlation helps to measure this assumption and a plot supports with a visualisation of the data. The two predictions for every tissue were merged into one data frame “compare_website”. A plot of the first tissue “Adipose Subcutaneous”, almost shows a straight line, with a Pearson correlation of 1. The visualisation can be found in figure 17. After calculating the Pearson correlation of the other tissues in Rstudio, the conclusion can be drawn that there exists a correlation of 1 for all of them. This means that the ExPecto framework makes the same predictions in the setup of the student, compared to the one from the researchers.

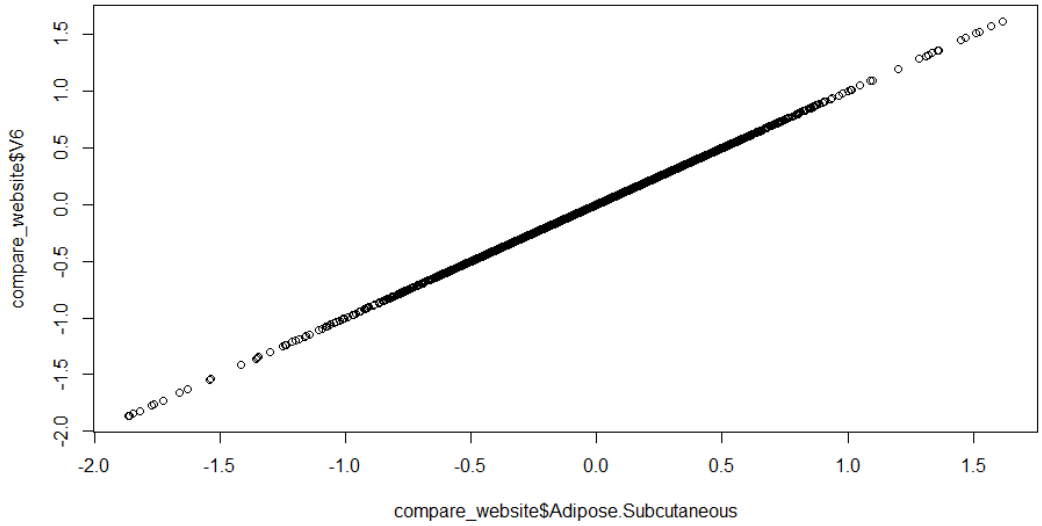


Figure 17. Observations on the Y axis versus predictions on the X axis.

6.3 Validation MPRA dataset

Next to the GTEx dataset, another available dataset with variants found by the Massively Parallel Reporter Assay (MPRA) technique, published in (53), will be used for validation in this thesis project. The paper describes a study on Genome-Wide Association Study (GWAS) variants in regulatory regions, that are likely to have an effect on gene expression. These variants are analysed by the MPRA technique and then, specific variants that had an altered gene expression as a result, were identified. This dataset is also not related to ALS, because there is no option with effects of variants in patients available. However, it is part of the Lymphoblastoid cell lines of subjects. The dataset "GSE75661_79k_collapsed_counts.txt.gz could be downloaded from the webpage https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75661.

The gene expression changes are measured in cells of different subjects. Only data from subject NA12878 is extracted, since predictions for the same subject are also available in the ExPecto models. The goal is to use the MPRA data as observations and compare this with the predictions of ExPecto and CADD. Methods and metrics to see how the observations correspond to the predictions, are correlations, an ROC curve and a confusion matrix.

The exact expression changes between two alleles in the Lymphoblastoid cell line of the subject have been reported. These changes will serve as validation for the models. If the MPRA dataset indicates a high expression change, there must also be predicted a high expression effect by ExPecto and CADD.

The limitation of this dataset, is the uncertainty of the expression change direction. The reason for this, is that it is unclear which one of the two SNP alleles is the reference and which one is the allele. For every SNP, there is at least one reference and an alternative allele, denoted as A and B. As is stated before: it is unclear what allele belongs to A or B. This dataset is about the general expression change between the two alleles.

Expression changes in five replicates of subject NA12878 are extracted from the original file. The experiment to measure changes is performed five times per allele. An example of this, can be found in table 7. Taking the average of these five replicates will have a more reliable effect than only using one column.

Allele	NA12878_r1	NA12878_r2	NA12878_r3	NA12878_r4	NA12878_r5
rs11548103_RC_A	2050	1786	2405	2538	2241
rs11548103_RC_B	2034	1885	2136	2630	2157
rs2016366_A	352	478	463	548	509
rs2016366_B	557	767	487	683	800

Table 7. MPRA expression changes per allele, done in five experiments (column).

Initially, this dataset consisted of 78.956 rows, which means that there were 39.478 allele pairs involved. After the pre-processing steps, which are elaborated upon in the next section, a final number of 27.138 rows was left. The goal was to create a file that can be used by ExPecto and CADD. In order to make this possible, the file must consist of five subsequent columns: chromosome (CHROM), position of the variant (POS), ID, reference allele (REF) and alternative allele (ALT). This is a simplified VCF file, since a VCF file could contain much more information about the variants, which is not necessary in this case.

After receiving the ExPecto and CADD results, they were analysed in Rstudio. It was found, that ExPecto produced many outcomes where the distance, the closest gene and the expression change were unknown. The exact reason for this, was not found. However, for the comparison between ExPecto and CADD on the MPRA data, it is important that the same variants are used. This is why the ExPecto and CADD results will only contain values without NA's (values that are empty/Not Available). This resulted in datasets of 10.235 rows for both tools.

6.3.1 ExPecto MPRA validation

The variants of the MPRA dataset are used as input for the ExPecto framework. ExPecto has predicted the corresponding gene expression changes for all rows. To show the performance of the model on the observation data, a ROC curve is used with its corresponding Area Under the Curve. First, the observation data (MPRA mean expression changes) has to be divided in two classes: non-disruptive values (0) and disruptive values (1). In the distribution of this column, visualised in Appendix 1, there is no clear distinction in the data for these two classes. The data is right skewed, with most of the data located on the left of the distribution and some high values on the right. This results in a substantial difference between the median (164) and the mean (270.9). The summary data of the MPRA mean can be found in table 8.

Min.	1st Quartile	Median	Mean	3rd Quartile	Max.
0	63.8	164	270.9	333	63925.4

Table 8. Summary data of the MPRA means column.

The mean of the MPRA column (270.9) is used to divide the variants into two classes: 0 and 1. This information is added as an extra column for the observation data. The ROC curve for the comparison of the MPRA observations and the ExPecto predictions, can be found in figure 18. The plotting of the specificity against the sensitivity resulted in an AUC of 0.546.

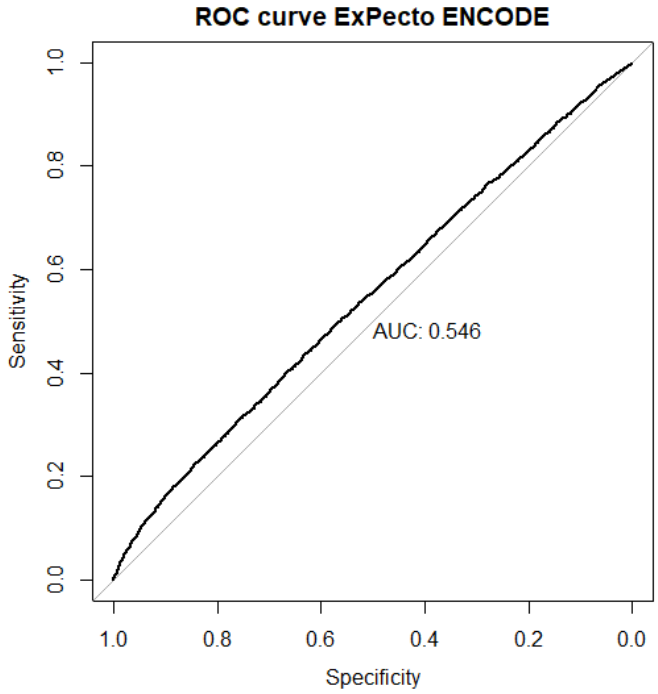


Figure 18. ROC curve of ExPecto ENCODE data in the MPRA analysis.

The best corresponding threshold for the ExPecto predictions, assigned by the function of the pROC library, is 0.001937568. Every score below this, is assigned to class 0 and everything above this, is assigned to class 1. The confusion matrix where predictions and observations are compared, can be found in the confusion matrix in table 9. The accuracy of ExPecto on the MPRA data was 61.6%.

	0	***1***
0	5298	2267
1	1663	1007

Table 9. Confusion matrix of the ExPecto MPRA analysis, with a predictions as rows and observations as columns.

6.3.2 CADD MPRA validation

The same 10.235 variants of the ExPecto MPRA validation, are used for the CADD validation. The mean of the MPRA column is used for dividing the variants of that column into two classes: 0 and 1. With these observation labels, a column is added that serves as input for the ROC curve. Specificity and Sensitivity of the MPRA data and the CADD scores are plotted in figure 19. The ROC curve results in an AUC of 0.505

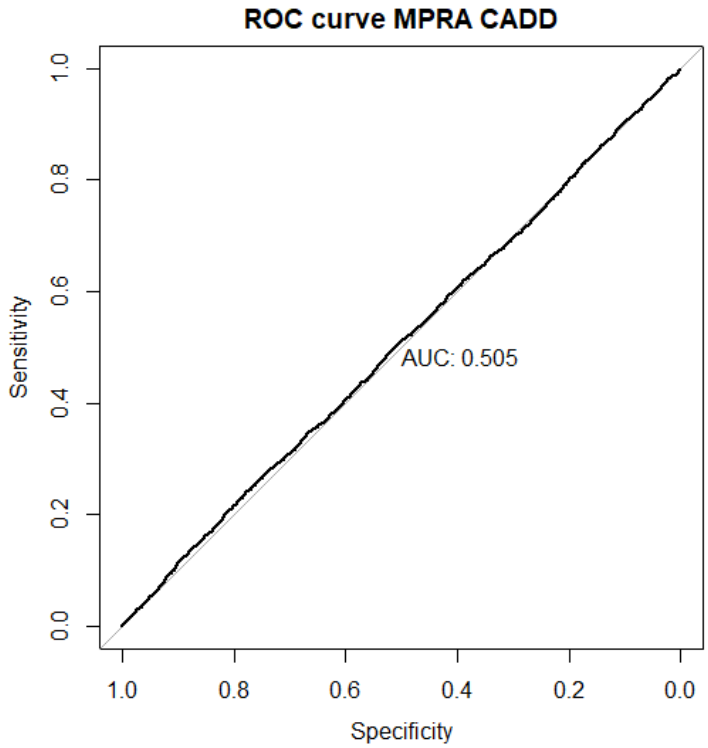


Figure 19. ROC curve of CADD data in the MPRA analysis.

The threshold of the CADD predictions, assigned by the function of the pROC library, is 0.937. Every score below this, is assigned to class 0 and everything above this, is assigned to class 1. The predictions and observations are compared in a confusion matrix (table 10). This resulted in an accuracy of 39.43%.

	0	***1***
0	1554	792
1	5407	2482

Table 10. Confusion matrix of the CADD MPRA analysis, with a predictions as rows and observations as columns.

With an AUC of 0.505 and an accuracy of 39.43%, the CADD framework has a rather disappointing performance on the MPRA data. The observation data that was used for this analysis, was retrieved from one cell-type: Lymphocytes. However, this could show that CADD is not able to make accurate predictions for one specific cell-type.

6.4 Conclusion of validation

ExPecto and CADD have been validated by using GTEx data and MPRA data. They are validated individually, but the performances have also been compared. The latter has been done by creating a threshold and extracting ROC curves.

When analysing the distance of ExPecto's prediction with the GTEx data, it is shown that there is more agreement between the two datasets, if the distance is smaller than 20.000 base pairs from the TSS. On top of that, ExPecto performs better on higher predictions. This could indicate that ExPecto annotates more random guesses to effects of variants it is not certain of.

Furthermore, the ROC curves show a bad performance of the models on the GTEx and MPRA observation data. The AUC's are all just above 0.5. When choosing the most optimal threshold, the ExPecto model has the best performance on the MPRA data, with an accuracy of 61.6%. A summary of all AUC- and accuracy-values can be found in table 11.

	GTEx		**MPRA**	
	CADD	**ExPecto**	**CADD**	**ExPecto**
AUC	0.518	0.522	0.505	0.546
Accuracy	55.15%	41.06%	39.43%	61.6%

Table 11. AUC and Accuracy for the CADD and ExPecto models on the validation datasets.

The validation on these two datasets serves as a quality check before using the models on the large MinE dataset. This MinE dataset contains the genetic variants of ALS patients and healthy controls. The interpretation of the gene expression changes of the variants in the GTEx dataset is more difficult to interpret than the MPRA variants. Because the GTEx variants are identified by GWAS, they could be linked to disruptive variants, but they do not have to be causal themselves. The MPRA variants, however, are more clear in terms of gene expression change. The gene expression with one allele is measured and then the gene expression resulting from another allele at the same spot is measured. For instance the expression of a gene with a G allele somewhere, which was changed to an A allele at the same spot. Then the change in gene expression was calculated. This could be seen as a more reliable validation method. ExPecto scores better on the MPRA dataset, compared to CADD.

7. Research execution

After the validation of the models, the actual research execution is initiated. This encompasses the running of algorithms on Project MinE data, which includes data of patients and controls. In May 2019, a new version of this data was released among the ALS group. This happens every once in a while, since a new batch of sample data is included. The research group strives for an increasing number of samples. This will make their analyses more reliable and additional conclusions could be drawn. For instance, new significant DNA regions of interest could pop up with a larger sample size.

The MinE dataset is divided into chromosomes 1 to 22 and sex chromosomes X and Y. For every chromosome, there is a file that contains all the variants that occurred in controls and/or patients combined. Every row contains information on the chromosome, position, id, reference allele, alternative allele and a quality score for the variant. Thereafter, a column is included for every control and patient, to see the specific allele of an individual. Information on quantity and individuals is not needed until after running the models, since the models only predict a certain effect of a variant. After predicting the effects for every variant, another step is taken to see if a variant is specific to patients or controls.

There are currently 158.644.898 variants in the MinE dataset. Samples have been collected from ALS patients and controls. Pre-processing steps were needed to make the 24 datasets suitable for the models. They are elaborated upon in chapter 7.1.

7.1 Pre-processing of MinE dataset and running of ExPecto scripts

The whole MinE dataset consists of a very large number of variants. This takes a lot of time to be analysed by the machine learning models. During the validation phase, it became clear that variants within a distance of 20.000 base pairs from the Transcription Start Site have more accurate ExPecto predictions than the ones further away. For the sake of uniformity and comparison, the choice is made to only take variants in 40 kb regions (20 kb upstream and 20 kb downstream the TSS) into account for both models (ExPecto and CADD). The benefits are a higher accuracy and also time saving in running the models.

To select the MinE variants in certain regions, the TSSs of genes must be known. Fortunately, the ExPecto directory possesses a geneanno.csv file. This has information on the ENSEMBL genes, their position and their Transcription Start Site. For the regions, the CAGE_representative_TSS is used, since this is more accurate.

1. In the command line, I used awk to get the third column (chromosome), the position 20.000 base pairs downstream of the TSS and the position 20.000 base pairs upstream of the TSS. The three outcome columns were added to a new BED file, called “geneanno.40kb.bed”. The first 10 rows of this dataset is shown in figure 20.

```
tail -n+2 geneanno.csv | awk -F "," '{print $3,$6-20000,$6+20000}' > geneanno.40kb.bed
```

```
(base) [mdegroot@hpcs03 resources]$ tail -n+2 geneanno.csv | awk -F "," '{print $3,$6-20000,$6+20000}' > geneanno.40kb.bed
(base) [mdegroot@hpcs03 resources]$ head geneanno.40kb.bed
chrX 99871748 99911748
chrX 99819933 99859933
chr20 49555069 49595069
chr1 169843037 169883037
chr1 169744186 169784186
chr1 27941654 27981654
chr1 196601174 196641174
chr6 143812857 143852857
chr6 53389899 53429899
chr6 41020714 41060714
```

Figure 20. First 10 rows of the geneanno.40kb.bed file.

2. Since we are looking at 40 kb regions, there is a chance that some of them are overlapping. It will be more efficient to get the “unique regions”. An option to look for overlapping regions is the package “Bedtools”. This can be installed in the Miniconda environment via the following way:

```
conda install -c bioconda bedtools
```

3. Basically, Bedtools will merge overlapping regions. A prerequisite to use it, is sorting the regions file on chromosome and position. By using the awk-, sort-, and sed-commands, the first two columns will be sorted from chromosome 1 to chromosome X.

```
tail -n+2 geneanno.csv | awk -F "," '{print $3,$6-20000,$6+20000}' |
sort -k1,1 -k2,2n | sed 's/ /\t/g' > geneanno.40kb.sorted.bed
```

4. Next, Bedtools was used to merge the regions and these were placed in the outcome BED file geneanno.uniqueb.sorted.bed. This file includes the chromosome and the start and end position of the region. It contains less rows than the 40 kb regions file, so there were overlapping regions. In total, the regions in the geneanno.uniqueb.sorted.bed file contains 745.749.470 base pairs, which is approximately 1/6 of the whole human genome.

```
~/miniconda3/bin/bedtools merge -i geneanno.40kb.sorted.bed >
geneanno.uniqueb.sorted.bed
```

5. After finding the regions where the focus is on, the corresponding project MinE variants must be found. This is done with another package suited for Python, called “Bcftools”. This was installed in the same Miniconda environment. To use the function, a VCF file is needed that contains all the variants of a chromosome, together with the regions file that was created in step 4. The function “view -R” is able to select variants in the specific regions. The first nine columns of the VCF file are selected and everything is placed in a compressed output file per chromosome. Only the first nine columns are selected, since the information on cases and controls is not needed to run the models. For chromosome 1, the example code is placed below.

```
~/miniconda3/bin/bcftools/bcftools view -R
/hpc/hers_en/mdegroot/ExPecto/resources/geneanno.uniqueb.sorted.bed
/hpc/hers_en/projectMinE2/2019-02-
16/output/filtered/gvcfgenotyper.9600.2019-02-
16.chr1.filt.norm.vcf.gz | cut -f1-9 | bgzip >
/hpc/hers_en/mdegroot/chromosomes/projectMinE.chr1.vcf.gz
```

6. For every chromosome VCF file, the variants of the selected regions were gathered. The number of variants in the original files can be found in table 12. The total number of project MinE variants was 158.644.898. Additionally, the number of variants after assigning the specific regions can be found in table 12. After assigning the regions, there was a total of 41.888.505 variants left in the chromosome files.

Chromosome	Number of variants in original file	Number of variants after specific regions	Number after splitting multi-allelic
1	12.220.358	4.095.131	4.567.306
2	13.206.321	3.049.493	3.372.139
3	10.870.449	2.405.960	2.662.761
4	10.595.453	2.103.637	2.322.263
5	9.824.965	2.248.508	2.480.313
6	9.295.027	2.153.376	2.382.865
7	8.847.546	2.031.255	2.260.455
8	8.461.379	1.911.995	2.108.966
9	6.615.291	1.672.027	1.859.120
10	7.387.298	1.792.438	1.995.297
11	7.469.654	2.436.264	2.693.873
12	7.273.499	2.331.682	2.592.328
13	5.307.604	947.498	1.044.695
14	4.934.522	1.414.794	1.576.896
15	4.525.380	1.324.015	1.478.861
16	5.149.003	1.442.087	1.617.720
17	4.506.232	1.752.080	1.972.797
18	4.190.539	571.731	636.853
19	3.617.450	2.241.506	2.569.036
20	3.420.313	1.170.925	1.306.443
21	2.158.585	644.893	718.285
22	2.108.752	920.224	1.037.209
X	6.317.911	1.195.723	1.482.918
Y	341.367	31.263	40.735
Total	**158.644.898**	**41.888.505**	**46.780.134**

Table 12. Number of variants in every chromosome file after three different steps.

7. The input for CADD and ExPecto cannot contain multi-allelic variants. However, they can still be found in the files. Multi-allelic variants are a way to present more than one alternative allele for a specific reference allele. For instance at reference allele C on position 10447 on chromosome 1, an A and a T were both found as alternative alleles in patients and control. This is saved in one row, so only a comma separates the A and T. This one row must be separated in two rows, so there is one for alternative allele A and one for alternative allele T. An example can be found in table 13. With the following code, the multi-allelic variants were splitted, which resulted in a row for every allele.

```
zcat
/hpc/hers_en/mdegroot/chromosomes/chr1/FINALprojectMinE.chr1.vcf.gz
| awk '{print $0"\t.\t.\t."}' | python
/hpc/hers_en/kkenna/lib/process_gvcf/scripts/parseMultiAllelic.py -
| cut -f1-5 | bgzip -c >
/hpc/hers_en/mdegroot/chromosomes/chr1/split.FINALprojectMinE.chr1.v
cf.gz
```

Chromosome	Position	ID	Reference	Alternative
chr1	10447	chr1:10447:C:A,T	C	A, T

Chromosome	Position	ID	Reference	Alternative
chr1	10447	chr1:10447:C:A	C	A
chr1	10447	chr1:10447:C:T	C	T

Table 13. The multi-allelic variants and the variants after splitting.

8. After step 7, there is a VCF file for every variant. However, these contain millions of variants, which will take a long time to run. To do this faster, the VCF files are split into files of 100.000 variants. Here is an example that was executed in the terminal for chromosome 1:

```
zcat
/hpc/hers_en/mdegroot/chromosomes/chr1/split.FINALprojectMinE.chr1.v
cf.gz | awk
'NR%100000==1{out="/hpc/hers_en/mdegroot/chromosomes/chr1/analysis.s
plit.FINALprojectMinE.chr1."++i".vcf"}{print > out}'
```

9. After splitting the chromosome files in smaller parts, the chromatin.py script was used to make predictions for more than 2000 chromatin features. The following job was therefore initiated on the HPC environment of the UMC. This is an example for chromosome 19, which consists of 26 separate files. A total of 432 hours and 35 Gigabytes were added to run the script. This was used on the CPU's, but other parts were done via the GPU's. By dividing the running of the data on both options, the running of scripts went a lot quicker.

```
echo -e "cd /hpc/hers_en/mdegroot/ExPecto;
/hpc/hers_en/mdegroot/miniconda3/bin/python chromatin.py
/hpc/hers_en/mdegroot/chromosomes/chr19/analysis.split.FINALprojectM
inE.chr19.\${SGE_TASK_ID}.vcf.gz" | qsub -l "h_rt=432:00:00" -l
"h_vmem=35G" -N Chr19ExPecto -o
/hpc/hers_en/mdegroot/chromosomes/chr19/ -e
/hpc/hers_en/mdegroot/chromosomes/chr19/ -t 1:26
```

10. Next to the chromatin models of step 9, there is another input for the final prediction step 11: the closest genes. For every variant in the VCF files of the chromosomes, there is a closest gene. This means the closest Transcription Start Site of a gene to a specific variant. The ExPecto Github directory contained a file (geneanno.pc.sorted.bed) with genes, their position on the DNA sequence and their TSS's. This step is necessary, because then it will be known on what gene the variant has an influence. With the following code, a calculation is made for the variants of chromosome X:

```
echo -e "cd /hpc/hers_en/mdegroot/ExPecto;
/hpc/hers_en/mdegroot/miniconda3/bin/closest-features --delim '\t' -
-closest --dist <(zcat
/hpc/hers_en/mdegroot/chromosomes/chrX/analysis.split.FINALprojectMi
nE.chrX.\${SGE_TASK_ID}.vcf.gz | awk '{printf \$1\"\\\t\"\$2-
1\"\\\t\"\$2\"\\\n\"}' | /hpc/hers_en/mdegroot/miniconda3/bin/sort-
bed - )
/hpc/hers_en/mdegroot/ExPecto/resources/geneanno.pc.sorted.bed >
/hpc/hers_en/mdegroot/chromosomes/chrX/analysis.split.FINALprojectMi
nE.chrX.\${SGE_TASK_ID}.vcf.bed.sorted.bed.closestgene" | qsub -l
```

```
"h_rt=00:05:00" -l "h_vmem=0.1G" -N ChrXClosestGene -o
/hpc/hers_en/mdegroot/chromosomes/chrX/ -e
/hpc/hers_en/mdegroot/chromosomes/chrX/ -t 1:15
```

11. After making the closest gene file as another input for the predict.py script, the final predictions can be generated by regularized linear regression models. These models calculate the exact gene expression for every variant in over 200 different cell types and tissues. The input files are the VCF files with the variants. In this example, there are 34 separate VCF files for chromosome 2. Secondly, there are closest gene files for alle these 34 files. Lastly, the calculated chromatin models of step 9 are added and also the cell type and tissue specific models. The output is a collection of 34 CSV files with a gene expression prediction for all variants of that specific chromosome.

```
echo -e "cd /hpc/hers_en/mdegroot/ExPecto;
/hpc/hers_en/mdegroot/miniconda3/bin/python predict.py --coorFile
/hpc/hers_en/mdegroot/chromosomes/chr2/analysis.split.FINALprojectMi
nE.chr2.\${SGE_TASK_ID}.vcf.gz --geneFile
/hpc/hers_en/mdegroot/chromosomes/chr2/analysis.split.FINALprojectMi
nE.chr2.\${SGE_TASK_ID}.vcf.bed.sorted.bed.closestgene --
snpEffectFilePattern
/hpc/hers_en/mdegroot/chromosomes/chr2/analysis.split.FINALprojectMi
nE.chr2.\${SGE_TASK_ID}.vcf.gz.shift_SHIFT.diff.h5 --modelList
/hpc/hers_en/mdegroot/ExPecto/resources/modellist --output
/hpc/hers_en/mdegroot/chromosomes/chr2/output_chr2.\${SGE_TASK_ID}.c
sv" | qsub -l "h_rt=24:00:00" -l "h_vmem=10G" -N chr2Predict -o
/hpc/hers_en/mdegroot/chromosomes/chr2/ -e
/hpc/hers_en/mdegroot/chromosomes/chr2/ -t 1:34
```

7.2 CADD predictions for variants

The second machine learning tool (CADD) does not have a Github directory to clone and to use in the command line. Instead, CADD has a website where a VCF file can be uploaded. After a few hours, the CADD predictions are made for all variants in the file and can be downloaded from this same website. The ALS neurogenetics group of the UMC already had the CADD scores for Single Nucleotide variants (SNV's), but not yet for the indels. This was explained in chapter 5.2. The scores for SNV's had to be pulled from the UMC database, but the scores for the indels still had to be generated. The following steps were taken to get the CADD scores for indels:

1. First, the indels were extracted from every chromosome VCF file. Column 4 and 5 of the VCF file are checked on the number of letters (alleles) and if they are not equal to one. Column 4 contains the reference allele and column 5 contains the alternative allele. For instance, if column 4 contains the reference sequence TCC and the corresponding alternative sequence contains a T, then it is considered to be an indel. The following code was used to extract indels for chromosome Y:

```
zcat
/hpc/hers_en/mdegroot/chromosomes/chrY/split.FINALprojectMinE.chrY.v
cf.gz | awk 'length($4)!=1 || length($5)!=1' >
/hpc/hers_en/mdegroot/CADD/indel.CADD.chrY.vcf
```

2. Next, the indels for all other chromosomes were extracted and the rows were counted. The number of indels per chromosome can be found in table 14.

chromosome	indels	Files 100.000
1	926.121	10
2	655.794	7
3	526.751	6
4	440.881	5
5	472.832	5
6	473.832	5
7	455.941	5
8	384.929	4
9	363.235	4
10	402.436	5
11	513.245	6
12	535.516	6
13	201.606	3
14	319.758	4
15	305.800	4
16	324.583	4
17	437.421	5
18	132.199	2
19	618.550	7
20	267.445	3
21	145.045	2
22	222.221	3
X	328.535	4
Y	6.913	1
Total:	9.461.589	95

Table 14. Number of indels and number of files per chromosome.

3. Furthermore, the indel files were separated in files of 100.000 indels (variants). This step was taken, because the CADD website takes files with a maximum of 100.000 variants. The following code was used in the command line:

```
zcat /hpc/hers_en/mdegroot/CADD/indel.CADD.chrY.vcf.gz | awk
'NR%100000==1{out="/hpc/hers_en/mdegroot/CADD/indel.CADD.chrY."++i".
vcf"}{print > out}'
```

4. All 95 files were uploaded separately to the CADD website, so that the predictions could be made for all variants.

8. Data analysis

In this chapter, the results of ExPecto and CADD are shown and analysed. ExPecto was used to predict gene expression changes for over 200 tissues and cell types. However, for this project, we focused further analysis on four cell-types. Three of them have been proven to be involved in the development of ALS: the frontal cortex (part of the brain), the spinal cord and stem cell derived motor neurons. As a negative control we also included a cell type that has not been implicated in the development of ALS: adipose subcutaneous (fat cells). The objective was to explore the impact of cell types on variant predictions. Furthermore, the ExPecto scores were also compared to the CADD scores. CADD does not have the ability to make a distinction between cell types, so the general scores were used.

Our objective was to identify DNA variants with strong effects on gene regulation. Therefore only the highest predictions of both tools were important for this analysis (further explained in chapter 8.2 about gene burden testing). By only importing the highest and most relevant scores in the database, the runtime was reduced significantly. The top 1% of prediction scores was taken into account. For all five files (ExPecto predictions across 4 cell types and 1 non-cell specific prediction by CADD), a threshold was calculated in R to find the top 1% of variants with the highest prediction values. These thresholds can be found in table 15. Variants above these threshold values were included in further analysis and imported into the database, because those variants have the highest ExPecto or CADD scores of the whole datasets. Variants below these threshold values were not included in further analysis. For instance, it does not make sense to look at a variant that has an ExPecto score of 0.000001, which indicates a very small gene expression change. The difference in threshold scores of ExPecto (which are centered around zero) and the scores of CADD (threshold is higher than 20) is explained in chapter 3.3.5. The models have different ranges in their output predictions, this is why a CADD score can be much higher.

FILE	THRESHOLD TOP 1% PREDICTIONS
MOTOR NEURONS	0.08142306
FRONTAL CORTEX	0.05139208
SPINAL CORD	0.05128547
ADIPOSE SUBCUTANEOUS	0.0710034
CADD	21.100

Table 15. Thresholds of the ExPecto and CADD predictions, to make a separated set as input for the database.

8.1 Preparation of the database

In order to analyse the predictions of ExPecto and CADD, information had to be gathered in a database. In this environment, which was set up via the command line, several tables were included. A visualisation of the database with its tables can be found in figure 21. At first, the tables for the ExPecto predictions of the four different cell types were included: Motor Neurons, Frontal Cortex, Spinal Cord and Adipose Subcutaneous. Another file that was added as a table, is the CADD score file. In total, there were five tables with variants and their predicted gene expression changes. A “closest genes” file was added to the database which describes which genes are impacted by a given DNA variant. Information on which samples carry a given DNA variant can be found in table “dosage”. On top of that, the table “cohort” provides information on what sample is a patient and what sample is a healthy control. Lastly, the table “var” contained meta data for all variants combined, including the chromosome, position, reference allele and alternative allele.

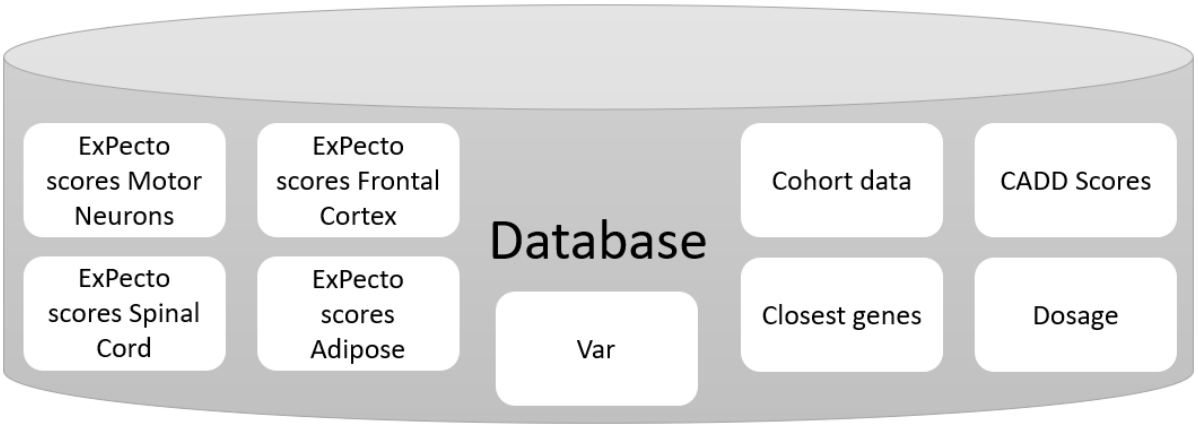


Figure 21. Database with its tables necessary for the data analysis.

8.2 Gene burden testing

After organising all the tables in the database, a gene burden test was carried out. The idea of this test, is to find genes in patients that significantly differ from genes in controls. The steps that were carried out, are shown in figure 22. Variants that were linked to a gene, were counted per case and per control. As an example in the figure below, both cases and controls have fourteen variants that are linked to gene 1. The rows of variants in figure 22 are not a genomic sequence, but they were found in different spots in and around the gene. This is not a row of random alleles that was found in this order. It is just an example of variants that were close to a gene, and thus linked to this gene. All the letters are variants and they differed from the reference sequence.

All the variants in of the MinE analysis (40 kb regions) have a certain prediction score that was saved in one of the five tables (ExPecto scores or CADD score). For this analysis, only the variants with high scores were relevant, since they are most likely to have a disruptive effect. Four disruptive variants in cases are identified in the example (yellow marks). Only one disruptive variant was identified in controls (yellow mark). These identified variants have high prediction scores and all the other ones have low prediction scores. The last letter of the row of variants is different, but these did not have a large expression effect change in both situations, so they are not highlighted. The outcome is a p-value for a certain gene, which was calculated after comparing cases and controls.

	Identify all variants that belong to a gene	Identify variants with high prediction score	Statistical comparison
	Gene 1	Gene 1	
Case	AATAGATTTCAAAT	AATAGATTTCAAAT	AATA
	Gene 1	Gene 1	Vs.
Control	ACTTGATTTCCAAC	ACTTGATTTCCAAC	T

Figure 22. Gene burden testing.

The commands that were used for the gene burden tests, can be found in Appendix 3. With the outcome files of the gene burden test, that contain the genes with their burden p-value, visualisations were made in R to gain more insights about significance and quality of the analysis.

8.3 Results

The five outcome files of the gene burden tests still contained thousands of p-values. Manhattan plots were created in R, to help visualise and interpret these result. The Manhattan plot is a classic way to show significant variants in the genetics research field. The name of this plot is chosen, since it resembles the skyline with high buildings, which are the variants that are significant. The x-axis is organised by position in the genome and contains chromosomes 1 to 22. The y-axis indicates the -Log10 of the p-value. In GWAS, a variant in the plot must be linked to a gene afterwards, which is very often a difficult task. For the gene burden testing performed in this thesis research, the variants were already linked to a gene. Every single dot in the Manhattan plots in this chapter indicates a certain gene. Because of the log-transformation, the small p-values become the highest values and are easier to detect in the plot. The threshold for significance within a given analysis is $2.5x10^{-6}$ (bonferronni multiple testing correction for analysis of up to 20K genes).

The code for pre-processing of data can be found in Appendix 4. The plots in this chapter were all created with the R language. The pre-processing and plotting required four different libraries: rvat, ggplot2, dplyr and sqldf. The gene burden files were loaded into R. Then, the header names were changed and added for further analysis and more clarity. Furthermore, the gene file was uploaded into R, since information about the genes (chromosome, start position, end position, strand and the ENSEMBL gene names) had to be added. The gene burden file was merged with the gene file with an sqldf-command. A Manhattan plot was created for ExPecto results on the four cell-types and for the CADD results.

Manhattan plots for ExPecto results on cell-types Motor Neurons, Adipose Subcutaneous, Frontal Cortex and Spinal Cord can be found in figures 23, 24, 25 and 26. No genes with significant p-values have been found in these cell-types. All genes of the ExPecto cell types were on or under the border of value 3 on the y-axis, except for one gene in the plot of the Frontal Cortex. The protein-coding gene PPP1R1C (ENSEMBL gene id: ENSG00000150722) on chromosome 2 has the lowest p-value of 0.0002965711. This gene has an altered expression in the Frontal Cortex, which is the result of disruptive genetic variants. A disruptive variant (with a high ExPecto score) linked to this gene, was found in 6067 cases versus 2294 controls.

The p-values in the Manhattan plot for CADD (figure 27) were also not significant and almost all of them were under a 3 on the y-axis. There was one gene that was just above the value 3, but this p-value was not lower (not more towards significance) than the best ExPecto gene. On thop of that, it was also not the same gene, since the one from CADD is located on chromosome 10.

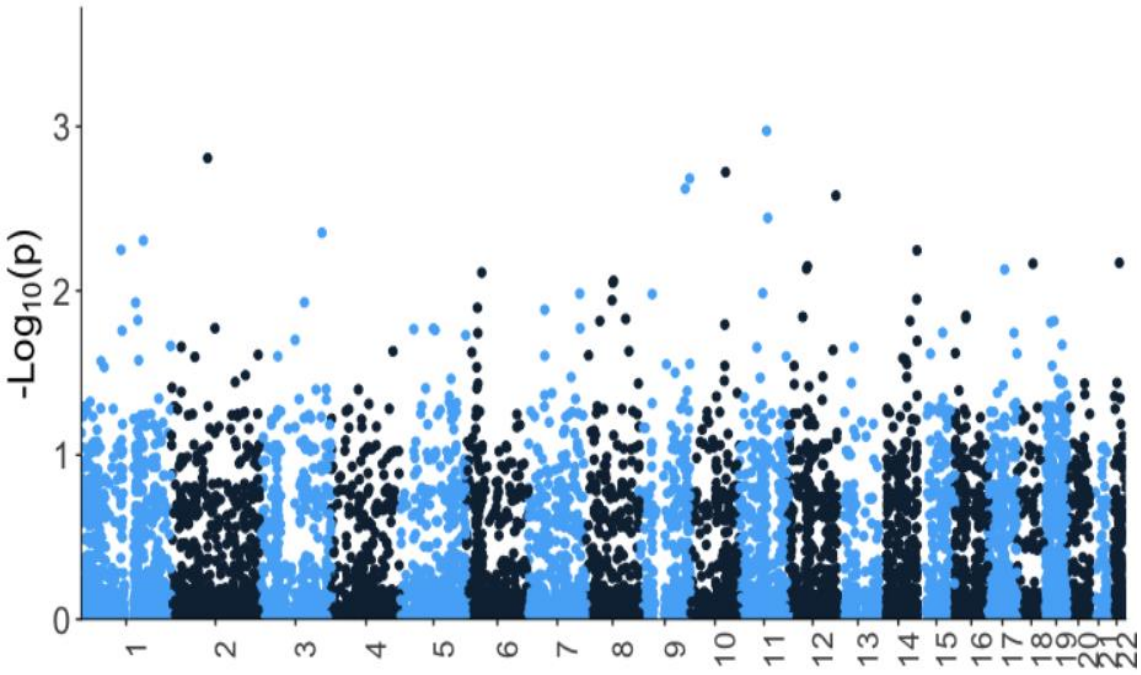


Figure 23. Manhattan plot of ExPecto predictions for Motor Neurons.

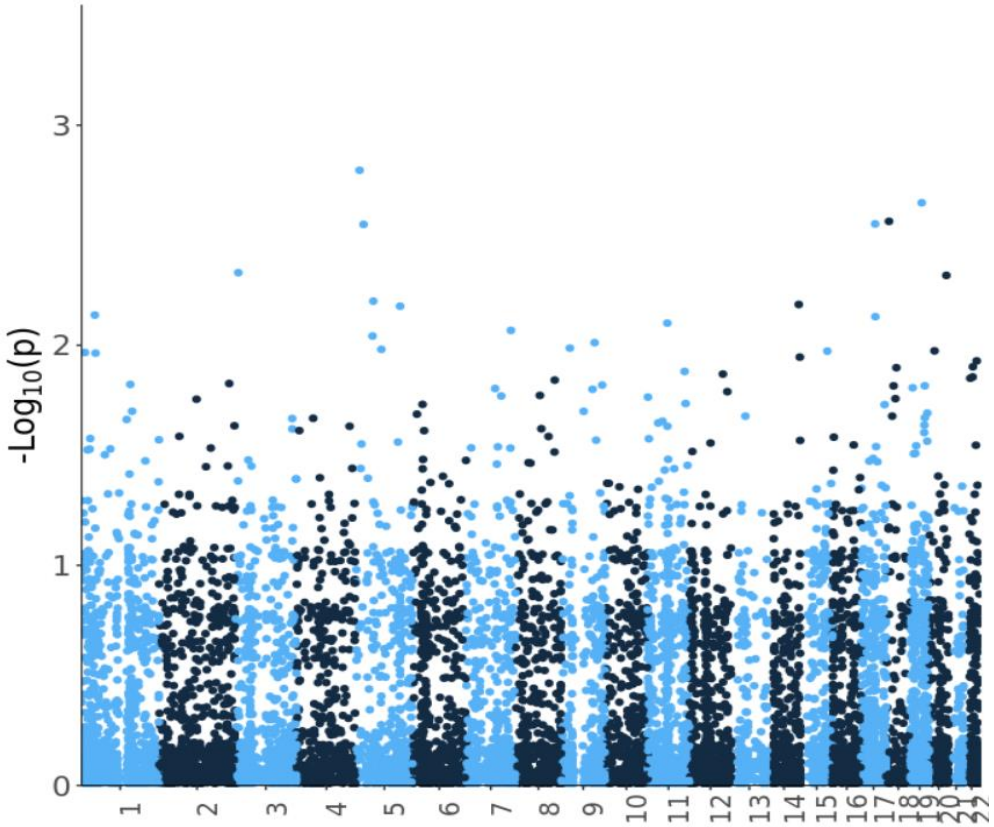


Figure 24. Manhattan plot of ExPecto predictions for Adipose Subcutaneous.

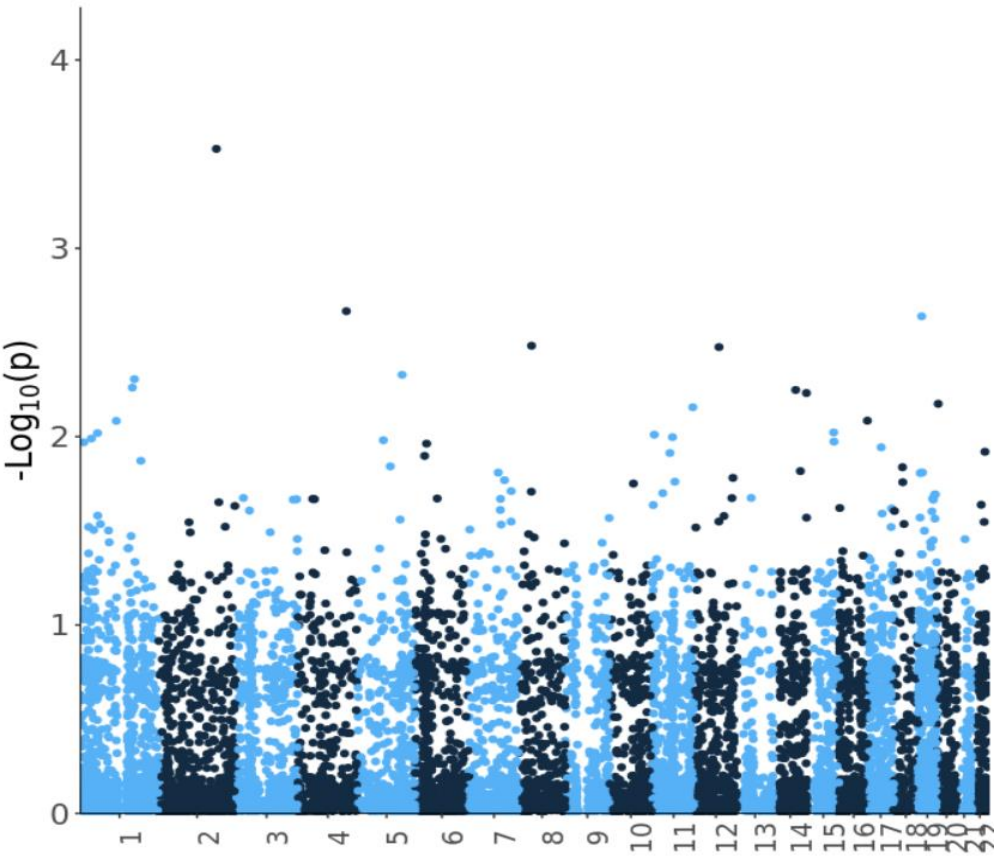


Figure 25. Manhattan plot of ExPecto predictions for Frontal Cortex.

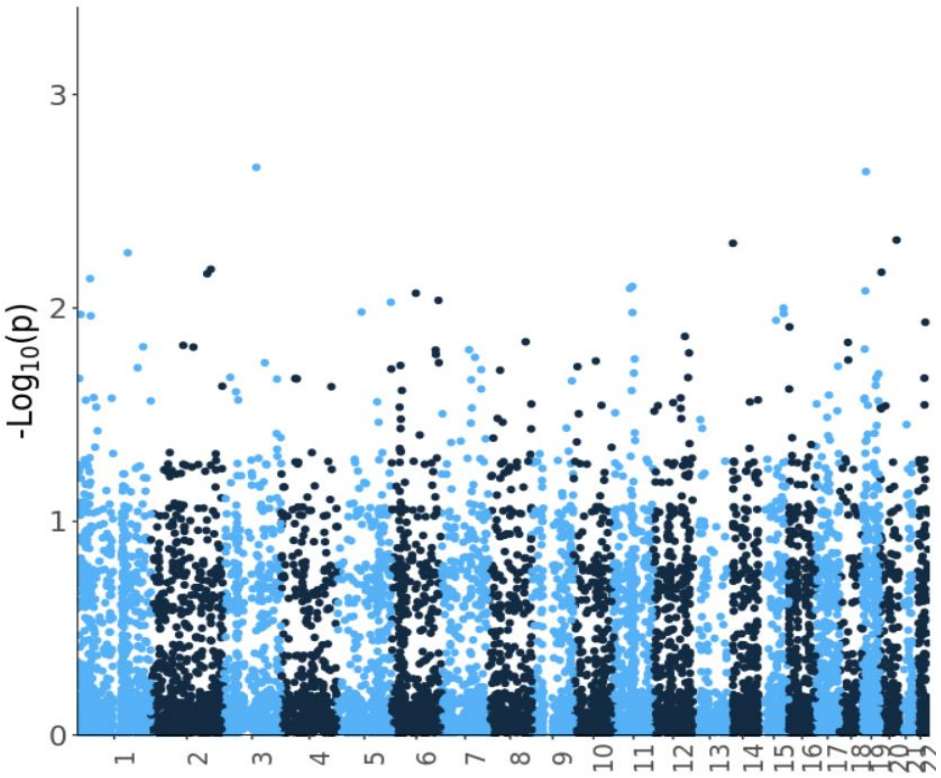


Figure 26. Manhattan plot of ExPecto predictions for Spinal Cord.

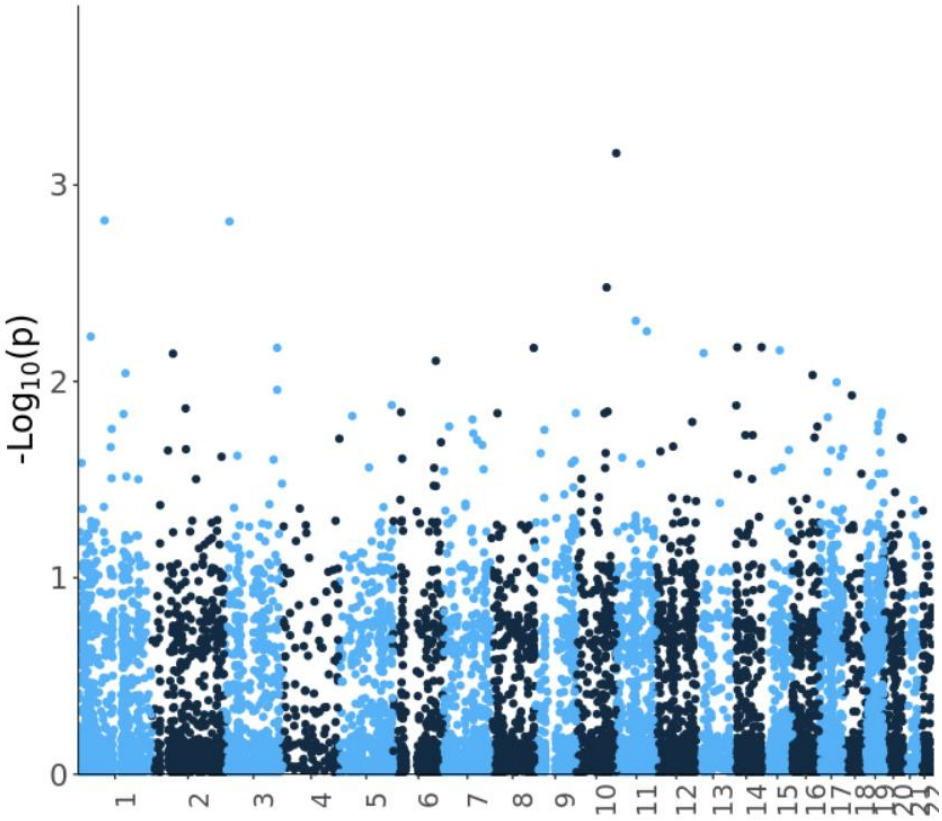


Figure 27. Manhattan plot of CADD scores.

Next to the Manhattan plot, there is another visualisation method that gives information about the generated p-values: the quantile-quantile plot (QQ-plot). This is a visual representation of the quality of the outcome data, which needs to follow a certain desired distribution. Two substantial elements are needed for such a plot: the observed p-values and the values sampled from the theoretical null (uniform) distribution. The observed p-values are ordered and for every value, there is a corresponding distribution value. If the observed values are distributed exactly the same as the expected values, the dots will follow the diagonal line. However, some values may be more or less significant than the expected ones, so these dots are deviated from the line. The Lambda gives an additional indication of the quality of the data. A Lambda value close to 1 means that the data follows the distribution (54). High lambda values could reflect confounding of the gene burden test analyses by technical artefacts (DNA sequencing differences in cases vs controls) or biological stratification (differences in ancestry of cases vs controls).

Figures 28-31 visualise the QQ-plots for the four cell-types of ExPecto predictions. All four plots show that the observed p-values reasonably fit the distribution of the expected values.. The motor neurons have the highest Lambda value of the ExPecto cell types: 0.98. This cell-type is followed by the Frontal Cortex, with a Lambda value of 0.94. The Adipose Subcutaneous and the Spinal Cord both have a Lambda of 0.9. These QQ-plots are a way to check the quality of the predictions, given by the Machine Learning tools. The lambda values of the ExPecto cell types show that the p-values for the genes follow the distribution and that the majority has high p-values that do not stand out. Only a small part of the data has lower p-values, which is how it is supposed to be. Figure 32 visualises the QQ-plot for the CADD scores, with a Lambda of 1.08. This could indicate a mistake in technical artefacts of the sequencing or biological stratification, in an early stage of the data retrieval. There are more high p-values in the CADD data table than expected.

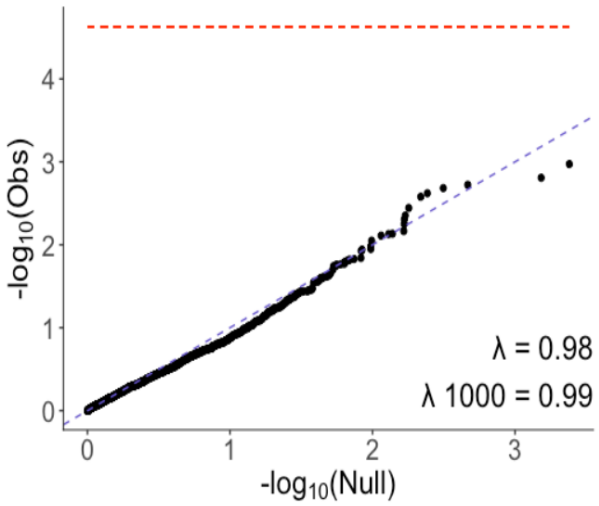


Figure 28. QQ-plot of Motor Neurons.

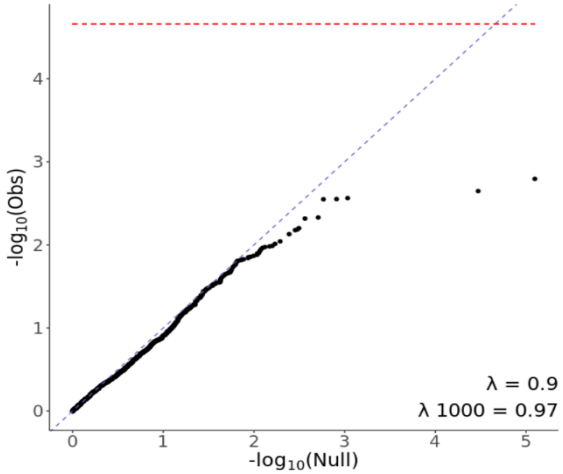


Figure 29. QQ-plot of Adipose Subcutaneous.

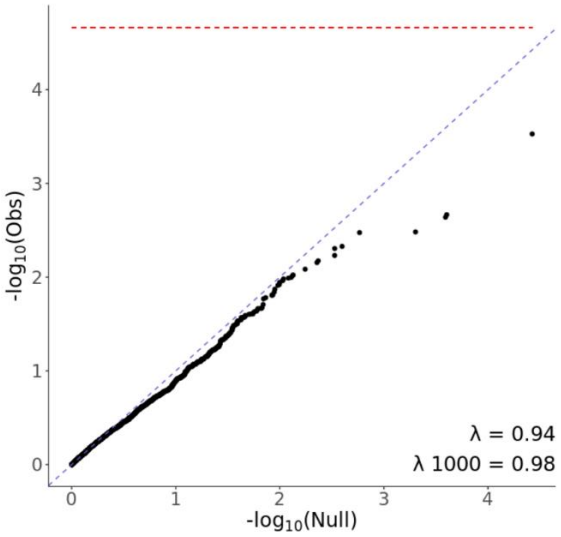


Figure 30. QQ-plot of Frontal cortex.

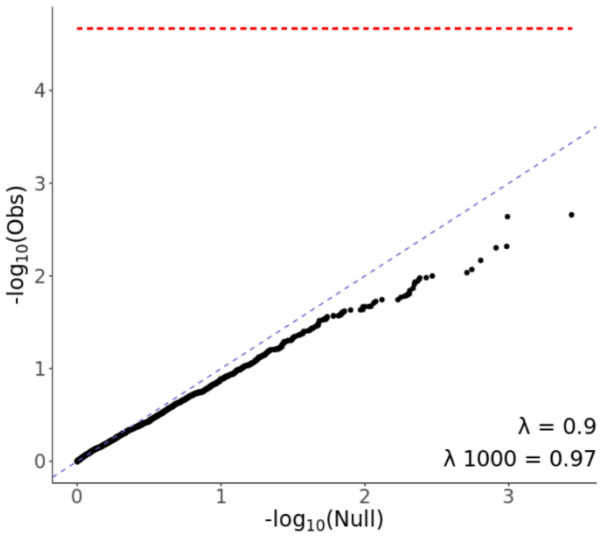


Figure 31. QQ-plot of Spinal Cord.

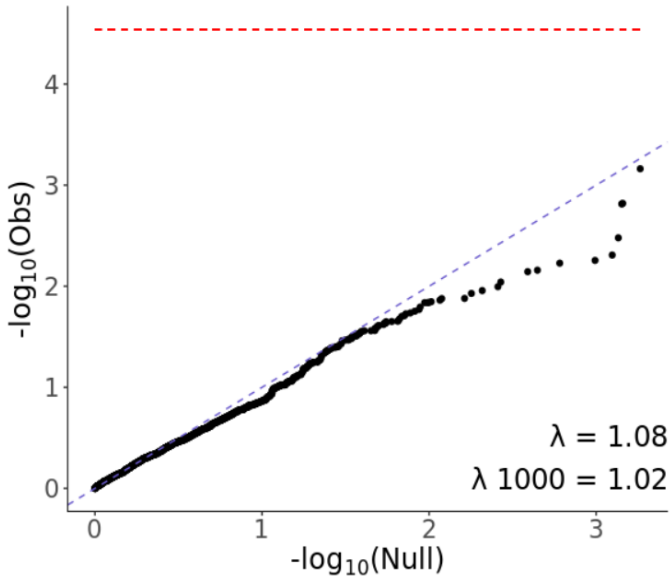


Figure 32. QQ-plot of CADD scores.

9. Conclusion

The goal of this research was to find the most suitable machine learning algorithm that could predict disruptive activities of variants in regulatory DNA regions of ALS patients. The following research question was chosen for this thesis:

How can machine learning algorithms predict the way that genetic variants disrupt the activity of regulatory DNA sequences in ALS patients?

To answer this research question, there were five sub questions formulated.

SQ1. How are genetic variants able to disrupt the activity of regulatory DNA sequences in ALS patients?

Advanced techniques make the sequencing of DNA of individuals possible. The whole string consists of a long combination of the four letters A, C, T and G, that indicate the nucleotides. They are also called “alleles”. Every individual has his or her own unique DNA sequence. A large part of the sequence is identical in all humans, but a small part contains genetic variants. These variations in the DNA, could cause diseases like ALS, since they have an effect on the expression of genes. The genes are locations in the DNA that are a blueprint for the production of proteins. Other regions, called “regulatory regions”, can have an indirect effect on this process, since they regulate genes from a small or a large distance. It is important to detect variants in these regulatory regions, since they could have a disruptive effect on the human body.

SQ2. What scientific research has been done on predicting effects of genetic variants in regulatory regions?

ALS is a disease that is caused by genetic faults, which means in this case: a combination of many variants. In ALS research, the focus has been on variants in the protein coding gene regions for a long time, but answers could also be found in the regulatory regions. On top of that, research was focussed on finding common variants (that occur in more than 5% of the population), while ALS is a combination of common and rare variants. These rare variants are more difficult to find, since they occur in less than 1% of the population. A way to find these variants, is to look at the effect that a certain variant has on the gene expression, to see how disruptive it is. Eventually the effects of variants found in ALS patients, can be compared to the effects of variants found in controls.

SQ3. What ML tools are available to address this problem (predicting the expression effects of variants in regulatory regions) and how could they be compared?

Machine learning techniques that address these problems have been developed over time. These algorithms have the ability to find complex patterns in large DNA datasets. The datasets that are available for this thesis study, are files with genetic variants of ALS patients and controls, retrieved by the Project MinE. The first machine learning framework that was chosen for this study, is Combined Annotation Dependent Depletion (CADD), which is a tool that was first introduced in 2014. It has been the state of the art since than for calculating a (PHRED-scaled) score for variants, which gives an indication for the level of disruption. It uses logistic regression and it is often used as a comparison for other tools that have the same aim. The second tool is the machine learning framework ExPecto, that was more recently introduced in 2018. This makes use of a convolutional neural network in combination with linear regression and it calculates the log2 gene expression change of a genetic variant. In contrast to CADD, ExPecto is able to make tissue-specific predictions. Both tools are able to analyse regulatory regions, to analyse common and rare variants and to take genetic variants as input for the algorithms.

The two frameworks are already trained and validated, but for this thesis research, an extra validation step was added. The tools are validated on Lymphocytes datasets of the GTEx project and an MPRA project. In both datasets, gene expression changes are known for every genetic variant. The quality of the models individually and compared to each other, can be decided if their predictions are validated with the observation data. In order to make ROC curves, the predictions and observations had to be divided into to classes: high effect variants (1) and low effect variants (0). The results of the quality analysis can be found in table 16. In the GTEx analysis, ExPecto performs slightly better on the AUC, but CADD performs better in terms of the accuracy. In the MPRA analysis, ExPecto performs better on the AUC and on the accuracy. However, all performances were disappointing, since models with an AUC of approximately 0.5, are hardly able to discriminate between the positive and the negative classes.

In the GTEx analysis, another individual validation step was taken for the ExPecto model. The GTEx slope was negative or positive, indicating the direction of the change of gene expression (increasing or decreasing). ExPecto also has this ability, so the observational directions could be compared with the predicted directions. A conclusion of this analysis, was that ExPecto could predict the direction of expression change better in regions of 20.000 base pairs (20 kb) from the Transcription Start Site of a gene. This is also what the ExPecto framework was designed for. The decision was made to work with 40 kb regions around a TSS for the large project MinE analysis, since this results in more accurate predictions.

	GTEx		**MPRA**	
	CADD	**ExPecto**	**CADD**	**ExPecto**
AUC	0.518	0.522	0.505	0.546
Accuracy	55.15%	41.06%	39.43%	61.6%

Table 16. AUC and Accuracy for GTEx and MPRA analyses per machine learning framework.

SQ4. What kind of data and pre-processing steps are required by the tools?

This sub question revolves around the pre-processing of the Project MinE data. DNA profiles of ALS patients and healthy controls have been gathered. These resulted in large files with genetic variants. For every chromosome, there is a VCF file with information about the variants, their position, alleles and other characteristics. With the knowledge of sub question 3, it was decided to only take variants within regions of 40 kb around the TSS's of genes. Subsequently, the multi-allelic variants had to be split up. The machine learning tool both used simplified VCF files as an input. However, the large chromosome files with millions of rows had to be divided in smaller parts of 100.000 variants.

SQ5. What predictions on expression effects are made by the machine learning tools?

The frameworks CADD and ExPecto were used to make predictions on 46 million variants in 40 kb regions around the TSS, that were found in patients and controls in Project MinE. This resulted in a CADD score and an ExPecto log2 expression change for every variant. The next step was to do an analysis to compare effects of variants between patients and controls. This was done by a gene burden test, which is a way to find genes that were significantly different expressed in patients, compared to controls. First, all variants that were close to a specific gene were gathered. Then, only the variants with high predictions (high chance of being disruptive) were important for the further analysis. The last step was to calculate a p-value for every gene, by looking at the difference between variants with a high prediction value in patients and controls. Manhattan plots showed no significant genes for four ExPecto cell-type predictions and CADD predictions. The lowest p-value was found in the plot for the ExPecto scores in the Frontal Cortex. However, the p-value of this gene did not reach significance and it has not been mentioned in scientific research to be associated to ALS.

Research Question

It was shown that there exist trained Machine Learning frameworks that are able to make a certain prediction on the effects of variants on gene expression. CADD and ExPecto are designed to identify disruptive variants, while taking common and rare variants into account. Additionally, they are both able to predict the effect of variants in regulatory regions, instead of exclusively looking into protein coding regions.

The validation phase showed the predictive performance on the MPRA dataset and the GTEx dataset. In both datasets, there were variants included with their labelled gene expression effects, so these could serve as a validation method for CADD and ExPecto. ROC curves made clear that the two Machine Learning tools were hardly able to distinguish disruptive variants (class 1) from non-disruptive variants (class 0). The Areas Under the Curves for both validation datasets were just above 0.5, which means that they are close to the performance of a model that gives random predictions. However, ExPecto was able to show an overall better performance, compared to CADD.

The GTEx dataset provided an extra validation method for ExPecto besides the gene expression, which was the direction of gene expression (up or down). After looking into GTEx variants within 40 kb regions around the TSS, it was shown that ExPecto had an agreement with GTEx of 63% of the variants for ExPecto scores above 0.1. Furthermore, ExPecto had an agreement with GTEx of 79% for ExPecto scores above 0.4. The predictive performance increased with higher ExPecto scores. Unfortunately, CADD is not able to provide information on the direction of gene expression, so the two tools could not be compared on this metric.

In the ALS data analysis with variants of Project MinE, there were no genes identified that were significantly different expressed in ALS patients, compared to healthy controls. In earlier research, there were genes identified that play a role in the development of ALS. These did not pop up in the large analysis of this thesis. However, if one of these genes had turned out to be significant, than it would have been another validation step to see that one of the Machine Learning tools (or both tools) was able to perform well.

10. Discussion

A remarkable result was the bad performance of CADD, since this framework has been the state-of-the-art for years. At first, the validation phase was showing AUC's of 0.518 for the GTEx analysis and 0.505 for the MPRA analysis. Secondly, the CADD Manhattan plot for the MinE data didn't show any signs of significance. A note has to be made, that the CADD framework is supposed to be for the whole DNA sequence. ExPecto has the limitation of taking only 40 kb regions around the TSS of a gene into account, so the decision was made to do the whole MinE analysis for this thesis on variants in these regions. Supposedly, it is not clear if CADD would have given other outcomes in the gene burden test, if the whole genome was taken into account. On top of this, the CADD framework produces general scores and is not made to be tissue-specific nor cell-type-specific. However, in the MPRA and the GTEx analyses, data was retrieved from a specific cell-type. The gene expression changes for a genetic variant, were only meant for this specific cell-type. This might show that CADD is not suited for predicting scores for one cell-type, instead of general effects for a variant.

As has been mentioned, CADD and ExPecto did not perform well in the validation phase. A difficulty in the creation of the ROC curves, was deciding about the threshold of the two classes. These two classes divide variants in a non-disruptive group (0) and a disruptive group (1). There was no clear deviation in the distribution of the GTEx or MPRA observations. However, the data was skewed right, so the mean was taken as a threshold. The idea behind it, was that there were fewer observations

above this threshold, since there are fewer disruptive variants in the data than non-disruptive variants. However, when the threshold of the GTEx slope and the MPRA mean of gene expressions were scaled up, it was proven that the performance of CADD and ExPecto improved. For instance, there was a slight improvement in the AUC. This shows that different thresholds show different performances. Although the differences are small, this should be taken into account.

The performance of CADD and ExPecto on the MinE dataset, only accounts for 40 kb regions around the Transcription Start Site of a gene. This means that a large part of the genome has been left out of the analysis, while this part could also be influencing expression effects. Some regulatory elements, like enhancers, are located thousands of base pairs away from the gene. A genetic variant in such a region, could disrupt gene expression from a large distance, further than 40.000 base pairs away. However, expanding the regions could also cause trouble, since more noise is added to the data. On top of that, the ExPecto framework was not designed to predict effects for variants in larger regions, so it would have resulted in non-reliable output.

11. Future research

This thesis study shows that there still exists a gap in research, in terms of finding the exact disruptive effects of genetic variants in regulatory regions. However, new tools that aim to solve this problem, are added every year. In the ALS research field, it is important to keep track of these tools, to see their advantages and their shortcomings. The UMC Utrecht ALS team has recently started a new research group that will focus more on creating new algorithms. These algorithms will address the gaps that exist in effects of genetic variants. The use of the ExPecto framework on ALS patient- and control-data, will help to clarify what is needed in new algorithms.

One of the disadvantages of ExPecto, is that the framework is focused on a range of 40 kb around a TSS. It could be beneficial to look at even broader regions, since an enhancer (a regulatory part of the genome) is likely to be much further away from the gene. One tool that is capable of taking further regions of the genome into account, is Basenji. It uses a Dilated convolutional Neural Network (55).

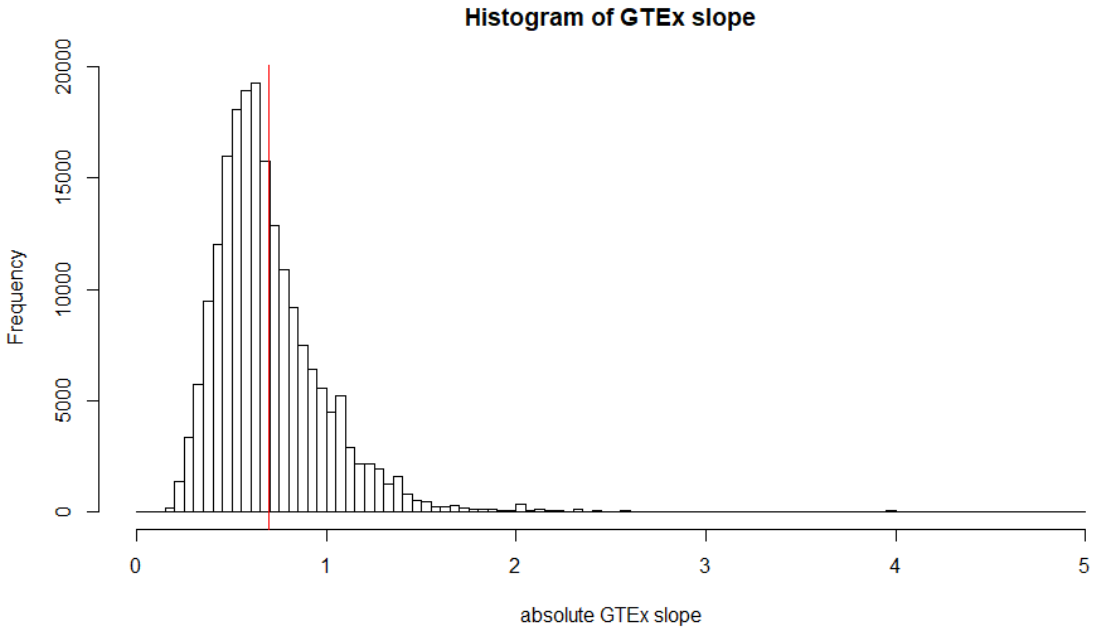
It is also interesting to compare new tools to the state of the art, like CADD, to see if they add any (new) value. CADD has been used in this thesis project, but there are other tools that address at least a part of the same problem as ExPecto and CADD. There are more machine learning options to find disruptive variants in regulatory regions, like the support vector machine gkm-SVM.

Appendix 1

Histograms of GTEx slope and MPRA mean. The red line is the mean of the dataset.

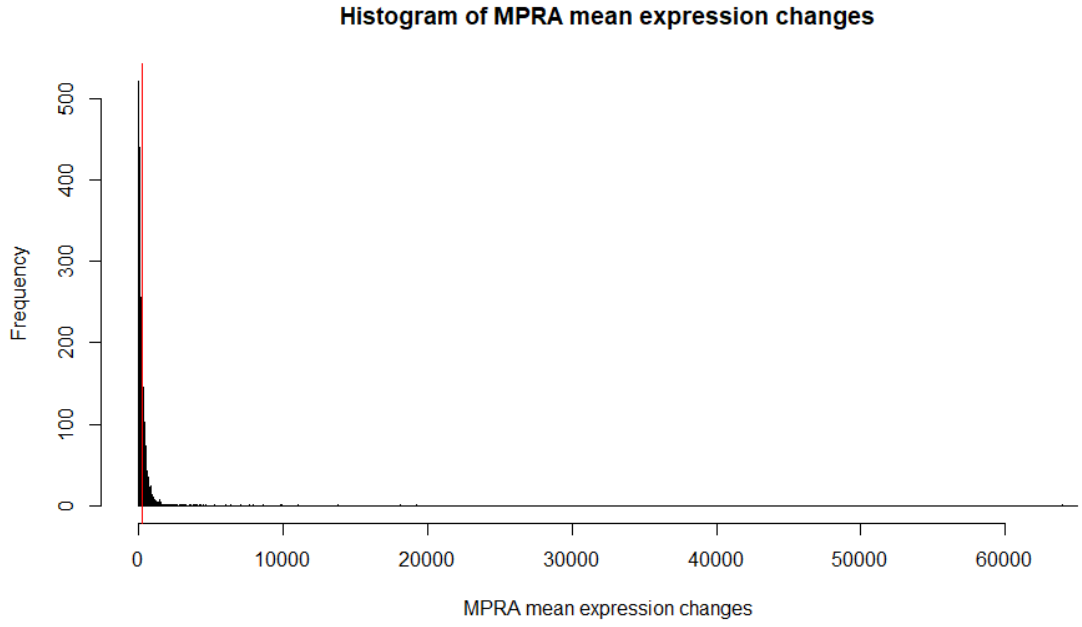
Histogram of GTEx slope

```
hist(abs(roc_gtex_expecto$slope), xlab = "absolute GTEx slope",
breaks = seq(0, 5, 0.05), main = "Histogram of GTEx slope")

abline(v=mean(abs(roc_gtex_expecto$slope)),col="red")
```



Histogram of MPRA mean expression changes

```
hist(abs(roc_mpra_cadd$mean), xlab = "MPRA mean expression changes",
breaks = seq(0, 65000, 10), main = "Histogram of MPRA mean
expression changes")

abline(v=mean(abs(roc_mpra_cadd$mean)),col="red")
```



Appendix 2

The code in this appendix is used for creating the ROC curves for the GTEx analysis and the MPRA analysis.

```
# ROC gtex expecto analysis
library(pROC)
roc_gtex_expecto <- x4[,c(6,16)]
hist(abs(roc_gtex_expecto$slope), xlab = "absolute GTEx slope",
breaks = seq(0, 5, 0.05), main = "Histogram of GTEx slope")
abline(v=mean(abs(roc_gtex_expecto$slope)),col="red")
roc_gtex_expecto$observation_class <- ifelse(test =
abs(roc_gtex_expecto$slope) < mean(abs(roc_gtex_expecto$slope)), yes
= 0, no= 1)
# remove ugly padding
par(pty = "s")
roc_curve_gtex_expecto <- roc(roc_gtex_expecto$observation_class,
abs(roc_gtex_expecto$Cells.EBV.Transformed.Lymphocytes), plot=TRUE,
print.auc=TRUE, main="ROC curve ExPecto GTEx")
coords(roc_curve_gtex_expecto, "best", ret = "threshold")
# best threshold of expecto scores is 5.960464e-08
# make class column based on threshold
roc_gtex_expecto$expecto_class <- ifelse(test =
abs(roc_gtex_expecto$Cells.EBV.Transformed.Lymphocytes) < 5.960464e-
08, yes = 0, no= 1)
library(caret)
confusionMatrix(table(roc_gtex_expecto$expecto_class,
roc_gtex_expecto$observation_class))

# to put graphs to normal size again.
par(pty = "m")
```

Appendix 3

```
# Frontal gdb with 1%
cut -f11 -d',' outputfrontal_combined.csv > scores.txt
awk -F "," '{ if(($11 > 0.05139208) || ($11 < -0.05139208)) { print
} }' outputfrontal_combined_filter2.csv >
outputfrontal_combined_filter3.csv

/hpc/hers_en/kkenna/lib/miniconda3/bin/Rscript
/hpc/hers_en/kkenna/lib/miniconda3/lib/R/library/rvat/exec/rvat.R --
importAnno gdb=/hpc/hers_en/mdegroot/chromosomes/genome.gdb --
name=frontal2 --
value=/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/outputfron
tal_combined_filter3.csv --sep=,

/hpc/hers_en/kkenna/lib/miniconda3/bin/Rscript
/hpc/hers_en/kkenna/lib/miniconda3/lib/R/library/rvat/exec/rvat.R --
genVarSet --gdb=/hpc/hers_en/mdegroot/chromosomes/genome.gdb --
unitTable=frontal2 --unitName=gene --intersection=var --
output=/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/frontalco
rtex2.varSet.txt.gz

# divide varset file in smaller bits.
mkdir -p /hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/split2

zcat
/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/frontalcortex2.v
arSet.txt.gz | awk 'BEGIN{FS="|";OFS="|"}{print $1,$2,"1"}' | awk
'NR%100==1{out="/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/
split2/frontalcortex2.v2."++i".varSet.txt"}{print > out}'

gzip
/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/split2/frontalco
rtex2.v2.*.varSet.txt

# do gene burden analysis on all files.
echo -e "/hpc/hers_en/kkenna/lib/miniconda3/bin/Rscript
/hpc/hers_en/kkenna/lib/miniconda3/lib/R/library/rvat/exec/rvat.R --
rvb --gdb=/hpc/hers_en/mdegroot/chromosomes/genome.gdb --
varSet=/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/split2/fr
ontalcortex2.v2.\${SGE_TASK_ID}.varSet.txt.gz --
varSetName=frontalcortex --cohort=df2v1 --pheno=pheno --
covar=pc1,pc2,pc3,pc4 --aggregationMethod=allelic --test=burden --
maxAF=0.001 --
output=/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/split2/fr
ontalcortex2.burden.\${SGE_TASK_ID}.txt.gz" | qsub -l
"h_rt=24:00:00" -l "h_vmem=8G" -N frontalcortex2_burden -o
/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/ -e
/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/ -t 1:174

zcat
/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/split2/frontalco
rtex2.burden.*.txt.gz | gzip -c >
/hpc/hers_en/mdegroot/chromosomes/gdb_frontalcortex/frontalcortex2.b
urden.txt.gz
```

Appendix 4

Code for Manhattan and QQ-plots

```
# R code for manhattan and qq-plot
library(rvat)
library(ggplot2)
library(dplyr)
library(sqldf)
assoc=read.table(gzfile("/hpc/hers_en/mdegroot/chromosomes/gdb_front
alcortex/frontalcortex2.burden.txt.gz"),h=F,as.is=T,sep="|")
names(assoc)=unlist(strsplit("varSetName|unit|pheno|covar|aggregatio
nMethod|test|case|ctrl|caseN|ctrlN|caseMean|ctrlMean|caseMeanGeno|ct
rlMeanGeno|caseSdGeno|ctrlSdGeno|OR|ORlower|ORupper|P",split="\\|"))
genes=read.table("/hpc/hers_en/mdegroot/ExPecto/resources/geneanno.p
c.sorted.nochr.bed",sep="\t",h=F,as.is=T)
names(genes)=c("CHROM","START","STOP","STRAND","unit")
genes$POS=round((genes$START + genes$STOP)/2)

assoc=sqldf("select * from assoc left join genes using (unit)")
assoc$P[assoc$P<(10^-16)]=10^-16
rvat::manhattan(assoc)
rvat::qqplot(assoc$P[(assoc$case +
assoc$ctrl)>3],case=max(assoc$caseN), control=max(assoc$ctrlN))
```

Bibliography

1. Clerget-Darpoux F, Elston RC. Will formal genetics become dispensable? Hum Hered. 2014;76(2):47–52.

2. Zitnik M, Nguyen F, Wang B, Leskovec J, Goldenberg A, Hoffman MM. Machine Learning for Integrating Data in Biology and Medicine: Principles, Practice, and Opportunities. Inf Fusion. 2018;50:71–91.

3. Maston GA, Evans SK, Green MR. Transcriptional Regulatory Elements in the Human Genome. Annu Rev Genomics Hum Genet. 2006;7(1):29–59.

4. Wieringa RJ. Design science methodology: For information systems and software engineering. Design Science Methodology: For Information Systems and Software Engineering. 2014. 1-332 p.

5. Rojano E, Seoane P, Ranea JAG, Perkins JR. Regulatory variants : from detection to predicting impact. 2018;(March):1–16.

6. Caballero-hernandez D, Toscano MG, Cejudo-guillen M, Garcia-martin ML, Lopez S, Franco JM, et al. The ' Omics ' of Amyotrophic Lateral Sclerosis. Trends Mol Med [Internet]. 2016;22(1):53–67. Available from: http://dx.doi.org/10.1016/j.molmed.2015.11.001

7. Hardiman O, Al-chalabi A, Chio A, Corr EM, Robberecht W, Shaw PJ, et al. Amyotrophic lateral sclerosis. Nat Rev. 2017;3.

8. Traynor BJ, Arthur KC, Calvo A, Price TR, Geiger JT, Chio A. from 2015 to 2040. 2016;1–6.

9. Stifani N. Motor neurons and the generation of spinal motor neuron diversity. 2014;8(October):1–22.

10. Bodmer W, Bonilla C. Europe PMC Funders Group Common and rare variants in multifactorial susceptibility to common diseases. 2008;40(6):695–701.

11. Brown RH, Al-Chalabi A. Amyotrophic lateral sclerosis. Prog Med Chem. 2017;58:63–117.

12. Smith BN, Ticozzi N, Fallini C, Gkazi AS, Topp S, Kenna KP, et al. Report Exome-wide Rare Variant Analysis Identifies TUBA4A Mutations Associated with Familial ALS. 2014;324–31.

13. Nicolas A, Kenna KP, Renton AE, Shaw CE, Traynor BJ, Landers JE, et al. Genome-wide Analyses Identify KIF5A as a Novel Article Genome-wide Analyses Identify KIF5A as a Novel ALS Gene. Neuron. 2018;97:1268–83.

14. DeJesus-Hernandez M, Mackenzie IR, Boeve BF, Boxer AL, Baker M, Rutherford NJ, et al. Expanded GGGGCC Hexanucleotide Repeat in Noncoding Region of C9ORF72 Causes Chromosome 9p-Linked FTD and ALS. Neuron [Internet]. 2011;72(2):245–56. Available from: http://dx.doi.org/10.1016/j.neuron.2011.09.011

15. Taylor JP, Brown Jr RH, Cleveland DW. Decoding ALS : from genes to mechanism. Nat Genet. 2016;539.

16. Germain DP, Jurca-Simina IE. Principles of Human Genetics and Mendelian Inheritance. In: Burlina AP, editor. Neurometabolic Hereditary Diseases of Adults [Internet]. Cham: Springer International Publishing; 2018. p. 1–28. Available from: https://doi.org/10.1007/978-3-319-76148-0_1

17. Consortium IHGS. Finishing the euchromatic sequence of the human genome. Nature [Internet]. 2004;431(7011):931–45. Available from: https://doi.org/10.1038/nature03001

18. Wheeler DA, Srinivasan M, Egholm M, Shen Y, Chen L, McGuire A, et al. The complete genome of an individual by massively parallel DNA sequencing. Nature. 2008;452(7189):872–6.

19. Pray LA. Discovery of DNA Structure and Function : Watson and Crick The First Piece of the Puzzle : Miescher Discovers DNA Laying the Groundwork : Levene Investigates the Structure of DNA. Nat Educ. 2008;1(2008):1–8.

20. NIGMS. The New Genetics [Internet]. National Institute of General Medical Sciences, editor. 2010. 4-31 p. Available from: https://www.nigms.nih.gov/education/Booklets/the-new-genetics/Pages/Home.aspx

21. Saitou N. Human Evolution and Human Genome at a Glance. In 2017. p. 3–17.

22. Carlberg C, Molnár F. Human Epigenomics. Kuopio: Springer; 2018. 5-8 p.

23. Mariño-Ramírez L, Kann MG, Shoemaker BA, Landsman D. Histone structure and nucleosome stability. Expert Rev Proteomics. 2005;2(5):719–29.

24. Nica AC, Dermitzakis ET. Expression quantitative trait loci: Present and future. Philos Trans R Soc B Biol Sci. 2013;368(1620).

25. Takahashi MU, Nakagawa S. Transcription Factor Genes. Evol Hum Genome I Genome Genes [Internet]. 2017;241–63. Available from: http://link.springer.com/10.1007/978-4-431-56603-8_12

26. Jimenez-Pacheco A, Franco JM, Lopez S, Gomez-Zumaquero JM, Magdalena Leal-Lasarte M, Caballero-Hernandez DE, et al. Epigenetic Mechanisms of Gene Regulation in Amyotrophic Lateral Sclerosis. In: Delgado-Morales R, editor. Neuroepigenomics in Aging and Disease [Internet]. Cham: Springer International Publishing; 2017. p. 255–75. Available from: https://doi.org/10.1007/978-3-319-53889-1_14

27. Gates LA, Foulds CE, O'Malley BW. Histone Marks in the 'Driver's Seat': Functional Roles in Steering the Transcription Cycle. Trends Biochem Sci. 2017;42(12):977–89.

28. Strahl BD, Allis CD. The language of covalent histone modifications. Nature. 2000;403(6765):41–5.

29. Marees AT, de Kluiver H, Stringer S, Vorspan F, Curis E, Marie-Claire C, et al. A tutorial on conducting genome-wide association studies: Quality control and statistical analysis. Int J Methods Psychiatr Res. 2018;27(2):1–10.

30. Korte A, Farlow A. The advantages and limitations of trait analysis with GWAS : a review Self-fertilisation makes Arabidopsis particularly well suited to GWAS. Plant Methods. 2013;9(1):29.

31. Li B, Liu DJ, Leal SM. Identifying rare variants associated with complex traits via sequencing. Curr Protoc Hum Genet. 2013 Jul;Chapter 1:Unit 1.26.

32. Smedley D, Schubach M, Jacobsen JOB, Ko S, Zemojtel T, Spielmann M, et al. ARTICLE A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease. 2016;595–606.

33. Todorovic V. Predicting the impact of genomic variation. Nat Publ Gr [Internet]. 2016;13(3):203. Available from: http://dx.doi.org/10.1038/nmeth.3793

34. He Z, Liu L, Wang K, Ionita-Laza I. A semi-supervised approach for predicting cell-type specific functional consequences of non-coding variation using MPRAs. Nat Commun [Internet]. 2018;9(1). Available from: http://dx.doi.org/10.1038/s41467-018-07349-w

35. Rentzsch P, Witten D, Cooper GM, Kircher M, Shendure J. CADD : predicting the

deleteriousness of variants throughout the human genome. 2018;1–9.

36. Kircher M, Witten DM, Jain P, O'roak BJ, Cooper GM, Shendure J. A general framework for estimating the relative pathogenicity of human genetic variants. Nat Genet [Internet]. 2014;46(3):310–5. Available from: http://dx.doi.org/10.1038/ng.2892

37. Zhou J, Theesfeld CL, Yao K, Chen KM, Wong AK, Troyanskaya OG. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. Nat Genet [Internet]. 2018; Available from: http://dx.doi.org/10.1038/s41588-018-0160-6

38. Ghandi M, Mohammad-Noori M, Ghareghani N, Lee D, Garraway L, Beer MA. GkmSVM: An R package for gapped-kmer SVM. Bioinformatics. 2016;32(14):2205–7.

39. LeCun Y, Bengio Y, Hinton G. Deep learning. Nat Methods. 2015;13(1):35.

40. Demir-Kavuk O, Kamada M, Akutsu T, Knapp EW. Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features. BMC Bioinformatics. 2011;12:1–10.

41. Quackenbush J. Microarray data normalization and transformation. Nat Genet [Internet]. 2002;32 Suppl(december):496–501. Available from: http://www.ncbi.nlm.nih.gov/pubmed/12454644

42. Dunham I, Kundaje A, Aldred SF, Collins PJ, Davis CA, Doyle F, et al. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012;489(7414):57–74.

43. Karolchik D, Kent WJ. USCS Genome Browser. 2010. 1-34 p.

44. Dreiseitl S, Ohno-Machado L. Logistic regression and artificial neural network classification models: a methodology review. J Biomed Inform [Internet]. 2002;35(5):352–9. Available from: http://www.sciencedirect.com/science/article/pii/S1532046403000340

45. Bennett SA, Tanaz R, Cobos SN, Torrente MP. Epigenetics in amyotrophic lateral sclerosis: a role for histone post-translational modifications in neurodegenerative disease. Transl Res [Internet]. 2019;204:19–30. Available from: https://doi.org/10.1016/j.trsl.2018.10.002

46. Fries GR, Colpo GD, Monroy-Jaramillo N, Zhao J, Zhao Z, Arnold JG, et al. Distinct lithium-induced gene expression effects in lymphoblastoid cell lines from patients with bipolar disorder. Eur Neuropsychopharmacol [Internet]. 2017;27(11):1110–9. Available from: http://dx.doi.org/10.1016/j.euroneuro.2017.09.003

47. Zheng HF, Forgetta V, Hsu YH, Estrada K, Rosello-Diez A, Leo PJ, et al. Whole-genome sequencing identifies EN1 as a determinant of bone density and fracture. Nature. 2015;526(7571):112–7.

48. Kelley DR, Snoek J, Rinn JL. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. 2016;26(7):990–9.

49. Agarwal V, Shendure J. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. bioRxiv [Internet]. 2018;416685. Available from: https://www.biorxiv.org/content/early/2018/09/13/416685

50. Rodriguez-Murillo L, Salem RM. Insertion/Deletion Polymorphism. In: Gellman MD, Turner JR, editors. Encyclopedia of Behavioral Medicine [Internet]. New York, NY: Springer New York; 2013. p. 1076. Available from: https://doi.org/10.1007/978-1-4419-1005-9_706

51. Gagliano SA, Sengupta S, Sidore C, Maschio A, Cucca F, Schlessinger D, et al. Relative impact of indels versus SNPs on complex disease. Genet Epidemiol. 2019;43(1):112–7.

52. Neph S, Kuehn MS, Reynolds AP, Haugen E, Thurman RE, Johnson AK, et al. BEDOPS: High-performance genomic feature operations. Bioinformatics. 2012;28(14):1919–20.

53. Tewhey R, Kotliar D, Park DS, Liu B, Winnicki S, Reilly SK, et al. Direct identification of hundreds of expression-modulating variants using a multiplexed reporter assay. Cell [Internet]. 2016;165(6):1519–29. Available from: https://doi.org/10.1016/j.cell.2016.04.027

54. Ehret GB. Genome-wide association studies: contribution of genomics to understanding blood pressure and essential hypertension. Curr Hypertens Rep. 2010 Feb;12(1):17–25.

55. Kelley DR, Reshef YA, Bileschi M, Belanger D, McLean CY, Snoek J. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. bioRxiv [Internet]. 2018; Available from: https://www.biorxiv.org/content/early/2018/03/22/161851