Predicting gene expression using artificial neural networks
(HS-IDA-MD-02-204)

Lisa Lindefelt (a98lisli@student.his.se)

Department of Computer Science,

University of Skövde, P.O.Box 408
SE-541 28 Skövde, SWEDEN

Masters Dissertation Spring 2002

Study Program in Bioinformatics

Supervisor: Björn Olsson University of Skövde

[Predicting gene expression using artificial neural networks]

Submitted by Lisa Lindefelt to Högskolan Skövde as a dissertation for the degree of M.Sc., in the Department of Computer Science.

[date]

I certify that all material in this dissertation which is not my own work has been identified and that no material is included for which a degree has previously been conferred on me.

Signed: _____

Predicting gene expression using artificial neural networks

Lisa Lindefelt (a98lisli@student. his.se)

Abstract

Today one of the greatest aims within the area of bioinformatics is to gain a complete understanding of the functionality of genes and the systems behind gene regulation. Regulatory relationships among genes seem to be of a complex nature since transcriptional control is the result of complex networks interpreting a variety of inputs. It is therefore essential to develop analytical tools detecting complex genetic relationships.

This project examines the possibility of the data mining technique artificial neural network (ANN) detecting regulatory relationships between genes. As an initial step for finding regulatory relationships with the help of ANN the goal of this project is to train an ANN to predict the expression of an individual gene. The genes predicted are the nuclear receptor PPAR-g and the insulin receptor. Predictions of the two target genes respectively were made using different datasets of gene expression data as input for the ANN. The results of the predictions of PPAR-g indicate that it is not possible to predict the expression of PPAR-g under the circumstances for this experiment. The results of the predictions of the insulin receptor indicate that it is not possible to discard using ANN for predicting the gene expression of an individual gene.

Keywords: Artificial neural networks, gene expression data, machine learning, diabetes

Table of contents

1 Introduction

In this project the possibility of the data mining technique artificial neural network detecting regulatory relationships between genes is examined. One of the reasons that this is important to explore is that the Human Genome Project will uncover the template behind all human biological functions, and more complex problems can be investigated (Kanehisa, 1996). One of the greatest aims within this area today is to gain a complete understanding of the functionality of genes and the systems behind gene regulation, that is how the genes interact (Tamayo et al., 1999). Reaching this aim is a big and demanding problem and no simple solutions exist. Advances in molecular biological and computational technologies are enabling us to investigate the processes underlying biological systems (D´haeseleer et al., 2000). Great progress has recently been made through gene expression analysis.

Almost every cell of an organism contains the entire genome of the organism (Campbell, 1999). However, in each cell only some of the genes of the genome are transcribed from DNA to RNA, and then translated into a protein. It is the proteins that are believed to decide the function of the cell, and in general an organism consists of many different types of cells where each cell has a certain biological function (Campbell, 1999). The different cells react differently to different environments. Cells in a muscle tissue, for example, have a completely different function from the cells of the brain. Not all the genes of a cell are expressed at the same time, the genes expressed can differ between different time points due to different circumstances, e.g. during the cell cycle (Campbell, 1999).

By performing a gene expression analysis, it can be decided which genes are expressed in a cell. It is through the development of the microarray technique in 1995 that gene expression analysis is possible. With help of this technique it is possible to observe the expression of thousands of genes simultaneously (Dopazo et al., 2001). Comparing expression data from different tissues can for example give clues about the function of important genes since it is likely that co-expressed genes are involved in the same regulatory process (D´haeseleer et al., 2000).

Various methods have been developed to analyse the huge amounts of data generated by the microarray technique. However today there are many data mining approaches that have not yet been investigated for this purpose. Data mining is a method commonly used when large sets of data need to be analysed. Persidis (2000) describes data mining as the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. Data mining is a huge industry in many areas and the process is becoming more important within biotechnology. This project focuses on the potential of using one possible data mining technique, artificial neural networks, to detect regulatory relationships between genes.

The goal of the project is to train an artificial neural network (ANN) to predict the expression pattern of one gene, using gene expression data as input. The genes, whose expression is predicted by an ANN, are the nuclear receptor peroxisome proliferator-activated receptor gamma (PPAR-g) and the insulin receptor (INSR). As the prediction of the target gene is relatively successful in one of the cases this opens up possibilities for detecting regulatory relationships. By interpreting the ANN it can be possible to understand which genes are influencing the expression of the target gene. Interpreting an ANN is not an easy task and will not be done in this project. In this

project only an initial step is taken towards detecting regulatory relationships with help of ANNs.

ANNs have proven to be effective in solving classification problems, such as pattern recognition, and modelling complex and highly non-linear problems in science and engineering (Patterson, 1996) and the potential of ANNs for gene prediction are considered, in this project, to be very good. The purpose is to discover biologically meaningful patterns in the data. As the results of predicting the expression of a gene by using an ANN is relatively successful in one of the cases, this shows that the information intrinsic in gene expression data can be enough for finding regulatory relationships between genes.

The nuclear receptor PPAR is involved in the activation and repression of the transcription of genes. The role of PPAR is to maintain an appropriate level of molecules that facilitate the state of normal insulin sensitivity (Olefsky and Saltiel, 2000). Another compound important for insulin sensitivity is the insulin receptor (INSR). The receptor enables a cell to extract glucose from the blood. Glucose is a source of energy for the body and the mechanism of uptake of glucose from the blood is essential. The change of insulin sensitivity is crucial and can lead to the development of diabetes. The data that used in the experiments during this project is collected from gene expression of human cells in different tissues and general information about the individual (see Chapter 2.7).

2 Background

In this chapter all concepts needed for this project are introduced. Chapter 2.1 gives an overview of data mining and Chapter 2.2 describes the data mining technique used in this project, namely artificial neural networks, and the fundamental concepts of artificial neural networks. Chapter 2.3 gives an overview of gene expression data, Chapter 2.4 gives an introduction to nuclear hormone receptors and PPAR-g, and in Chapter 2.5 a description of the insulin receptor is given. Chapter 2.6 describes the disease diabetes and Chapter 2.7 describes the gene expression data used for this project.

2.1 Data mining

Data mining is the process of finding trends and patterns in data. Persidis (2000) describes the objective of data mining as to extract previously unknown and potentially useful information from large datasets. One of the definitions of data mining is:

“Data mining is the efficient discovery of valuable, non-obvious information from a large collection of data.” (Bigus, 1996)

Some tasks well-suited for data mining are classification, estimation, prediction, affinity grouping, clustering, and description (Berry and Linoff, 1997). Data mining only makes sense when there are large volumes of data, and most data mining algorithms require large amounts of data in order to build and train the models that are used to perform data mining tasks. Most data mining methods learn from examples (Persidis 2000). For example the neural network or decision tree generator is fed with

thousands and thousands of training examples, and by doing so the data mining tool finds patterns and subtle relationships in data, and infers rules that allow the prediction of future results. After seeing enough of these training examples, the data mining tool comes up with a response model. The response model is in a form of computer program which allows the prediction of future results (Berry and Linoff, 1997).

Data mining is a huge industry in many areas with a lot of companies providing software products and services to clients that obtain, generate, and rely on large quantities of data (Persidis, 2000). Industries like manufacturing, database providers, government, the travel industry, banking and financial industry, telecommunications, and engineering are some examples where data mining is an important technique (Persidis, 2000). Data mining is increasingly used within the pharmaceutical industry. It is an approach to help deal with the enormous amounts of biological information that the industry collects (Persidis, 2000). The type of biological data needed to be interpreted today range from annotated databases of disease profiles and molecular pathways to sequences, structure-activity relationships, chemical structures of combinatorial libraries of compounds, and individual and population clinical trial results (Persidis, 2000). The aim of data mining is to help make sense of these complex data sets in an intuitive and efficient manner.

2.2 Artificial neural networks

In this chapter artificial neural network is described. Chapter 2.2.1 gives an overview of artificial neural networks and Chapter 2.2.2 introduces the tool used of creating an artificial neural network.

2.2.1 Overview of artificial neural networks

Among the techniques used for data mining are artificial neural networks (ANNs).

Much of the research on ANNs has been inspired by the knowledge of biological nervous systems. The nervous system of animals consists of a large number of interconnected neurons (nerve cells). A neuron is a small cell receiving electrochemical stimuli from multiple sources, and responds by generating electrical impulses transmitted to other neurons or effector cells (Patterson, 1996). A brain cell summarizes all incoming signals from surrounding brain cells. If the total sum of these signals is high enough the brain cell switches to “active”, that is, the neuron is responding (Patterson, 1996).

Artificial neural networks can be described as simplified models of the central nervous system (Patterson, 1996). ANNs are biologically inspired in that they perform in a manner similar to the basic functions of the biological neuron (although the ANNs are very simplified). The principle for constructing ANNs is based on how the neurons are inter-connected. ANNs are networks of highly interconnected neural computing elements. These elements, or nodes, have the ability to respond to input stimuli and to adapt to the environment (Patterson, 1996). The main advantage of using the concepts of ANNs in computational strategies is that they are able to modify their behaviour in response to their environment (input-output). The knowledge of the network is encoded in weights, where weights are numeric values associated with links connecting network nodes. By weight change it is possible for the network to learn, and thus respond to the environment (Diederich, 1990). ANNs have been shown to be effective as computational processors for various tasks and are applied to

prediction, classification and clustering (Berry and Linoff, 1997). A simple ANN is shown in Figure 1.

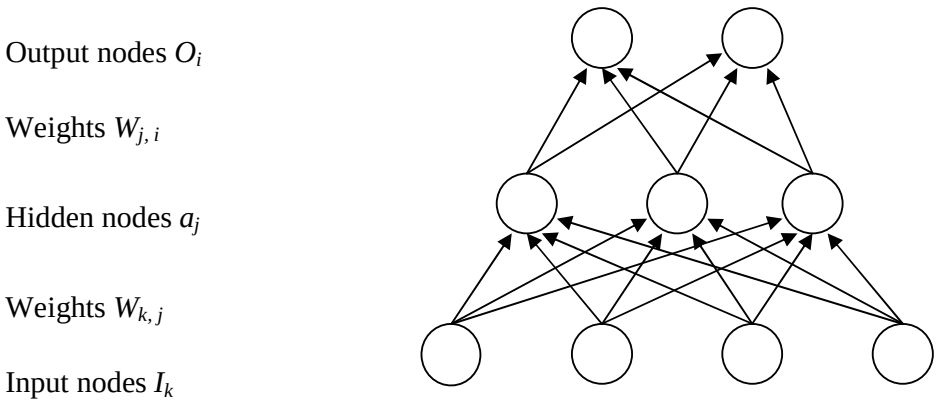


Figure 1. A two-layer feed forward neural network. The network consists of four input nodes, three hidden nodes, and two output nodes (circles in the figure). Each input node connects to all three hidden nodes and each hidden node connects to the two output nodes. Each connection is represented by a weight (arrows in the figure).

A neural network is a set of interconnected simple mathematical elements or units, called artificial neurons. As seen in Figure 1 each connection between nodes has a certain weight, the weights between the input nodes and the hidden nodes are denoted $W_{k,j}$ and the weights between the hidden nodes and the output nodes are denoted $W_{j,i}$. This weight influences the activation sent between the two nodes either by increasing or decreasing it. An artificial neuron summarizes all incoming signals from connected neurons. Then the summarized values from all incoming signals from connected neurons are used in an activation function to calculate the output of the neuron (Patterson, 1996). Depending on the activation function the output can vary quite a lot.

The training process for ANNs can be supervised or unsupervised (Bigus, 1996). Supervised learning is used when the answer is known, and the aim is to train ANNs

to map and generalise a certain function, output (Bigus, 1996). In these cases ANNs are trained on examples of inputs and corresponding outputs. For each example input the ANN generates in an output, and this output is compared with the correct output and an error rate can be calculated based on the difference between the generated output and correct output. Because the correct answer is known the weights can be adjusted so that the error rate of the output is reduced, and next time the prediction is closer to the correct answer. To reduce the error rate by changing the weights a learning algorithm is used (Bigus, 1996). Backpropagation is a common learning algorithm, and is used for this project.

Unsupervised learning is used in cases where the amount of data available is large and the answer is not known. Unsupervised learning is used when one wants to know how the data are related, what items are similar or different and in what way (Bigus, 1996).

2.2.2 Matlab

In this project Matlab is used for creating and training neural networks. According to Mathwork[1] the Matlab Neural network toolbox provides a complete set of functions and graphical user interface for the design, implementation, visualization, and simulation of neural networks. The Neural Network toolbox supports the most commonly used supervised and unsupervised network architectures and a set of training functions. The toolbox provides the user with elements necessary for creating a network. For this project the Matlab toolbox is used in a UNIX environment.

[1] The software Matlab toolbox is developed by Mathworks Inc and is freely available for 30 days at http//www.mathworks.com/products/neuraltnet/tryit.shtml

2.3 Gene expression data

This chapter introduces gene expression data and the microarray technique in Chapter 2.3.1. Chapter 2.3.2 describes the tool Genecluster used in this project for clustering.

2.3.1 Gene expression and the microarray technique

Because of the Human Genome Project the amount of DNA sequence data has been growing exponentially in recent years. One of the ultimate goals to accomplish is to understand the functions of the genes as well as the rules governing their interaction (Brazma and Vilo, 2000). By performing a gene expression analysis it is possible to decide which genes are expressed within a sample. The development of this DNA microarray technique in 1995 has provided scientists with a tool with the help of it is possible to simultaneously measure the expression of thousands of genes (Dopazo et al. 2001). By comparing expression data from different tissue samples it is, for example, possible to get clues about the function of important genes.

The microarray technique has been described by Dopazo et al, among others. One type of array is the complementary DNA-array (cDNA). This technique gives information about the expression level of thousands of genes in a single experiment. A cDNA array holds thousands of spots on a small glass plate or chip. Every spot on the array contains thousands of nucleotide sequences, which are complementary to a certain gene sequence. When the array is washed with fluorescent mRNA from a cell culture the mRNA binds to its complementary gene sequence on the array, and by

measuring the magnitude of fluorescence, the expression level for the gene can be decided, see Figure 2 (Dopazo *et al.*, 2001).

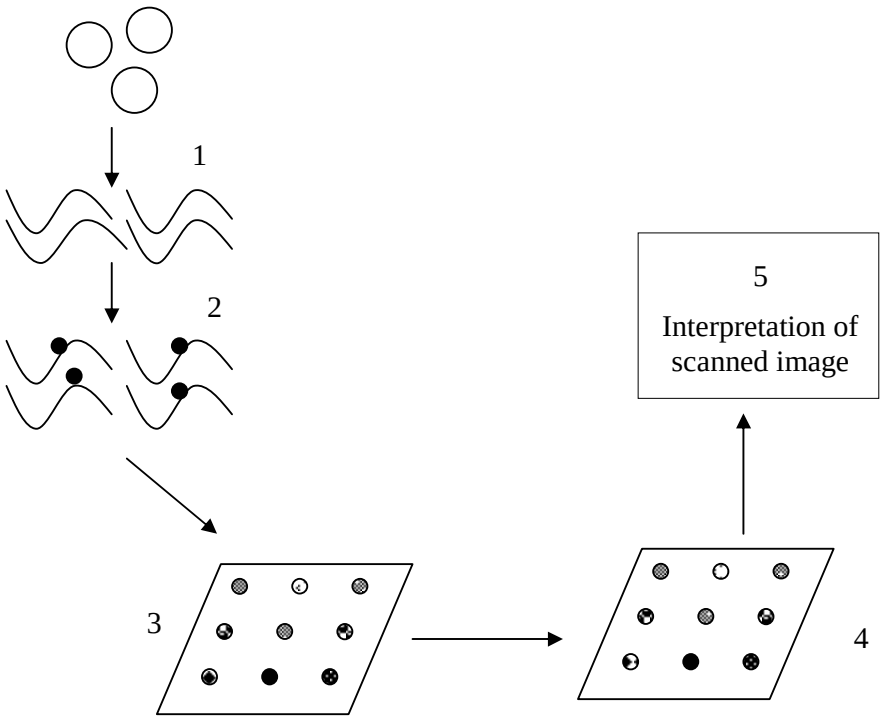


Figure 2: Gene expression analysis using a DNA microarray. (1) Extracting mRNA molecules from the cell cultures and reverse transcribing them to cDNAs. (2) Flourescent labeling of cDNAs. (3) Hybridization to a cDNA array. (4) Scanning the hybridized array. (5) Interpreting the scanned image.

This microarray technique generates enormous amounts of data and the challenge now is to interpret and analyse the resulting gene expression profiles. D´haeseleer et al. (2000) remarks that by classifying gene expression patterns it may be possible to investigate regulatory and functional relationships, and this is often done with help of clustering. Further D´haeseleer et al. (2000) believe that genes with similar expression pattern are likely to be involved in the same regulatory pathways. However, the

expression pattern of a gene may be due to a combination of different regulatory elements conferring different effects at different times or in different tissues (Birnbaum et al., 2001).

An area, in which the microarray technique is very useful, is in the study of differential gene expression in disease (Debouck and Goodfellow, 1999). Differential gene expression patterns in diseases are possible to explore by comparing the expression of thousands of genes between diseased and normal tissues and cells (Debouck and Goodfellow, 1999).

2.3.2 Genecluster

Tamayo et al. (1999) has developed an implementation of Self-organizing maps, SOMs, called Genecluster. Self-organizing maps is a type of neural network that learns to classify input vectors according to how they are grouped in the input space, and Genecluster is a product used to reveal important patterns for a set of gene expression data. Genecluster is used in this project to cluster the genes in order to get an average profile for the set of genes in each cluster, see Chapter 5.2.6. When deriving a SOM-map it is possible to save the centroid of each cluster. The centorid of a cluster is the average profile for that cluster.

2.4 Nuclear hormone receptors

Nuclear receptors form a super family of evolutionary related proteins involved in many important physiological processes and diseases (Aranda and Pascual, 2001).

Nuclear hormone receptor proteins are located in the cell nucleus and the receptors act there as transcription factors, regulating gene expression of hormonally regulated target genes (Tenbaum and Baniahmad, 1997). They function as ligand-activated transcription factors as they bind small lipophilic hormones produced by the organism's endocrine system and regulate gene expression by interacting with specific DNA sequences, known as hormone response elements, upstream of their target genes (Tenbaum and Baniahmad, 1997).

When bound to specific sequences of DNA, nuclear hormone receptor proteins serve as on-off switches for transcription within the cell nucleus (Parker, 1991). These switches control the development and differentiation of for example skin, bone, and behavioural centres in the brain, as well as the continual regulation of reproductive tissues (Parker, 1991). Based upon the observation of an inactive and an active state of the receptor a two-step mechanism of action has been proposed for these receptors. The first step involves activation through binding of the lipophilic hormone, a ligand, and the second step consists of receptor binding to DNA and regulation of transcription (Parker, 1991). Some nuclear receptors, however, have no known ligand and are referred to as (nuclear) orphan receptors (Robinson-Rechavi et al., 2001). Progress has been made over the last years to elucidate the role of these orphan receptors in animal biology (Chawla et al., 2001).

The superfamily of nuclear hormone receptors includes the classic steroid receptors (androgen, estrogen, glucocorticoid, mineralocorticoid, and progesterone receptors), the thyroid, vitamin D, and retinoid receptors, as well as many others more recently characterized (Robyr et al., 2000). One of the members of the group of nuclear

hormone receptor proteins is peroxisome proliferator-activated receptor (PPAR) for fatty acids (Chawla et al., 2001). There are three known PPAR sub types, two of which - PPAR-g and PPAR-α - have well described physiological importance as key modulators of lipid metabolism, while the third PPAR-δ is less observed (Jones 2001). This project focuses on PPAR-g. PPAR-g exists as a heterodimer with the nuclear receptor retinoid X (RXR) (Olefsky and Saltiel, 2000). The heterodimer binds to PPAR response elements within the promoter regions of target genes. In the unliganded state the heterodimer is associated with a multiprotein co-repressor complex. This co-repressor complex has histone deacetylase activity, which means that the transcription is inhibited. When the ligand binds the receptor the co-repressor complex dissociates (Olefsky and Saltiel, 2000).

The role of PPAR is to maintain an appropriate level of molecules that facilitate the state of normal insulin sensitivity (Olefsky and Saltiel, 2000). The change of insulin sensitivity is crucial and can lead to the development of diabetes.

2.5 Insulin receptor

The body is constantly using energy to drive the vital processes keeping us alive. However the energy intake is not constant and therefore it is necessary for the body to be able to store energy for subsequent use between meals. Insulin is the hormone responsible for regulating the sugar levels in the blood, and thereby insulin is coordinating and regulating the storage of the body energy, glucose (Campbell, 1999). Insulin is a hormone secreted into the blood when the level of glucose rises above a threshold. When secreted into the blood insulin gives signals to muscle liver and fat

cells that glucose is available for extraction of storage (Campbell, 1999). For the cells to be able to extract glucose from the blood insulin receptors are needed. An insulin receptor is a trans-membrane receptor protein and is able to respond to the signals given by insulin (Chen et. al., 1997). Insulin can bind to the insulin receptor. As insulin binds to the insulin receptor the shape of the receptor changes and glucose can enter the cell. A cell can increase or decrease the intake of glucose by regulating the number of receptors. If the level of glucose in the blood is constantly high, the insulin level remains high, and eventually all insulin receptors are removed from the surface of the cell. An inactive insulin receptor is a possible cause for diabetes (Chen et. al., 1997).

2.6 Diabetes

Diabetes is a disease where the sugar level of the blood is above normal (Harris, 1985). There are two types of diabetes, type 1 diabetes mellitus also known as insulin dependent diabetes mellitus (IDDM) and type 2 diabetes mellitus also known as non insulin dependent diabetes mellitus (NIDDM) (Campbell, 1999).

Pancreas is a gland excreting two different hormones, insulin and glucagon, directly into the blood (Campbell, 1999). Both insulin and glucagon are hormones regulated by the concentration of glucose in the blood. Glucose is a major fuel for cells and the metabolic balance is dependent on keeping the glucose concentration in the blood around a certain level. In humans this level is around 90mg per 100ml of blood (Campbell, 1999). When the concentration of glucose exceeds this level, insulin is secreted from the pancreas. Insulin gives signals to muscle, liver, and fat cells that

glucose is available for energy extraction or storage (Campbell, 1999). With help from insulin these cells can absorb glucose, and by doing so the insulin decreases the concentration of glucose in the blood. When the concentration of glucose falls below 90mg per 100ml blood, glucagon is released and the concentration of glucose in the blood increases (Campbell, 1999).

The antagonistic effects of glucagon and insulin are very important as forming the mechanism regulating the balance of glucose. When this mechanism does not work as it is supposed to, the consequences can be very serious. Diabetes mellitus is a disease caused by the lack of insulin (IDDM) or that the cells no longer respond to insulin (NIDDM). This results in very high concentrations of glucose in the blood. Because insulin is no longer available for the target cells, glucose is no longer an available source of energy for the cells of the body and the energy has to be taken from fat instead. In severe cases of diabetes, acids are produced and accumulate in the blood when fat is broken down. The consequence is a decrease of the pH-level in the blood, which is fatal (Campbell, 1999).

2.7 Data

The gene expression data used for this project is from a database that AstraZeneca has leased from the company Gene Logic Inc. The database consists of the three sub-databases BioExpress, ToxExpress and PharmExpress. BioExpress consists of gene expression data from normal and diseased tissues, and cell lines from humans and animals. The content of ToxExpress is effects of toxic compounds on rat tissues and rat and human primary cells. The third sub-database, PharmExpress, is under

development and in the future, this part is supposed to give information about the effects of therapeutic compounds on human and animal tissues and cell lines.

For this project the gene expression data from humans in BioExpress is used. All tests and analysis are done only on data from humans. Today there is microarray data for 65,000 human transcripts in the database from around 6,800 different samples. The technique used to generate the gene expression data is U95 microarray chip. To cover the entire genome, five arrays are needed. U95 microarray chip gives one quantitative and one qualitative measurement per spot. The quantitative measurement can be any real value. The qualitative measurement is divided into three different categories. The qualitative measurement for the mRNA level for a transcript can be “absent”, A, which means it has not been possible to detect any mRNA for that transcript, “present”, P, which means it is possible to find mRNA for the transcript, or “marginal”, M, which means the measurement of the mRNA for the transcript is in a “grey-zone” between the two former categories. Although marginal is interpreted as closer to absent than present.

It is possible to get information about the donor and chemical factors like, for example, the glucose level in the blood. The data about the donors stored in the database are shown in Table 1.

Basic donor information	**Social history**
- Data of birth	- Diet information
- Age at excision	- Smoking history
- Gender	- Alcohol consumption history
- Race	- Recreational drug use history
- Height	**Medical information**
- Weight	- Medical history
Obstetric information	- Surgical history
- Menstrual history	- Medications
- Last menstrual period	**Laboratory values**
- Pregnancy information	- Lab testes taken on day of surgery, lab values taken while in hospital
Family history	
- Family members with significant medical conditions	

Table 1. Donor information stored in the database.

Different donor data can be advantageous to have when the results from gene expression analysis are interpreted. But it is important to remember that much of the information that exists about a donor is given by the donor him-/herself. This may result it insufficient information, where the donor for example does not give information about the entire family history.

Other types of data that are stored in the database are different chemical measurements, like for example the glucose level of the blood, the cholesterol lever and the level of triglycerides.

3 Related works

In this project the aim is to train an ANN to predict the expression pattern of one gene. There is no former work performed concerning exactly the described aim. The article by Khan et al., (2001), described in Chapter 3.1 is based on a project where the aim is to train an ANN to be able to distinguish between four different kinds of cancer types using gene expression data as input. The article by Gruvberger et al., (2001) described in Chapter 3.2 is based on a project where the tumors are classified according to ER status by using ANNs and hierarchical clustering techniques.

In this project an ANN is trained to predict the expression of an individual gene, which is a different task from classifying cancer types. The reason for describing these releated works is to illustrate that attempts have been made for combining gene expression data and ANNs.

3.1 Classification and diagnostic prediction using ANN

Khan et al., (2001), performed classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. They trained the ANNs using the small, round blue-cell tumours (SRBCTs) as a model. These cancers belong to four distinct diagnostic categories and it is often hard to distinguish them by common clinical methods. To calibrate the ANN models to recognize cancers in each of the four SRBCT categories gene expression data from cDNA microarrays containing 6567 genes were used. From the entire data set of in total 88 samples there were 63 training sets and 25 test sets. The data set of 88 samples was quality filtered

and the number of genes was reduced to 2308. By principal component analysis (PCA) the dimensionality was further reduced into 10 PCA components. By performing a three-fold cross-validation procedure 3750 ANN models were produced, and with these models all 63 training samples were correctly classified to their respective categories. The 25 test experiments were subsequently classified using all the calibrated models.

After successful classification the next step was to determine the contribution of each gene to the classification by the ANN models. By measuring the sensitivity of the classification to a change in the expression level of each gene it was possible to rank the genes according to their significance for the classification. The classification error rate was determined by using increasing numbers of these ranked genes. The classification error rate minimised to 0% at 96 genes. The 10 dominant PCA components for these 96 genes contained 79% of the variance in the data matrix. By using only these 96 genes the ANN models were recalibrated and again correctly classified all 63 samples.

Khan et al., (2001) developed a method of diagnostic classification of cancers using gene expression profiles as input for an ANN. They identified the genes contributing to the classification, and they were able to define a minimal set that correctly classified their samples into the four diagnostic categories. The article by Khan et al. shows that using gene expression data for classification and pattern recognition with ANNs is promising. However in this project ANNs are used for investigating the possibility of finding regulatory relationships between genes by training an ANN to predict the expression of an individual gene. Training an ANN to predict the expression of an individual gene is a more complex task than classifying four types of

cancers. It is more complex compared to classifying four types of cancers because it requires that the ANN finds connections between several genes to predict the expression of a single gene. It is also interesting to investigate if it is possible to use ANNs for classification and pattern recognition and on other data sets. The gene expression data used by Khan et al., (2001) is generated from cancer cells only. In this project the gene expression of one gene is predicted using gene expression data associated with diabetes and gene expression data not associated with diabetes.

3.2 Classifying estrogen receptor status using ANN

Estrogens regulate gene expression via the estrogen receptorER, but the signalling pathways are yet not fully understood. In the article by Gruvberger et al., (2001) artificial neural networks and standard hierarchical clustering techniques were used to classify tumors according to ER status and to generate a list of genes which discriminate tumors according ER status, ER+ or ER-. ANNs and conventional methods were applied to analyse cDNA microarray data from a selected group of node-negative breast cancers that differ with respect to their ER status. The authors show that ER+ and ER- tumors display different phenotypes and this is thought to be due to their evolution from distinct cell lineages. In the experiments gene expression data from 3,389 genes were used. The dimensionality of these 3,389 genes was reduced by PCA to 10 components used as input for the ANN. The samples used for training and testing the ANN were classified into two categories using a three-fold cross-validation procedure. The sensitivity of an individual gene for classification was calculated. The sensitivity being large for a gene was considered to imply that changing the expression of the gene influences the output significantly, and thereby the genes could be ranked. For comparison with the ANN method the data was

analysed and the differences between tumors based on ER status was visualised by using two clustering techniques.

The ANN was able to classify all the 47 training samples and the 11 blinded test samples using only 100 of the genes most important for the classification. Conclusions were drawn that ER+ and ER- tumors exhibit distinct patterns of gene expression. The standard clustering algorithms support the conclusions based on the ANN models.

The differences between this project and the project by Gruvberger et al., (2001) are the same as the differences between this project and the project by Khan et al., (2001) described in Chapter 3.1. The big difference is that in Gruvberger et al., (2001) the aim of the training of the ANN was to investigate if an ANN was be able to classify ER status, whereas in this project ANNs are used for investigating the possibility of finding regulatory relationships between genes by training an ANN to predict the expression of an individual gene.

4 Problem definition

This project is examining the possibility of the data mining technique artificial neural network for detecting regulatory relationships between genes. The focus is on training an ANN to predict the expression of an individual gene using gene expression data as input for the ANN. This project illustrates problems with analysing large sets of gene expression data in order to find biologically interesting data.

Data mining is increasing within the pharmaceutical industry, and is a tool to help deal with the enormous amounts of biological information of various forms that the industry collects (Persidies, 2000). To be able to interpret the enormous amount of data generated from gene expression microarrays, methods for analysis have been developed. To analyse gene expression data many statistical methods have been used for clustering similar gene expression profiles e.g. (Tamayo et al., 1999, Eisen et al., 1998). Such techniques can group together co-expressed genes and genes thought to share similar function. However there are relationships among genes that can not be statistically expressed, for example regulatory relationships (Ando et al., 2001). Since transcriptional control is the result of complex networks interpreting a variety of inputs, the development of analytical tools detecting the multivariate nature of complex genetic relationships is essential (Bicciato et al., 2001). This project focuses on training an ANN to predict the expression pattern of an individual gene using gene expression data as input for the ANN. ANNs are used because the technique is known to be effective for, for example, pattern recognition and finding non-linear relationships (Patterson, 1996). Pattern recognition is achieved by adjusting parameters of the ANN by a process of error minimization through learning from

experience. ANNs can be calibrated using any type of input data, such as gene expression levels generated by cDNA microarrays (Khan et al., 2001). The output can be grouped into any given number of categories (Khan et al., 2001).

4.1 Hypothesis

The hypothesis is that an ANN using gene expression data as input predicts the approximate expression of an individual gene.

As example target genes the nuclear receptor PPAR-g and INSR are used. If the prediction made by the ANN is clearly better than a prediction made by random guessing, then it is not possible to falsify the hypothesis and thus the hypothesis is considered true.

The aim is to teach an ANN to predict the expression pattern of an individual gene from gene expression data. As input to the ANN the gene expression value from different genes for a sample is used and as output the gene expression value of the target gene for that sample is used. The expression of a gene for a sample can be either present (P), absent (A) or marginal (M) (see Chapter 2.7). In this project the nuclear hormone receptor PPAR-g and INSR are the two different target output genes, and thereby ANNs are trained to predict the expression of the receptor PPAR-g and the INSR respectively.

The ANN is fed with gene expression data from several samples for training. The more samples the ANN has for training the greater is probably the chance that the ANN is trained to recognise patterns in the gene expression data that makes it possible to predict the expression of the target gene.

The ANN is then tested on test samples, where the test samples have not been shown to the network earlier. By testing the ANN with test samples it is possible to determine how successful the training of the network is. If the network is able to predict the correct value of the gene expression of the target gene for all the test samples then the network training has been very successful. For further details about how the network is evaluated in this project (see Chapter 5.1.4).

Because of the great amount of gene expression data stored in the database BioExpress (for further information about BioExpress see Chapter 2.7) it is necessary to reduce the amount of input data for the ANN. The reduction will be done by selecting data (see Chapter 5.2.1). Because of the reduction of the amount of data and because the output of the ANN is known it can be discussed whether the approach in this project is data mining. However, as an ANN is used for finding unknown patterns in the selected data this can be considered to be a data mining approach.

4.2 Motivation

The development of analytical tools detecting the multivariate nature of complex genetic relationships is essential (Bicciato et al., 2001). One type of important information underlying the expression profile data is the 'genetic network', that is, the

regulatory network among genes (Toh and Horimoto, 2002). Different methods, like Boolean networks, continuous linear, and non-linear models (D´haeseleer 2000), have been used trying to create genetic networks. None of the methods used so far for deriving genetic networks have produced really reliable results. The aim of this project is to see if an ANN can find patterns in gene expression data for predicting the gene expression of one gene. Predicting the expression of one gene ought to be an easier task than deriving a genetic network. If it turns out to be possible to get reliable results when predicting the expression of one gene, it should also be possible to apply the method to one gene at the time, and thereby derive a genetic network. In this project, however, only the possibility of predicting the expression of a single gene is tested. ANNs are useful for finding relationships with high accuracy (Ando et al., 2001). Therefore using ANNs for pattern recognition in gene expression data can give indications whether the information intrinsic in gene expression data would be enough for finding regulatory relationship.

PPAR-g is chosen because it is a gene thought to be involved with diabetes, whereas INSR is a gene known to be involved in diabetes. Much research has been done to try to understand the mechanisms behind diabetes. Diabetes mellitus is a common disease affecting approximately 5 % of the population (Harris, 1985). Because diabetes mellitus is a common disease PPAR-g and INSR are very interesting genes for the pharmaceutical industry.

4.3 Aims and objectives

In this chapter the aims and objectives of the project are described. The aim is to train the ANN to predict the gene expression of PPAR-g and INSR respectively. In order to achieve this aim, the objectives described in the Chapters 4.3.1 to 4.3.6 need to be attained.

4.3.1 Reducing the amount of input data by selecting data

The amount of data stored in the database used for this project is very large, see chapter 2.7, and the ANN will become very large if all the genes stored in the database are used. Reducing the amount of input data prevents the ANN from growing to a size where it takes a lot of computer power to train the ANN. There is also a risk that the more input nodes the more examples are needed for the training of the ANN to be successful. Therefore it is necessary to reduce the amount of data that is used as input for the ANN. The first stage of reduction is excluding some of all the measurements stored in the database. It is for example be interesting to exclude the measurements for a sample where the body mass index (BMI) of the donor is not known. A high BMI (>25) is known to be associated with diabetes (Müller-Wieland, 2001) and when analysing the results of the project it can be interesting to have information about the BMI of the donors.

4.3.2 Deriving data for training and test the artificial neural network.

This is done by cross validation, which is a technique commonly used. Cross validation involves that the data set is divided into a number of equally large sub data sets. The ANN is trained with all of the sub data sets except for one, used as the test data for validation. Then the process is repeated using all of the sub data sets for validation.

4.3.3 Training the ANN for predicting the expression of the target gene.

The gene expression value for each gene for a sample is used as input for the ANN. If the expression of a gene does not show any variation over the different samples the gene is excluded from the data set. Then the remaining data is used as input for the ANN, and the network is trained to predict the expression profile of the nuclear receptor PPAR-g and INSR respectively.

4.3.4 Testing different network architectures and training algorithms

When training the network different architectures are used in order to find out which architecture suits this problem the best. There is however no attempt to do this exhaustively or systematically.

In Matlab it is possible to choose between different training algorithms for backpropagation. For this project different training algorithms are tested to investigate if the different training algorithms generate different results.

4.3.5 Validating the result for the ANN by comparing with random guessing

To validate the results from the prediction of the target gene by ANN random guessing is used.

If the result from the prediction of the expression of the receptor by the ANN is better than a prediction of the expression of the receptor by chance then a next step can be to interpret the weights in the ANN to understand which cluster has the greatest impact on the receptor. This is however not done in this project.

4.3.6 The different experiments

The predictions of PPAR-g and INSR are made using different datasets as input for the ANN. The following experiments are conducted:

- One prediction of the two target genes respectively is made with the transcripts of the genes shown in Appendix I associated with diabetes and PPAR-g.
- Another prediction of each of the target genes is made with a small set of arbitrarily chosen genes.
- One prediction of INSR is made with a larger set of arbitrarily chosen genes. The dimensionality of the larger dataset is reduced by clustering the genes with SOM (see Chapter 5.2.6).

5 Method

In this chapter the method is described. In Chapter 5.1 the experimental design for this project is described. In Chapter 5.2 the reduction of data and the different experiments is described.

5.1 Experimental design

This chapter describes the experimental design used in this project. Chapter 5.1.1 describes how a neural network is designed in Matlab. Chapter 5.1.2 describes the transfer functions used in the experiments, and Chapter 5.1.3 describes the training of the network. Chapter 5.1.4 describes how to evaluate the networks and Chapter 5.1.5 describes network generalization.

5.1.1 Neural network design

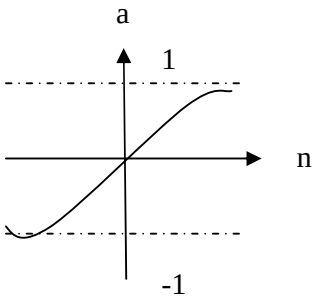
The architecture of a network is the network configuration of nodes and connections between the nodes. It is a description of how many layers a network has, the number of neurons in each layer, the transfer function for each layer, and how the layers connect to each other. The best architecture to use depends on the type of problem to be represented by the network. A single layer of neurons can represent simple problems. This type of network is widely used for linear separable problems, but it is not capable of solving non-linear problems (Rumelhart et al., 1986). A network with multiple feed-forward layers, however, a network can solve more complex problems. A multilayer feed-forward network can solve a non-linear problem by employing hidden layers and a non-linear transfer function. A feed-forward network is a network

that has one or more inputs propagated through a variable number of hidden layers (where each layer has a variable number of neurons) and then reaches the output layer. The values are fed forward through each layer, where the output from every node for one layer becomes the input for the next layer.

It is difficult to know the best architecture for a problem beforehand, therefore in this project a number of different architectures is used in order to find out which architecture suites this problem best.

5.1.2 The transfer function

The transfer function, also called the activation function, for a given neuron provides the means by which the inputs of that neuron are converted to outputs with desired characteristics. There are many transfer functions included in the Matlab toolbox. In this project the two transfer functions called tansig and logsig, in Matlab, is used. Tansig is a function returning elements between -1 and 1 see Figure 3.



a = tansig (n)

Figure 3. The tansig function in Matlab, ranging from -1 to 1.

According to the Matlab toolbox the transfer function tansig is commonly used between the input nodes and the layer of hidden nodes, and is used for that purposehere.

The target output for the experiments in this project (the experiments are described in Chapter 5.2) is 0 for absent and 1 for present (see Chapter 5.2.2). The target output should agree with the transfer function. Because the target is either 0 or 1 the log-sigmoid transfer function is used between the nodes in the hidden layer and the output nodes. The log-sigmoid function is used to scale the input of a neuron from the range of plus or minus infinity to the range of zero to one, see Figure 4.

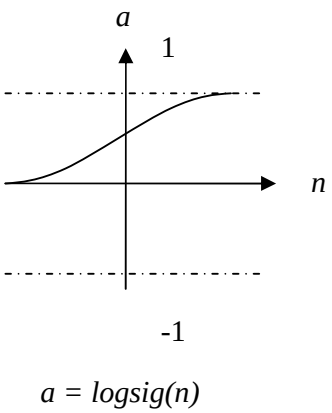


Figure 4. The logsig function in Matlab, ranging from 0 to 1.

When the desired output is 0 or 1 and the log-sigmoid function is used as the activation function for the output there is one important issue to consider (Mehrotra et al., 1997). The log-sigmoid transfer function returns an output value of 0 only when the net input is minus infinity, and the output value is 1 only when the net input is

infinity. According to Mehrotra et al., (1997) it is preferable to use a smaller value (1-t) instead of 1 and a larger value t instead of 0 for the desired output. Typically, $0.01 < t < 0.1$. Therefore in this project the target, the desired output is set to range between 0.05 and 0.95.

5.1.3 The learning rules

The learning rules provided in the neural network toolbox are defined as a procedure for modifying the weights of a network. This procedure may also be referred to as a training algorithm. The learning rules used here is backpropagation. Backpropagation is when input vectors and the corresponding target vectors are used to train a network until a goal is reached, that is the training algorithm is used to adjust the weights of the network in order to move the network outputs closer to the targets. Networks properly trained by backpropagation tend to give reasonable answers when presented with inputs that they have never seen, the network has generalised (see Chapter 5.1.5).

There are several different backpropagation training algorithms to choose between in the Matlab toolbox. Here two different training algorithms for backpropagation are tested to investigate if the different algorithms generate different results. One of the algorithms is according to the Matlab toolbox, a very good general purpose training algorithm performing well on pattern recognition problems. The other training algorithm is the fastest training algorithm for networks of moderate size found in Matlab[2] and is working well for networks containing up to a few hundred weights. In

[2] For further information see the neural network toolbox manual for Matlab version 6.1.6.450 release 12.1.

Matlab the former training algorithm is called trainscg and the latter is called trainlm. The Matlab toolbox uses these two training algorithms in example experiments similar to the experiments in this project.

As soon as the network weights have been initialized, the network is ready for training. The weights are initially set at random. The weights of the network are iteratively adjusted during training in order to minimize the network performance function. The default performance function in Matlab is Mean Square Error, MSE, which is the average squared error between the network outputs and the target outputs.

5.1.4 Evaluating the network

After the network has been trained and tested the performance function MSE shows the mean squared error between the network outputs and the target outputs. The MSE does not say anything about how accurately the network classifies the samples. Therefore, having MSE as the performance function makes it hard to interpret the network performance. Preferable instead is to calculate the accuracy of the network. The accuracy is how many percent of the input samples that are correctly classified as having absent or present expression of the target gene. To be able to calculate the accuracy it is necessary to change the performance function. During the experiments the target output of the ANN can be set to 0 and 1 (see Chapter 5.2). The new performance function used is then set to classify a received output with a value between 0.5 and 1 as 0.95, i.e. present, and a received output with a value between 0 and 0.5 as 0.05, i.e. absent. With this performance function it is possible to calculate

the accuracy, where the accuracy is the number of correctly classified samples divided by the total number of samples.

5.1.5 Generalisation of the network

By a successful training experiment the network is generally allowed to capture the essential relationships between inputs and outputs. In such cases a network has the capability of generalising, meaning that the network is able to perform well on examples not included in the training set. However, it is well known that excessive training on the training set sometimes decreases the performance on the test set. The network architecture is crucial for successful training (Mehrotra et al., 1997). A network with a larger number of nodes than required overtraining usually occurs. Overtraining is when the network is capable of memorizing the training set, and may not generalize well to test data. In these cases the network may learn undesirable features and therefore perform poorly on test data (Mehrotra et al., 1997). For this reason networks of smaller sizes are preferred over larger networks, but if a network is too small it does not learn the data. As discussed in Chapter 5.1.1 it is difficult to know which network architecture is the best for a certain problem, therefore different architectures is tried during the performances of the experiments.

According to Mehrotra et al., (1997) overtraining can be avoided by using networks with a small number of hidden nodes and weights. The number of parameters should be small compared to the number of samples the network is trained with. Therefore, in this project different experiments are carried out on different training sets where the number of genes, that is inputs for the ANN, differs.

5.2 Experiments

This chapter describes the performance of the different objectives stated in Chapter 4.3. Chapter 5.2.1 describes how the data used for this project was selected. The remaining part, that is Chapter 5.2.2 to Chapter 5.2.6, describe the different experiments performed. Each experiment describes a prediction of some kind, made by an ANN. For each experiment the training sets and test sets for the ANN were derived by cross-validation. Different network architectures have been used for the ANN in order to find out which architecture suites the particular experiment the best. For training the network the backpropagation function trainlm respectively trainscg were used.

5.2.1 Reducing the amount of input data by selecting data

To be able to handle the large amount of values of different kinds stored in the database the first aim was to come up with a way to reduce the amount of data. The first thing done was to choose samples where the biopsy was made on adipose, liver and muscle tissue. Through literature study it is shown that it has been proved that PPAR-g is expressed mostly in adipose tissue but also in liver and muscle tissue (Aranda and Pascual, 2001). This knowledge is the reason why biopsies from these tissues are chosen. To accomplish further reduction a variation filter was used. This variation filter excludes all genes not showing at least a 5% variation over the different samples for the chosen tissues. There is a risk taken when a variation filter like this is used. It is possible to exclude some genes that could be contributing with valuable information for the results if they were used. To lower the risk of excluding

genes that could contribute with valuable information for the results, genes that are, found in related literature, proved to be involved in diabetes and with PPAR-g are not excluded from the data set even though they do not show a 5% variation over all samples regardless which tissue the biopsy was made on. The list of these genes, involved in diabetes and/or with PPAR-g, are shown in Appendix I. The last type of reduction was to exclude all samples where no BMI-value or glucose-value for the donor was stored. The reason why this reduction is chosen is that these values can be interesting to have when the results are analyzed.

The selected dataset consisted of 35,540 transcripts from 71 samples. It is interesting to know the distribution of the expression values of the two target genes. The distribution of PPAR-g was that for approximately 80% of the chosen samples the expression value for PPAR-g was absent and approximately 20% was present. For INSR the distribution was that for approximately 60% of the chosen samples the expression value was absent and approximately 40% was present. The expression value marginal was not a common expression value for the two target genes.

5.2.2 Experiment 1: predicting the expression of PPAR-g using diabetes related genes

In this experiment an ANN was trained to predict the expression of PPAR-g. The prediction of PPAR-g was made using transcripts from the 55 unique genes in Appendix I, as input for the ANN, and as output the expression of PPAR-g was used. The dataset used for this experiment consisted of 147 transcripts from 108 different samples. 146 transcripts were used as input for the ANN leaving the PPAR-g

expression which was used as target output. The expression of the gene PPAR-g was marginal, M, for two samples. When only two of the 108 samples have the value M there are very few samples with this expression value compared with the number of samples having the expression value of absent or present for PPAR-g, and there is a large possibility that this is ignored by the network. Therefore these two samples were excluded from the dataset, and the output could be set to 0 for absent and 1 for present.

For deriving the training and test sets a nine-fold-cross-validation was used and the distribution of the expression values of target gene PPAR-g was kept to be about the same as for the large selected dataset described in Chapter 5.2.1. Therefore the expression value of PPAR-g was absent for 80% of the samples and present for 20% of the samples in the training sets, and 73% absent and 27% present in the test sets. The training sets consisted of the gene expression values of 146 transcripts from 95 samples and the test sets consisted of the gene expression values of 146 transcripts from 11 different samples. Thus the network had 146 input nodes, and was trained with 95 samples before tested with 11 samples. The network was trained with the training function trainscg and trainlm (see Chapter 5.1.3).

5.2.3 Experiment 2: predicting the expression of PPAR-g using a small set of arbitrarily chosen genes

In this experiment an ANN was trained to predict the expression of PPAR-g using a dataset of 147 transcripts, chosen arbitrarily from a dataset of 35,540 transcripts. The 147 arbitrarily chosen transcripts were used as input for the ANN, thus the number of

input nodes was 147. The output for the network was the expression of PPAR-g, set to 0 for absent and 1 for present. The number of samples for this prediction was 67.

The training and test sets were derived by a six-fold-cross-validation. Due to results of experiment 1 it was investigated if the size of the training and test set had an influence of the network performance the network was trained and tested with datasets of different sizes. The trained network was tested on test sets consisting of only 7 test samples and larger test sets consisting of 13 test samples. The network was trained with the training function trainscg and trainlm respectively.

When testing the network with the test sets consisting of 7 samples the distribution of the expression value of the target, PPAR-g, was absent for 73% of the test samples and present for 27%, and the expression value was absent for 80% of the training samples and present for 20%. When testing the network with the test sets consisting of 13 samples the distribution of the expression value for the target was absent for 85% of the test samples and present for 15%, and the expression value was absent for 80% of the training samples and present for 20%

This experiment was performed with purpose of comparing with experiment 1. The transcripts in this experiment are arbitrarily chosen and because of that all of them can not be involved with diabetes. Therefore it is expected that the results form this experiment is not as good as the results form experiment 1 where transcripts from genes known to be involved with diabetes are used as input for the ANN.

5.2.4 Experiment 3: predicting the expression of INSR using diabetes related genes

An experiment trying to predict the expression of the insulin receptor INSR was made. This experiment was done as a compliment to the experiments predicting PPAR-g. The prediction of INSR was made using the transcripts of the genes found in Appendix I i.e. 147 transcripts from 55 genes. The transcripts of the genes in appendix I were used as input for the ANN, except for INSR which was used as output. As output the expression of INSR was set to 0 for absent and 1 for present. Training and test sets were derived by a seven-fold-cross-validation. Of the 106 samples the training sets consisted of 92 samples and the test sets consisted of 14 samples. The distribution of the expression value of INSR was absent for 64% of the test samples present for 36% of the samples. The distribution of the training sets was that 60% of the training samples had the expression value absent and 40% present. The network was trained with trainscg and trainlm respectively.

5.2.5 Experiment 4: predicting the expression of INSR using a small set of arbitrarily chosen transcripts

In this experiment a prediction of the expression of INSR was made using a dataset of 147 transcripts chosen arbitrarily from a dataset of 35,540 transcripts. This dataset of 147 arbitrarily chosen transcripts is the same dataset used for experiment 2.

The 147 chosen transcripts were used as input for the ANN. As output the expression of INSR was used, where the expression value absent was set to 0 and present was set to 1. A six-fold-cross-validation was used when training and testing the network. The training sets consisted of 57 samples and the test sets consisted of 10 samples. The distribution of the output was that for 60% of the test samples the expression value was absent and for 40% present. The distribution of the training sets was the same as for the test sets. The network was trained with the training function trainscg.

This experiment was performed with the purpose of comparing with experiment 3. The transcripts used as input for the ANN in this project were arbitrarily chosen and it is therefore not likely that all of them are involved with diabetes. It is expected that the results from experiment 3 is better than the results of this experiment, because in experiment 3 a prediction is made using genes known to be involved with diabetes as input for the ANN.

5.2.6 Experiment 5: predicting the expression of INSR using a larger set of arbitrarily chosen genes

In this experiment a prediction of the expression of INSR was made using a dataset of 1,000 transcripts, arbitrarily chosen from a dataset of 35,540 transcripts. Using 1,000 transcripts as input for the ANN means that the number of input nodes is 1,000. Having 1,000 input nodes is computationally too complex in this project, and therefore the number of input nodes is reduced. To reduce the dimensionality of the input for the ANN the transcripts were clustered by self-organising map (SOM). By clustering the transcripts with SOM an average expression profile for each cluster is

obtained. The average expression profiles from the different clusters are used as input to the ANN, and thereby the amount of input data is reduced.

It is the gene expression profile for the genes that are clustered. An expression profile for a gene can in this project be constructed due to the fact that the gene is measured in different samples. The expression of a gene in a sample can be either present (P), absent (A) or marginal (M) (see Chapter 2.7) and the profile for a gene is the expression for this gene over the different samples. The expression of a gene *g* over the different samples *s* can be thought of as an array *M(g, s)* where each position in the array is the expression of the gene *g* in sample *s*. Figure 5 illustrates how the expression for the genes over all samples can be visualised.

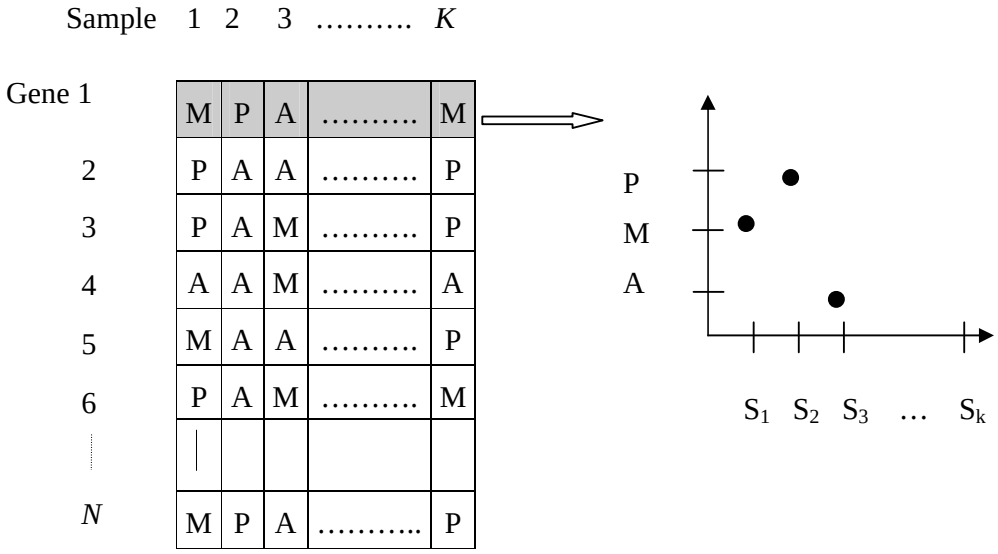


Figure 5. To the left is the expression of 1…*N* different genes during 1…*K* different samples. To the right is a visualisation of the expression profile of gene 1.

The reduction of the dimensionality of the input data is done by using the clustering algorithm SOM. Genecluster, described in Chapter 2.3.2, is used to produce and display SOMs of gene expression data, and is used here. To be able to cluster the different profiles with SOM it is necessary to set numeric values for the measurement

of the expression of a gene, thus these numeric values correspond to the values P, A and M. The numeric values that *M(g, s)* can take are shown in Equation 1. *M(g, s)* is an array of the expression value for a gene where each position is the array is the expression of a gene g in sample s.

$$M(g,s) = \begin{cases} 0 & \text{for A} \\ 0.1 & \text{for M} \\ 1 & \text{for P} \end{cases} \qquad (1)$$

where *M(g, s)* is the microarray value for gene *g* in sample *s*.

The numeric values chosen can of course be questioned. The thought here is that if a gene is absent, A, then the microarray value for a gene *g* in sample *s* is 0, if a gene is measured to be present, P, then the microarray value for a gene *g* in sample *s* is 1. The meaning of the value marginal, M, can be interpreted as the gene is only marginally present and it is not possible to say if the gene is either present or absent, although M is interpreted as closer to A than P and is therefore set to be 0.1. When using SOM it is important to consider the distance between the values. If the value of M would be close to 0.5 then SOM would interpret the expression of a gene where the expression value is M to be in between A and P. By setting the value of M to 0.1 then SOM is not interpreting M as closer to A than to P. The numeric values for A, P, and M is an arbitrary choice, and testing other numeric values could be done in a future work.

The average profile for the genes in a cluster is calculated by equation 2.

Let P_i be the average profile for cluster *i* and let c_i be the set of genes in cluster *i.* Then $P_{i,s}$ is the average expression value for the genes in cluster *i* for sample s and

$$P_{i,s} = \frac{\sum_{g=1}^{|C_i|} M_{g,s} \quad g \in c_i}{|c_i|} \tag{2}$$

where *g* denotes a gene.

Figure 6 illustrates how to reduce the input of the ANN by clustering the genes and creating an average profile for each cluster, where the average clusters will be used as input for the ANN. The genes showing no variation over the different conditions is excluded.

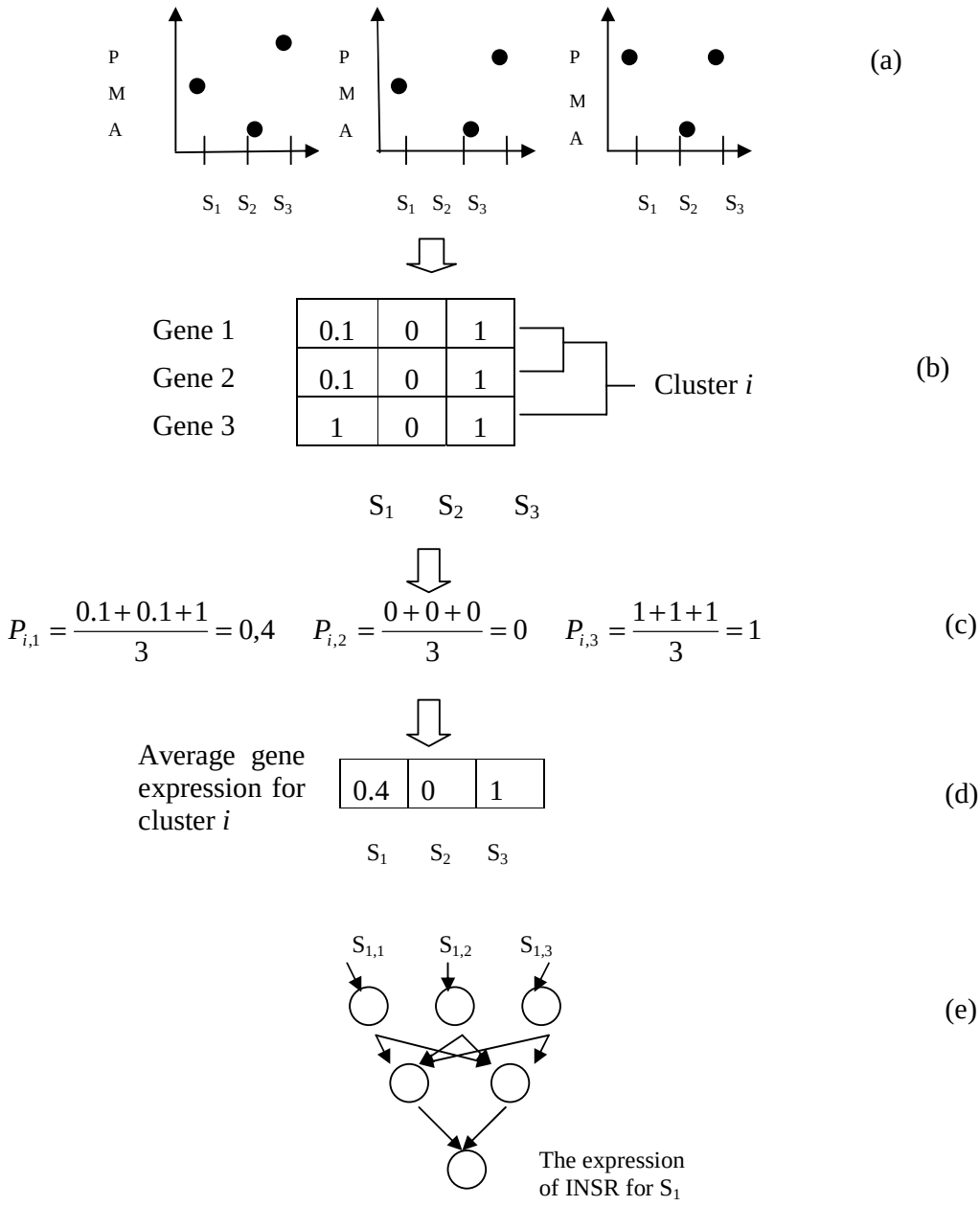


Figure 6. An illustrative example of the reduction of the amount of input data used for the ANN. (a) Gene expression profiles for the genes in a cluster *i*. (b) The numeric values for the expression of the genes. (c) Calculation of the average expression profile for the genes in the cluster. (d) The average profile created exchanging the genes in (b). (e) An example of the artificial neural network. In the first input node the average gene expression value for the genes for sample 1 in cluster 1 is used, in the second input node the average gene expression value for the genes for sample 1 in cluster 2 is used, and in the third input node the average gene expression for the genes for sample 1 in cluster 3 is used.

It is possible to choose the number of clusters in SOM. Here the number of clusters was arbitrarily chosen and the transcripts were clustered in respectively 12 and 24 clusters. The average profile for each cluster was used as input for the ANN, thus the number of input nodes for the ANN is the same as the number of clusters. As output the expression of INSR was used. The training and test sets were derived by a six-fold-cross-validation, where the training sets consisted of 57 samples and the test sets consisted of 10 samples. The distribution of the output was that the expression value of INSR was absent for 60% and present for 40% of the both training and test samples. The ANN was trained with the training algorithm trainscg.

6 Results

In this chapter the results of the experiments described in chapter 5.2 are presented. The architectures used for the experiments are presented *i*-*h*-*o* where *i* is the input, *h* is the hidden nodes, and *o* is the output. For example, 147-5-1 represents a network with 147 input nodes, 5 hidden nodes, and 1 output nodes.

6.1 Experiment 1: predicting the expression of PPAR-g using diabetes related genes

In this experiment a prediction of the expression of PPAR-g was made, using the expression of the transcripts of the genes associated with diabetes and PPAR-g in appendix I as input for the ANN. As output the expression of PPAR-g was used. Both the training function trainscg and trainlm were used for training the network.

For each time a network was trained and tested an accuracy value was calculated. The accuracy is calculated based on how many test samples the network can classify correctly. During the experiment different architectures were used, in order to find out the most appropriate architecture for this problem. For each architecture used in this experiment the network was trained and tested five times with the different training and test sets, and the mean accuracy was calculated for each architecture. Table 2 shows the different architectures and the mean accuracy for the network generated with the training function trainscg.

Network architecture	146-5-1	146-10-1	146-15-1	146-20-1	146-30-1	146-10-5-1	146-15-5-1
Mean acc.	0.73	0.73	0.73	0.73	0.73	0.73	0.73
Best acc.	0.73	0.73	0.73	0.73	0.73	0.73	0.73

Table 2. The upper row shows the network architectures, the center row shows the mean accuracy for each architecture, and the bottom row shows the best accuracy measured for each architecture.

Using the trainlm training function did not give any different results of the accuracy. The only difference was that the trainlm function generated slightly lower best-net-error values.

6.2 Experiment 2: predicting the expression of PPAR-g using a small set of arbitrarily chosen transcripts

In this experiment a prediction of the expression of PPAR-g was made.147 arbitrary chosen transcripts form a dataset of 35,540 transcripts were used as input for the ANN. To investigate if the size of the test set affects the results, test sets of two different sizes were used. Different network architectures were tried, and a network was for each architecture trained and tested five times with the different training and test sets. Table 3 shows the different architectures and the mean value of the accuracy for each architecture for the test set consisting of 7 test samples.

Network architecture	146-5-1	146-10-1	146-15-1	146-20-1	146-30-1	147-6-2-1	147-10-5-1	147-15-5-1
Mean acc.	0. 73	0. 73	0. 73	0. 73	0. 73	0. 73	0. 73	0. 73
Best acc.	0. 73	0. 73	0. 73	0. 73	0. 73	0. 73	0. 73	0. 73

Table 3. The results from experiment 2, testing the network on 7 test samples.

Table 4 shows the different architectures and the mean value of the accuracy for each architecture for the test set consisting of 13 test samples.

Network architecture	146-5-1	146-10-1	146-15-1	146-20-1	146-30-1	147-6-2-1	147-10-5-1	147-15-5-1
Mean acc.	0.85	0. 85	0. 85	0. 85	0. 85	0. 85	0. 85	0. 85
Best acc.	0. 85	0. 85	0. 85	0. 85	0. 85	0. 85	0. 85	0. 85

Table 4. The results from experiment 2, testing the network on 13 test samples.

Both the trainscg and the trainlm training functions were used for this experiment. The only difference between the results, obtained with the two training functions, is that trainlm gives slightly better best-net-error-values.

6.3 Experiment 3: predicting the expression of INSR using diabetes related genes

In this experiment a prediction of the expression of the insulin receptor INSR was made. The dataset used for this experiment was the 147 transcripts of the genes in Appendix I. Different architectures were tried and both the training function trainscg

and trainlm were used. The results, shown in Table 5, are the results generated with the training algorithm trainscg.

Network architecture	147-5-1	147-10-1	147-30-1	147-10-5-1	147-30-8-1
Mean acc.	0.64	0.67	0.73	0.68	0.71
Best acc.	0.64	0.79	0.79	0.79	0.86

Table 5. The results from experiment 3.

6.4 Experiment 4: predicting the expression of INSR using a small set of arbitrarily chosen genes

In this experiment a prediction of the expression of INSR was made using a dataset of 147 transcripts from the arbitrarily chosen genes as input and the expression of INSR as output. During the experiment different architectures were tried. When training the network the training algorithm trainscg was used and the results are shown in Table 6.

Network architecture	147-5-1	147-10-1	147-20-1	147-30-1	147-10-5-1	147-30-8-1
Mean acc.	0.63	0.65	0.6	0.72	0.63	0.65
Best acc.	0.8	0.9	0.8	0.9	0.9	0.8

Table 6. The results form experiment 4.

6.5 Experiment 5: predicting the expression of INSR using a larger set of arbitrarily chosen genes

In this experiment the expression of INSR was predicted using 1,000 transcripts arbitrarily chosen from 35,530 transcripts. To reduce the dimensionality of the input to the ANN, SOM was used to cluster the transcripts. The 1,000 transcripts were clustered in first 12 and then 24 clusters.

When deriving the clusters the average profile, that is the centroid, was used as input for the ANN and INSR was used as output. The number of clusters is also the number of input nodes for the ANN. Thus having 12 clusters generated in 12 inputs for the ANN, and having 24 clusters generated in 24 inputs for the ANN. Different architectures were used, although because of less input nodes in this experiment compared to the other experiments less hidden nodes were used. Table 7 shows the result with 12 input nodes for the ANN. Table 8 shows the result having a network with 24 input nodes. The training function trainscg was used for training the network.

Network architecture	12-5-1	12-10-1	12-5-2-1	12-10-5-1
Mean acc.	0.59	0.63	0.5	0.65
Best acc.	0.8	0.8	0.8	0.7

Table 7. The results from experiment 5, using 12 input nodes for the ANN.

Network architecture	24-5-1	24-10-1	24-20-1	24-5-2-1	24-10-5-1
Mean acc.	0.62	0.72	0.73	0.72	0.71
Best acc.	0.7	0.9	1.0	0.8	0.8

Table 8. The results from experiment 5, using 24 input nods for the ANN

7 Analysis and discussion

In this chapter the analysis and discussion follows. Chapter 7.1 analyses the results generated in Chapter 5. Chapter 7.2 discusses the results of the project and potential reasons for the generated results.

7.1 Analysis

In this chapter the results from the different experiments is discussed. Chapter 7.1.1 analyses the results from the predictions of the expression of PPAR-g and Chapter 7.1.2 analyses the results from the predictions of the expression of INSR using the transcripts of the genes in Appendix I and the small set of 147 arbitrarily chosen transcripts as input for the ANN. Chapter 7.1.3 analyses the results from the prediction of the expression of INSR using 1,000 transcripts.

Analysis of the results is made by comparing a prediction made by the ANN with if the network always predicts the value of the expression that is in majority.

7.1.1 Experiment 1 and 2, predicting PPAR-g

For experiment 1, predicting the expression of PPAR-g using diabetes related genes, the accuracy of the test sets was 73% for all architectures tried. When looking at the expression of PPAR-g over the different samples it is observed that in about 80% of the samples the expression value is absent, A. When creating a test set it is important that the distribution of the expression values of PPAR-g is approximately the same in

the training set and in the test set. In the test sets used for experiment 1 three out of eleven samples had the value of present for PPAR-g, and thus eight of eleven samples had the value of absent. Eight out of eleven is 73%, which is the accuracy received if the value absent always is assumed. When comparing 73% to the accuracy of 73% in Table 2 it is possible to draw the conclusion that the network always predicts the expression value of PPAR-g to be absent, and by doing so is correct in 73% of the test cases.

To further investigate if the received results indicate that the network always predicts the value of absent the performance function (see Chapter 5.1.4) was changed. From classifying an output between 0 and 0.5 as absent and an output between 0.5 and 1 as present the value of the classification of the output, 0.5, was changed to see if this could generate better results. The change of the value of 0.5 did not generate different results and thus the conclusion is that the network always predicts the expression value to be absent and by doing so is correct for 73% of the test samples.

A test where the distribution of PPAR-g is around 50/50 would have been interesting to perform in order to further investigate the influence the distribution of the output has on the results. This has not been done and is considered future work.

The same conclusions as above can be drawn when analyzing the results of experiment 2, predicting the expression of PPAR-g using a small set of arbitrarily chosen transcripts. In experiment 2 a trained network was first tested on test sets consisting of 13 samples. The distribution between the expression values absent and

present in the test set was 85/15, which means that if the expression value absent always is assumed a network would make a correct prediction in 85% of the test cases. The accuracy of the tested networks was 85% and thus it is possible that the network always predicts the expression value to be absent and by doing so is correct for 85% of the test samples. . To exclude the risk of having trained the network on too few samples compared with the number of test samples, a prediction with a larger training set and a smaller test set, where the test sets consisted of 7 samples, was made. For the smaller test sets the accuracy was 0.71. The distribution of PPAR-g in the test sets was that 71% of the expression values for the 7 test samples were absent, and thus the conclusion is the same as for experiment 2.

7.1.2 Experiment 3 and 4, predicting INSR

In experiment 3 a prediction of the expression of INSR using diabetes related genes, was made. The test set consisted of 14 test samples, of which nine had the expression value absent for INSR. The distribution of the expression value of INSR in the test sets was 64% of the samples had the expression value absent for INSR. If this experiment would generate with the same result as the former experiments the accuracy of a network should be around 64%. The value of 64% is compared to the values of accuracy in table 8. The comparison shows that a network architecture of 5 hidden nodes is not suitable for this kind of problem, as the accuracy for this network is 64%. The best results of the prediction of INSR are received with a network architecture of one hidden layer with 30 nodes where the accuracy was 73%.

The results in experiment 3 can be compared with experiment 4, the prediction of the expression of INSR using a small set of randomly chosen genes. As the distribution of the expression value of INSR is 60% absent and 40% present for the samples the prediction would generate an accuracy of 60% if the expression value absent is always predicted. Comparing 0.6 to the results in Table 6 shows that the results by an ANN are better than if the network always predicts the expression value to be absent. When comparing the results of Table 5 and 6 it is not possible to say that the results of one experiment is better than the results of the other.

7.1.3 Experiment 5: predicting the expression of INSR using a larger set of arbitrarily chosen genes

In this experiment the expression of INSR was predicted using a larger set of 1,000 arbitrarily chosen genes. SOM was used to decrease the dimensionality of the input for the ANN, creating an average profile for each cluster. If a network always predicts the expression value to be absent this would generate an accuracy of 0.6. The results from the prediction using 12 average profiles as input for the ANN presented in Table 7 are not equal compared to a prediction where the network always predicts the expression value to be absent. In Table 7 it is shown that the value of accuracy for the different architectures ranges from 0.5 to 0.65.

Reducing the dimensionality to 24 dimensions generated better results. These results are shown in Table 8 and the accuracy ranges from 0.62 to 0.72. When comparing the results shown in Table 8 to a prediction where the network always predicts the

expression value to be absent, that is having an accuracy of 0.6, the results in Table 8 are better.

7.2 Discussion

Predicting the expression of PPAR-g with the help of an ANN is not easy to do using the gene expression data used for these experiments. The predictions made of PPAR-g are not better than if a network always predicts the expression value that is in majority. The obtained results indicate that the network never finds any general pattern in the input data. There can be different reasons for these results. It is possible that the distribution of the expression of PPAR-g makes it hard for the network to find good patterns. The distribution of the expression of PPAR-g was that for 80% of the samples the expression value was absent, and for 20% of the samples the expression value was present. With a distribution like this it is clear that the network more or less ignores that the target in 20% is present and thus always predicts the expression value absent.

Because all results obtained when predicting the expression of PPAR-g are the same it is not possible to draw conclusions about which architecture suits the problem the best.

The predictions of INSR generate results better than an accuracy obtained if a network would always predict the expression value that is most common. The results of the predictions of INSR make it hard to discard ANN as a tool for predicting the

gene expression of individual genes. The question asked is instead why the predictions of PPAR-g gave such bad results compared with the predictions of INSR.

Above the distribution of the expression value of PPAR-g is given as a possible reason. It is also possible that the information intrinsic in the input data for the ANN not is enough for making a prediction of PPAR-g better than random guessing. It can be that the genes used as input for the ANN does not contain enough information for an ANN to be able to predict the expression of PPAR-g. For some of the predictions made the dataset used as input for the ANN was chosen arbitrarily. Even if the data set was chosen arbitrarily there is a possibility that the chosen genes cannot contribute with enough information for an ANN to make a prediction of PPAR-g better than a prediction by random guessing. However the genes can contribute with enough information for predicting INSR with the help of an ANN.

Limitations were made when selecting the data. Limitations can always be a reason for generating results not as inspiring as hoped for. One limitation made was to select a subset of genes from the database. Because of the large amount of data stored in the database it was necessary to find a way to reduce the data. The first reduction made was to exclude all samples where the biopsy was not made on adipose, liver or muscle tissue. Next a variation filter was used. This variation filter excluded all genes not showing a 5% variation over the tissues of interest (adipose, liver and muscle). When using a variation filter there is a risk of excluding genes that could be contributing with valuable information if they where used. To lower the risk of excluding genes that could be important it was decided to keep genes involved with diabetes and/or PPAR-g, regardless of which tissue the biopsy was made on, shown in Appendix I.

Choosing to keep these genes, even if they do not show a 5% variation, lowers the risk of excluding important genes, but only in the cases where the importance of the genes is already known. The variation requirement is low which decreases the risk of excluding important genes. However there is still a possibility of excluding genes we do not know the value of today by using the variation filter. The risk of excluding genes important for predicting the gene expression of PPAR-g is considered greater than excluding gene important for the predicting the gene expression of INSR. This because there is not as much knowledge about PPAR-g as there is about INSR today.

The predictions of the expression of INSR all resulted in predictions better than random guessing. When analyzing the results it could be expected that the results of the prediction made with the transcripts of the genes in Appendix I would be better than a prediction made with 147 randomly chosen transcripts. The genes in Appendix I all have a connection of some kind to diabetes and to PPAR-g. Having genes known to contribute with information should generate in better results than randomly chosen genes. However this was not the case. A conclusion drawn form this can be that INSR is not involved in diabetes, however this conclusion is not very likely. A possible explanation can be that the chosen genes in Appendix I do not affect the expression of INSR. The genes in Appendix I are all involved with diabetes or PPAR-g in some way but not all of them are involved in the same regulatory pathway.

The best results are generated when predicting the expression of INSR using 1,000 transcripts clustered together in 24 clusters. These results indicate that the more genes used for the prediction the more valuable information is intrinsic in the input and more the general patterns are therefore found by the ANN. Another limitation

concerning the data can therefore be that only a small part of the selected transcripts described in Chapter 5.2.1 were used in the predictions. Of the 35,540 transcripts selected only at the most 1,000 were used for a prediction. Due to implementation it was not possible to handle larger datasets.

For the prediction of INSR using the transcripts of the genes in Appendix I and the prediction of INSR using the 147 arbitrarily chosen transcripts an architecture of 147-30-1 gave the best results. For the predictions of INSR using a larger set of arbitrarily chosen genes, and reducing the dimensionality of these genes by SOM, it is not possible to draw conclusions about which architecture suits the problem the best. To be able to draw conclusions about which architecture is best for the problem further experiments must be done.

It is possible that another experimental setup for this project could generate in better results. As discussed in Chapter 5.1 it is hard to find the architecture appropriate for a certain problem. Different architectures have been tried during this project, but there is always a possibility of finding an architecture better suited for this problem.

There are different backpropagation algorithms to choose between, and maybe another backpropagation algorithm would be better than the ones used in this project. In this project two backpropagation algorithms were used, called trainlm and trainscg in Matlab. The trainscg generated in better results than trainlm, and is supposed to be good for pattern recognition.

The conditions for termination for the training functions can be changed. The conditions for termination of the training functions trainscg and trainlm were not changed in this project. It is possible that by changing the conditions for termination the network is trained longer and thus the obtained results can be different.

There are different transfer functions to choose between in the Matlab toolbox. By testing other transfer functions it may be possible to obtain different results. The log-sigmoid transfer function was used between the nodes of the hidden layer and the output nodes. As described in Chapter 5.1.2 the log-sigmoid transfer function returns an output value of 0 only when the net input is minus infinity, and the output value of 1 only when the net input is infinity. According to Mehrotra et al., (1997) it is preferable to use a smaller value (1-*t*) instead of 1, and a larger value *t* instead of 0 for the desired output. For this project the desired output was set to range between 0.05 and 0.95. By changing the value of *t* it may be possible to obtain better results.

When clustering the genes in order to get an average profile for each cluster numerical values had to be set for the expression values absent, present and marginal. Having the real value of the expression instead of the distinct values of absent, present and marginal could result in better clusters, leading to a better prediction. Also when using SOM one has to choose the number of clusters the data is split into. There are no directions for choosing the number of clusters. In this project the results with 24 clusters resulted in better results than 12. To find out which number of clusters is optimal for this problem, further investigations have to be made.

It is also possible to see things in a totally different way. There is a possibility that the results may indicate differences between genes. The behavior of one gene may not be suitable for predicting the expression of this gene with ANN, whereas the behavior of another gene makes this gene suitable for finding some general patterns. In this project only two genes were tested. Different results were received which indicate that the method is worth testing on more genes.

8 Conclusions and future work

The hypothesis for this project is that an ANN using gene expression data as input predicts the approximate expression of an individual gene. The two example genes used for testing the hypothesis is PPAR-g and INSR. If the prediction made by the ANN is clearly better than a prediction made by random guessing, then it is not possible to falsify the hypothesis.

The results obtained when predicting the target gene PPAR-g does not support the hypothesis. The results when predicting PPAR-g is not better than if a network constantly predicts the value of the expression that is in majority. The results obtained when predicting the target gene INSR however, support the hypothesis.

The conclusions of this project are that it is not possible to reject using ANN as a tool for predicting the gene expression of individual genes even if it, at this point, not is possible to say that using ANN for this purpose is reliable. There are disadvantages in using ANNs. One of the disadvantages of using ANNs is that an ANN requires a lot of samples to be trained on. The more samples the bigger the chances are for the network to be able to generalize. Another disadvantage is the ability to interpret the results generated by an ANN. It is well known that a feed-forward network is like a black-box, it is hard to understand how the results are generated.

Despite these disadvantages there are advantages of using ANN. ANNs are known to be effective for pattern recognition and for finding non-linear relationships. The

advantages does that ANN should have good possibilities for finding regulatory relationships. As described in Chapter 3 Khan et. al. (2001) tried to train an ANN to distinguish between four types of cancers, by using the gene expression data as input for the ANN. The results of their work were satisfying, and the question is why the results of this project were not as satisfying. An answer to the question could be that generating a prediction of one gene is a more detailed task compared to distinguish between four types of disease states. It may not be possible to find information in the data general enough for an ANN to make a good prediction of the expression of a gene.

The capacity of ANNs for predicting the gene expression of individual genes needs to be further investigated. As future work it would be of interest to predict the expression of all the genes in Appendix I individually. The results of these different predictions would help to draw further conclusions about the capacity of ANNs for predicting the expression of one individual gene.

Another interesting work would be to make another prediction of PPAR-g and this time use a distribution between the expression values absent and present closer to 50/50 in the training sets and test sets. By carrying out an experiment where the distribution of absent and present is 50/50 it is possible to draw further conclusions about the experimental design used in this project.

It would also be interesting to include more genes in the datasets used for the predictions. In this project the largest dataset used for a prediction was 1,000

transcripts. This dataset was derived from a larger dataset consisting of 35,540 transcripts. The prediction made using the 1,000 transcripts clustered in 24 clusters was the prediction that generated in the best results. By using a larger dataset than 1,000 transcripts it is possible investigate the value of the results in experiment 5. If a prediction with a larger dataset would generate in better results then the chance for the ANN of find general patterns good enough for predicting the expression of a gene is larger by using a larger dataset.

If it by further experiments is possible to get results where the hypothesis of this project is supported, the next step would be to interpret the ANN. By interpreting the ANN it may be possible to understand which factors can be involved with the gene. It may be possible to get further knowledge about which factors have an influence on the target gene and which factors are influenced by it. Interpreting ANNs is not easily done. Trained ANNs can perform well, but a common problem is that the decisions behind their classification cannot be found easily (Keedwell et al., 2000). This is often a problem within for example data mining, where it is important to have symbolic rules or other forms of knowledge structures (Keedwell et al., 2000).

References

Ando T., Honda H., Hanai T., & Kobsyashi T., (2001), Prognostic Prediction of Lymphoma by Gene Expression Profiling using FNN, *Genome Informatics*, No. 12, pp. 247-248

Aranda A., & Pascual A., (2001), Nuclear Hormone Receptors and Gene Expression, *Physiological Reviews*, Vol.81, No 3

Bicciato S., Pandin M., Didone' G., & Di Bello C., (2001), *Analysis of an associative memory neural network for pattern identification in gene expression data*, Zaki M. J., Toivonen H., Wang J. T-L., Eds., *Proceedings of the ACM SIGKDD Workshop on Data Mining in Bioinformatics*, University of Padova and Cittadella Hospital, Italy, pp. 22-30. ACM

Bigus J. P., (1996), *Data mining with neural networks- solving Business Problems-from Application Development to Decision Support*, McGraw-Hill

Birnbaum K., Benfey P. N., & Shasha D. E., (2001), *cis* element/transcription factor analysis (cis/TF): A method for discovering transcription factor/*cis* element relationships, *Genome Research*, Vol 11, pp. 1567-1573

Brazma A., & Vilo J., (2000), *FEBS Letters*, pp. 17-24

Campbell N.A., Reece J.B., & Mitchell L.G., (1999), *Biology* (Fifth Edition), Addison Wesley Longman

Chawla A., Repa J. J., Evans R. M., & Mangelsdorf D. J., (2001), *Science*, Vol. 294, pp. 1866-1870

Chen J, Sadowski HB, Kohanski RA, & Wang LH., (1997), Stat5 is a physiological substrate of the insulin receptor, *Biochemistry*, Vol. 94, pp. 2295-2300

Debouck C.,& Goodfellow P. N., (1999), *Nature Genetics Supplement*, Vol. 21

D´haeseleer, P., Liang, S., & Somogyi, R., (2000), Genetic network inference: from co-expression clustering to reverse engineering, *Bioinformatics* Vol. 16, No. 8, pp 707-726

Diedrich J., (1990), *Artificial neural networks Concept Learning*, IEEE Press

Dopazo, J., Zanders, E., Dragoni, I., Amphlett, G., & Falciani, F., (2001), Methods and approaches in the analysis of gene expression data, *Journal of Immunological Methods*, Vol 250, pp.93-112

Eisen, M., Spellman, P., Brown, P., & Botstein, D., (1998) Cluster analysis and display of genome-wide expression patterns, *Proceedings of the National Academy of Science, USA*, Vol. 95, pp. 14863-14868

Gruvberger S., Ringnér M., Chen Y., Panavally S., Saal L. H., Borg Å., Fernö M., Peterson C., & Meltzer P. S., (2001), Estrogen receptor status in breas cancer is associated with remarkably distinct gene expression patterns, *Cancer research*, Vol 1, pp. 5979-5984

Harris M., (1985), Prevalence of noninsulin-dependent diabetes and impaired glucose tolerance, in National Diabetes Data Group: Diabetes in America, *American Diabetes Association*, No 85, pp. 1-31

Jones A. B., (2001), Peroxisome Proliferator-Activated Receptor (PPAR) Modulators: Diabetes and Beyond, *Medicinal Research Reviews*, Vol. 21, No.6, pp. 540-552

Kanehisa, M., (1996), Toward pathway engineering: a new database of genetic and molecular pathways. *Science & Technology Japan*, No. 59, pp. 34-38

Keedwell E., Narayanan A., & Savic D., (2000), Creating rules from trained neural networks using genetic algorithms, *International Journals of Computers, Systems and Signals*, Vol. 1, pp. 30-42

Khan J., Wei J. S., Ringer M., Saal L. H., Landanyi M., Westermann F., Berthold F., Schwab M., Antonescu C. R., Peterson C., & Meltzer P. S., (2001), Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nature Medicine*, Vol. 7, pp. 673-679

Mehrotra K., Mohan C.K., & Ranka S., (1997), *Elements of Artificial Neural Networks*, MIT Press

Müller-Wieland D., Kneble B., Avci H., Lehr S., Ladues M., Ristow., Krone W., & Kotzka J., (2001), Insulin-regulated transcription factors: molecular link between insulin resistance and cardiovascular risk factors, *Internal Journal of Obesity*, Vol 1, pp. 35-37

Nardulli A. M. & Shapiro D. J., (1992), Binding of the estrogen receptor DNA-binding domain to the estrogen response element induces DNA bending. *Molecular and Cellular Biology*, Vol. 12, pp.2037-2042

Olefsky J. M., & Saltiel A. R., (2000), PPAR-g and the Treatment of Insulin Resistance, *TEM*, Vol. 11, No. 9, pp. 362-367

Parker M. G., (1991), editor. *Nuclear Hormone Receptors*. Academic Press

Persidis A., (2000), *Nature Biotechnology*, Vol. 18, pp.237-238

Patterson D. W., (1996), *Artificial neural networks-Theory and Applications*, Simon &Schuster Pte

Robinson-Rechavi M., Carpentier A. S., Duffraisse M., & Laudet V., (2001), How many nuclear hormone receptors are there in the human genome?, *TRENDS in Genetics*, Vol. 17, No. 10

Robyr D., Wolffe A. P., & Wahli W., **(**2000), Nuclear Hormone Receptor Coregulators In Action: Diversity For Shared Tasks, *Molecular Endocrinology*, No. 14, pp. 329-347

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

Tenbaum S. & Baniahmad A., (1992), Nuclear Receptors: Structure, Function and Involvement in Disease, *Int. J. Biochem. Cell Biol*., Vol. 29, No. 12, pp. 1325-1341

Tamayo P., Solonim D., Mesirov J., Zhu, Q., Kitareewan S., Dimitrovsky E., Lander E., & Golub T., (1999), Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, *Proceedings of the National Academy of Science USA*, Vol. 96, pp.2907-2912

Toh H., & Horimoto K., (2002), Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modelling, *Bioinformatics*, Vol 18, No 2, pp. 287-297

Vamecq J., & Latruffe N., (1999), Medical significance of peroxisome proliferator-activated receptors, *The Lancet*, Vol. 354

Witten I. H., & Frank E., (2000), *Data mining - practical machine learning tools and techniques with java implementations*, Morgan Kaufmann

Appendix I

The appendix shows the names of the genes associated with diabetes and PPAR-g in the left column of the table and the corresponding designations of the transcripts to the left in the table.

Genes	Transcripts	Genes	Transcripts
RAD1	1008_f_at	INS	35723_at
GHRHR	1123_at	GAD1	36129_at
IGF2	1288_s_at	HLA-A	36679_at
IGF2	1367_f_at	HNF4A	36722_s_at
GHRH	1390_s_at	HNF4A	36723_at
IGF1	1501_at	NEUROD1	36768_at
TNFRSF1A	1563_s_at	IGF2	36782_s_at
INSR	1572_s_at	RAD1	36857_at
IGF2	1591_s_at	SLC2A3	36979_at
TNF	1852_at	PPARG	37104_at
IGF1	1975_s_at	GAD1	37183_at
IGF2	2079_s_at	PCK2	37188_at
TNF	259_s_at	LMNA	37377_i_at
GNAS	31604_at	LMNA	37378_r_at
GNAS	31873_at	HLA-A	37383_f_at
GNAS	31907_at	IGF2	37407_s_at
SLC2A3	31997_at	HLA-A	37421_f_at
GAD2	32279_at	GNAS	37448_s_at
IGF2	32582_at	GNAS	37449_i_at
HLA-A	32878_f_at	GNAS	37450_r_at
INSR	33162_at	SLC2A2	38238_at
ASIP	33522_at	IL6	38299_at
TCF2	33621_at	CEBPB	38354_at
POMC	33711_at	TCF2	38506_at
LEPR	34266_at	IGF1	38737_at
LEPR	34267_r_at	MAPK8IP1	38775_at
ENPP1	342_at	SUPT4H1	39440_f_at
ENPP1	343_s_at	CAP	39733_at
SOD2	34666_at	IPF1	400_at

Genes	Transcripts
HLA-A	40369_f_at
HLA-A	40370_f_at
SLC2A1	40507_at
SUPT4H1	40536_f_at
PCSK1	40649_at
APM1	40657_r_at
APM1	40658_r_at
GPD2	41021_s_at
GPD2	41022_r_at
IRS1	41049_at
HLA-A	41237_at
MAPK8IP1	41279_f_at
MAPK8IP1	41280_r_at
SUPT4H1	41474_at
ADA	41654_at
GNAS	41752_at
SLC2A3	42945_at
GNAS	429_f_at
IGF2	43359_f_at
GNAS	43372_f_at
CEBPB	43806_at
GNAS	43832_at
IGF2	45259_at
GNAS	46155_at
GNAS	46156_at
GNAS	471_f_at
CAPN10	47416_at
PCK2	48623_at
HLA-A	49203_f_at
SUPT4H1	508_at
GNAS	50967_at
LMNA	51043_f_at
IGF2	51834_f_at
SLC2A3	52658_at
SOD2	54921_at
RAD1	55326_f_at
IGF2	55328_r_at
RAD1	55423_f_at
SLC2A3	56198_at
RAD1	56258_f_at
LEP	56955_f_at
RAD1	57153_f_at
HLA-A	57280_f_at
SUPT4H1	58643_at
LEPR	62761_i_at
MAPK8IP1	63444_f_at
RAD1	63705_f_at
IGF1	64305_s_at
LEPR	65957_at
GNAS	67045_r_at
GAD2	69177_at
APM1	69842_f_at
HLA-A	70198_f_at
CAP	70488_r_at
HLA-A	71787_r_at
LEP	72496_f_at
HLA-A	72736_f_at
SOD2	73312_s_at
PCK2	73397_at
MAPK8IP1	74541_at
HNF4A	74797_at
GNAS	75556_f_at
GNAS	75606_f_at
IGF2	767_at
IGF2	773_at
IGF2	774_g_at
HLA-A	77561_r_at
GHRL	80871_at
SUPT4H1	82276_f_at
GNAS	82731_f_at
GNAS	82735_f_at
GNAS	82927_f_at
GNAS	83906_at
GAD1	84447_at
HLA-A	85043_at
IRS1	850_r_at
IRS1	851_s_at
SUPT4H1	85718_f_at

Genes	Transcripts	Genes	Transcripts
PCK2	86118_at	LEPR	88091_f_at
SUPT4H1	87235_f_at	CAP	89431_r_at
IRS1	872_i_at	HLA-A	90501_r_at
SUPT4H1	87863_r_at	ADA	907_at
SLC2A2	88058_s_at	RAD1	90897_at
		CAP	935_at