

A Computational Method for Analyzing  
Microarray Gene Expression Data Using  
Support Vector Machines

Annette Höglund

December 8, 2000

Master's Thesis Project

### Abstract

Recent advances in the field of biotechnology enable analysis of global gene expression patterns at the mRNA level. Available databases with such data increase exponentially, which create new demands to develop software tools for data management, visualization, comparison and exploration. It has been recognized, in yeast, that genes with a similar expression pattern share common cellular functions. It is most likely that similarity analysis of expression data from other organisms, including humans, can reveal the function of not previously annotated genes, and illuminate groups of genes with related functions. Patterns seen in genome-wide expression experiments can be interpreted in different ways, some yet unknown. Analysis of microarray gene expression data has so far focused mostly on the clustering of gene expression vectors with similar patterns, using unsupervised machine learning methods based on clustering algorithms. The supervised machine learning method used here, is the knowledge-based analysis of microarray expression data using support vector machines. In this work an automatic program, using support vector machines on gene expression data, is developed. The knowledge-based method is further evaluated for function prediction and compared with an unsupervised prediction method.

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Background and Theory</b>	<b>6</b>
2.1	Functional Classification of Proteins . . . . .	6
2.2	DNA Microarrays . . . . .	7
2.2.1	Microarray Technology . . . . .	7
2.2.2	DNA Microarray Data . . . . .	8
2.3	Gene Expression and Analysis of Expression Patterns . . . . .	10
2.4	The Separation Problem . . . . .	11
2.5	An Introduction to Support Vector Machines and Other Methods used for the Analysis of Expression Data . . . . .	13
2.5.1	Unsupervised Learning Methods . . . . .	13
2.5.2	Supervised Learning Methods . . . . .	15
2.5.3	Support Vector Machines . . . . .	15
<b>3</b>	<b>Methods and Materials - Experimental Design</b>	<b>17</b>
3.1	DNA Microarray Data . . . . .	17
3.2	The SVM <sup>light</sup> : Support Vector Machine Used for Analysis . . . . .	18
3.2.1	SVM Training . . . . .	19
3.2.2	Cross Validation . . . . .	19
3.3	The Clustering Algorithm Used for Analysis . . . . .	20
3.4	Performance Measurements . . . . .	20
3.4.1	Cost Savings . . . . .	21
3.4.2	Matthew's Correlation Coefficient . . . . .	22
3.5	Description of the Program and Experimental Design . . . . .	23
3.5.1	Description of the Program . . . . .	23
3.5.2	Experimental Design . . . . .	24
<b>4</b>	<b>Results and Discussion</b>	<b>26</b>
4.1	The User Interface . . . . .	26
4.2	Results of Functional Classifications . . . . .	27
4.2.1	Results from Supervised Learning . . . . .	27
4.2.2	Results from Unsupervised Learning . . . . .	29
4.2.3	Effects of Defining the Functional Group Differently . . . . .	32
4.2.4	Anti Co-Regulated Genes . . . . .	34
4.2.5	Other Functional Classes Tested . . . . .	35
<b>5</b>	<b>Conclusions and Future Improvements</b>	<b>35</b>
<b>6</b>	<b>Acknowledgements</b>	<b>37</b>

## 1 Introduction

The genome is considered to be the key of life, since it contains all information needed in order to construct living organisms. Genes are encoded in the genetic material, the DNA (deoxy ribonucleic acid) or RNA in some cases. The coding parts of the DNA are called exons, and the non-coding parts are called introns. The exons contain all the precise information needed to build a protein. The introns are spliced out (mRNA splicing), and the exons are arranged in a specific order so that the desired protein is constructed. Different proteins play different roles and are needed at various stages of cellular differentiation and development.

As the genome projects proceed around the world, we are presented with an exponentially increasing number of completely sequenced genomes and proteins. So far entire genomic sequences of more than 40 organisms exist. In the mapping and sequencing of the human genome, over 1.1 million expressed sequence tags (ESTs) have been cataloged. These ESTs correspond to less than 50,000 human genes [Venter, 2000]. However, not much is known about the structure, function, expression and regulation of more than 80% of them. In the emerging area of functional genomics, which is the next phase in the human genome project, a major goal is to assign the function to as many genes as possible. The function of a gene is a broad definition; it can be anything ranging from a description of a protein being a reactant in a biochemical reaction to a description of a structural component of the cell.

There is an emphasized need for computational methods for function prediction of unknown genes, since the process of experimentally determining the function of a protein is very time-consuming. There are two main approaches to explore the function of a gene, where computational methods can be employed. First, sequence homology, when one look at the nucleic acid sequence and search structural domain motifs, thus providing clues to gene function. In this approach the sequence of a protein of known function is aligned with the sequence of a protein of unknown function, in order to identify possible functions for the protein. Second, one can study the expression pattern of a gene. The expression of genes is a complex and highly controlled and regulated process. Today there are a few methods for analyzing which genes are expressed under specific conditions.

The relatively recently developed high potential DNA microarray technology allows researchers to measure the expression levels of thousands of different genes in several experiments simultaneously [Brown et al, 2000]. The microarray approach allows us for the first time to get a global view on the transcription levels of many, or even all, genes in the cell during a series of experimental conditions. Measuring gene expression levels in different developmental stages, in different body tissues, different organisms, and under different experimental conditions, presents immense opportunity [Gerstein et al, 2000], to understand gene function, gene networks, biological processes and effects of medical treatments. One approach in analyzing the microarray data is to examine the extremes, the genes showing significantly differential expression patterns. This technique is particularly useful in screening for tumor markers or drug target candidates

[Gerold et al, 1999], and in expression profiling of tissues and cancer cell lines [Bittner et al, 2000] and [Alizadeh et al, 2000]. Another more holistic approach, is to analyze the entire repertoire of expressed genes and find ways to extract useful knowledge [Eisen et al, 1998]. This approach addresses the full potential of the genome-scale expression experiments. Today, microarrays are used for studying for example; developmental stages, pathologic processes, regulating factors, evolutionary relationships, and the metabolic machinery. Microarrays are also possible tools in the monitoring health and detecting diagnostic diseases. It is the great challenge that lies in analyzing and understanding gene expression data that was the igniting spark that started the project presented in this report.

A previously presented knowledge-based computational machine learning method [Brown et al, 2000], used for analyzing gene expression patterns, is described, evaluated, and further developed in this report. The method employs the theory of support vector machines (SVMs) [Cristianini et al, 2000]. SVMs are considered a supervised machine learning method because they exploit prior knowledge of gene function to identify genes of similar function. The tool for predicting function is constructed and tested using SVMs and expression data of the yeast *S. cerevisiae*, but intended to be used on expression data of several organisms such as mouse, rat and human in a near future.

Other computational approaches, based on predicting gene function using microarray expression data, employ clustering algorithms for partitioning the genes into sub-clusters. These types of methods are unsupervised and do not learn expression features before classification, instead they perform internal similarity analysis of the expression vectors and construct a matrix giving the measure of similarity between each pair of vectors. It is the definition of similarity that differs between the different clustering methods. One of the main goals in this field of research is to evaluate the existing methods for expression analysis and pattern recognition, develop new methods, and use the most power of each method to secure the future development of new and more accurate methods for predicting gene function. The work described in this report illustrates the fact that some functional groups of genes are expressed in a similar pattern, which is possible to discern when analyzing the expression data using SVMs. The knowledge of the characteristic expression patterns of functional classes of genes can be utilized in the annotation of unknown genes. All genes do not, however, obey the characteristic patterns and present a challenge for the interested and keen researcher developing computational methods in the field of functional genomics.

The question at issue is to improve the understanding of what microarray expression data can tell us, and use the knowledge to enhance the methods for making efficient and correct predictions of gene functions. The key step in analyzing gene expression data is the identification of groups of genes that manifest similar expression patterns over several conditions. Could expression motifs for functional groups be a possible outcome of the approach?

## 2 Background and Theory

### 2.1 Functional Classification of Proteins

Proteins play crucial roles in virtually all biological processes and are sometimes referred to as the machinery of life. Amino acids are the basic structural units of proteins. There is a repertoire of 20 different amino acids, all having a different structure. The amino acids are linked with polypeptide bonds to form polypeptide chains. The unique amino acid sequence of a polypeptide chain, is encoded in the genes of our genome, and will determine the three dimensional-structure (3D-fold), as well as the biochemical and cellular function of the protein.

In order to get a deeper understanding of the complex sequence of events occurring at the molecular and cellular levels, it is vital to elucidate the functions of a particular gene as well as the functions of a group of genes. There are some problems associated with functional classification of genes. The term 'function' in itself is vague. It is the specific biochemical structure of a protein that determines the function. The function of a protein can be described as a biochemical mechanism, e.g. a protein can be active in enzymatic catalysis. The function can also be described in terms of involvement in pathways or overall cellular role e.g. a protein can be involved in cellular metabolism or transport, it can be active in the protection against disease. A third way of describing protein function is to associate it with the phenotype; e.g. 'causes disease' such as cancer, or control of growth and cell differentiation etc. One protein can possess one or more functions, which makes the classification even more difficult. For example, the protease thrombin is primarily associated with blood clotting, but it also plays a role in cell receptor activation and neural development. Finally, there is a general confusion in the terminology used when describing function. There are several different schemes for classifying gene function. Some schemes aim the focus on classifying gene function based on the organisms ESTs (expressed sequence tags) that are expressed e.g. MIPS [Mewes et al, 2000] (yeast), EGAD (human) databases. Other schemes classify genes to subsets of functions across a variety of organisms e.g. KEGG [Ogata et al, 1999] (pathways), and ENZYME [Bairoch, 2000] (enzymes). There is an attempt going on, The Gene Ontology Project [Ashburner et al, 2000], to merge the different types functional classifications into one common source. This source will not be able to cover the full spectra of functional classifications, but it will hopefully make it easier to get a more comprehensive overview.

Approaches to predict the functional class of a protein of unknown function, have mostly focused on three areas. First, experimental determination of a protein function; second, recognition of function via homology searches, and third; measuring expression levels of a proteins. Homology-based analysis methods allow the prediction of a protein's fold based on sequence comparisons, perhaps leading to clues about the function. A lot of effort has been put into this area, which is still improving. Screening whole genomes for expression patterns is a fast procedure; the effective and productive analysis of the gene expression data is still in the early phases of development and has to be explored further.

Microarrays are used as a tool for predicting gene function, discussed further in Section 2.2. The classification of the functional roles of proteins in this way, is based on the systematic approach to investigate gene function by mapping the expression scripts in organisms. The gene expression pattern seen on microarrays is the end sum of all the processes encoded in the genetic script, and the reason for why cells have distinctive design and functional capabilities. Why is the gene expression closely related to the function? The response to environmental factors affects the behavior and functions of the cells, by changing the genetic script (mRNA expression). Comparing the expression patterns of two cells originating from different tissues e.g. cardiac muscle cell and kidney cell, they have different functions and express different sets of proteins.

## 2.2 DNA Microarrays

### 2.2.1 Microarray Technology

The DNA microarray technology has become a clear success and the gene expression data is starting to accumulate, as several research groups put tremendous effort into this field of research. The intelligent design of DNA microarray technology enables measurement of expression levels of thousands of genes in parallel in one single experiment. The DNA chips with arrays of hybridization spots provide scientists with a reproducible tool, which is fairly simple to use and also possesses high specificity and sensitivity. Previous analysis of expression data and biochemical experiments give indications that genes of similar function yield similar expression patterns in microarray hybridization experiments [Brown et al, 2000]. The biological significance of expression data is preferably analyzed and evaluated by using computational methods, due to the large amounts of data generated in one experiment.

DNA microarrays (or DNA chips) [Gerold et al, 1999] are small glass surfaces bearing arrays of discrete spots, where single-stranded fragments, or the probes, are available for hybridization, see Figure 1. Hybridization is the biological event, when one single stranded DNA fragment of a specific sequence, attaches to the DNA fragment with the complementary sequence. Hybridization, also called base-pairing, occur between single stranded fragments of DNA, mRNA and cDNA, to form double stranded fragments. The base A (adenine) pairs with base T (thymine) in DNA and U (uracil) in RNA, whereas base G (guanine) pairs with base C (cytosine) in both DNA and RNA. A tissue or culture specific mRNA population is amplified and fluorescently labeled to produce cDNA or cRNA target (sample) solutions that can be hybridized to the DNA chip.

There are several DNA chip formats, the size of the arrayed DNA fragments is one feature that differs between formats. Other things that distinguishes chip formats are, the chemistry and linkers used for attaching DNA to chip, and the different methods for hybridization and detection. The cDNA array format and the *in situ* synthesized oligonucleotide array format are two chip formats currently widely used.

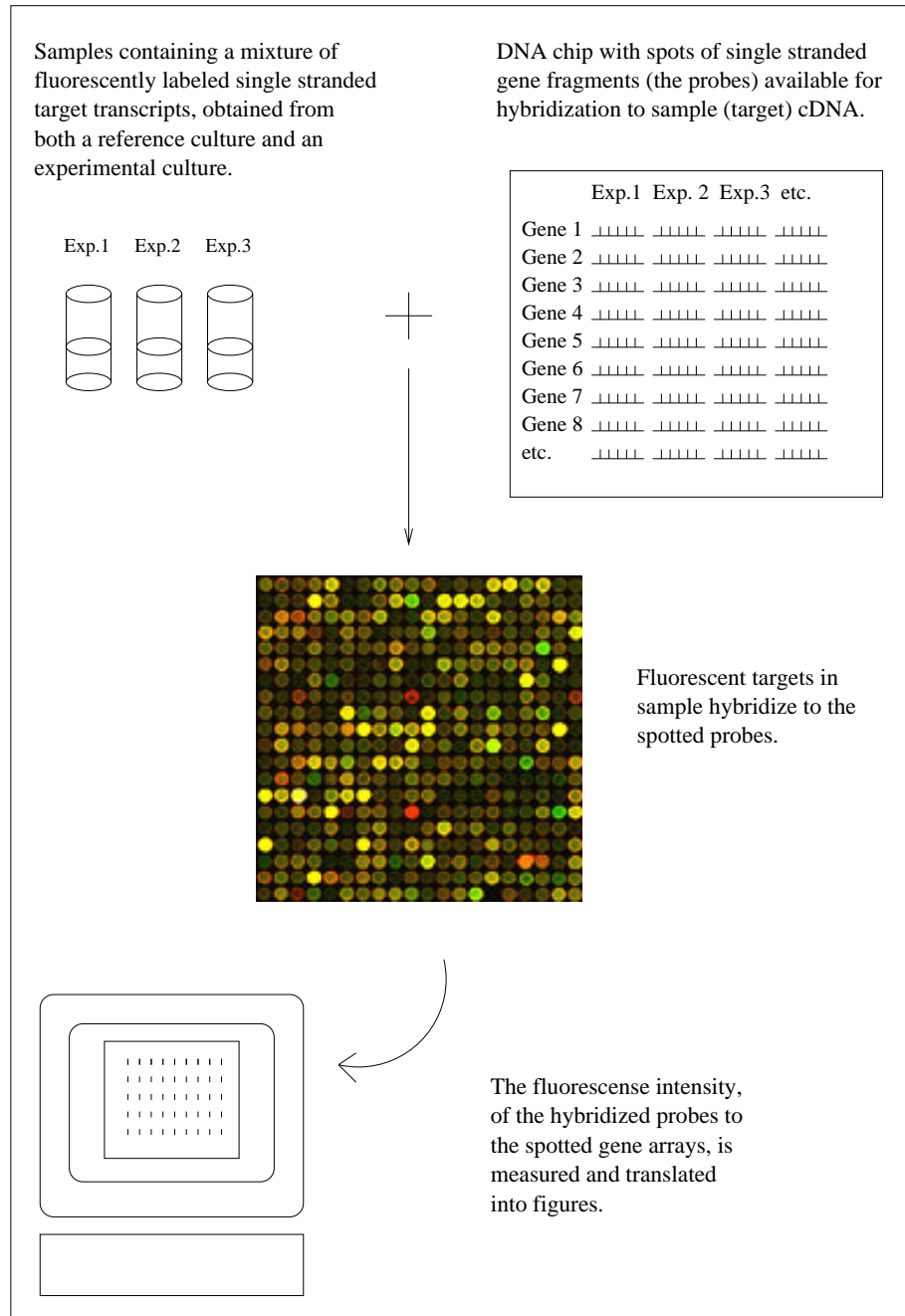
The cDNA microarrays, also called the Stanford microarrays, are produced by robotic deposition of single-stranded DNA fragments in spots, that are 50-150  $\mu\text{m}$  in diameter, onto a coated glass surface. This microarray format was developed at the Stanford University (<http://cmgm.stanford.edu/pbrown>). The arrayed spots are usually PCR-amplified inserts from cDNA clones, although long synthetic oligonucleotides can be used. DNA chips 3.6  $\text{cm}^2$  bearing up to 10,000 spots or more are common. The probes used, require less than 600 ng of mRNA per sample for a 10,000-spot DNA chip.

The other type of chip used is the oligonucleotide chip (or Affymetrix chip, see <http://www.affymetrix.com>) which generates a display of 65,000-400,000 DNA oligonucleotides that represent up to 9,000 genes on a 1.6  $\text{cm}^2$  glass surface. The oligonucleotide fragments are generated using DNA photolithography; ultraviolet light is shone through holes in masks in order to direct parallel and stepwise synthesis of oligonucleotides. What chip format is used in an experiment is determined by the researcher performing the experiment. Many labs construct their own chips, others buy their chips from some supplier.

There are a few things that may affect the accuracy of the chip data. For example, the targets on a spotted array are of different sizes, are spotted in different concentrations. Missing observations (gaps in the arrays) and outlier points are other factors that might be present in experimental data. The spotting of the target points is critical, the hybridization event depends highly on many experimental variables and measuring the intensity of the hybridization points adds further sources of error. Variations in GC content and secondary structural features can give rise to differences in hybridization affinities of target sequences for their respective probes. The areas most relevant to improve in the endeavor to generate homogeneous and noise free data, are to make the manufacturing of the DNA microarrays and the hybridization conditions as reproducible as possible, and also to make the absolute fluorescence intensities more comparable in and between experiments.

### 2.2.2 DNA Microarray Data

The microarray data obtained in parallel gene expression experiments provides two types of information. First, static information about gene expression is obtained, which tell us in what experiment or tissue the specific gene is expressed. Second, dynamic information about gene expression is obtained, that is, how the expression patterns between genes relate to one and another. The fluorescence intensity of the spots are measures of hybridization, which reflects the target concentration. In cDNA microarrays, this technology allows comparison of fluorescently labeled cDNA target populations from control and experimental tissues, such as diseased or drug-treated tissue, in two colors. The mixed dual-colored target solution can be hybridized to a single chip and then scanned at a separate wave length for each color, which makes it possible to assess differential gene expression. The two fluorescent probes, often green (Cy3) and red (Cy5), represent the reference level and the experimental level of gene expression. The differential gene expression is obtained by subtracting the reference level from



**Figure 1.** An overview of a microarray experiment and how the expression data is generated. The microarray is a small glass surface with spotted arrays of single stranded gene fragments. A tissue or culture specific mRNA population is amplified and labeled with fluorescent probes to produce cDNA or cRNA sample solutions that can be hybridized to the arrayed fragments on the DNA chip. The image of the arrayed fluorescent probes is captured and converted to expression ratios.

the experimental level. This procedure gives a measure of up- or down-regulation of gene expression. This type of measure of differential expression is used, since it is very difficult, or impossible, to assure that the level fluorescence is correct, that the amounts of sample are equal and that the method for analyzing the image is good. Image analysis is a critical event in obtaining the right expression ratios, see bottom part of Figure 1. Focusing the scanner so that all fluorescent light is measured, cutting out all background noise, and also detecting the whole target is an object of improvement in image analysis of microarrays.

The expression levels of the  $n$  genes of interest are measured under different conditions in  $m$  experiments. The data points form an  $m \times n$  gene expression matrix which can be scanned in the search for expression patterns. Each gene is represented by an  $m$ -element expression vector. The specificity and sensitivity of DNA microarray data relies on one core event in experimental biology, namely hybridization between spotted fragments on the chip and probes in sample.

### 2.3 Gene Expression and Analysis of Expression Patterns

In all life, gene expression is a complex and highly controlled event. One level of control of expression of eukaryotic as well as prokaryotic genes, is at the level of transcription, i.e. when DNA is being transcribed into mRNA copies. Eukaryotic genes are often silenced (repressed), unless they are induced (activated) by certain transcription-factor proteins that can bind to multiple control sites, e.g. promoter and enhancer regions, on the DNA strand. Some regulatory transcription-factor proteins bind to the DNA strand only if certain small regulatory molecules are bound to them, others only if such molecules are absent. When the transcription of a gene is initiated, mRNA copies of the gene are produced. In the case of eukaryotic mRNA, it contains introns that are to be removed, before the mature mRNA can be translated into protein. Regulation of splicing of the immature mRNA copy, and control of translation of the mature mRNA copy into protein, and protein (intein) splicing are thought to be similarly regulated for many genes, although proof exists showing that it is not true for all genes. Each one of the regulatory mechanisms has the simple objective of ensuring that the gene is expressed only under appropriate conditions. The subject of gene regulation is central to cell metabolism, to cell differentiation, and to the growth and development of organisms.

When dealing with microarray experiments, it is accepted that the concentration of mRNA of a particular gene in the cell relates, in most cases, to the amount of protein produced. The ratios obtained in the experiment are measures of the concentration, or rather, the change in concentration of mRNA in a sample. There are also techniques for detecting the absolute level of the mRNA concentration in a sample, e.g. SAGE (Serial Analysis of Gene Expression [Velculescu et al, 1995]), and northern blotting. The concentrations of different mRNA species are altered quickly, which enables the cell to make quick responses, such as altered metabolism, to meet environmental changes. The mRNA concentrations are very tricky to measure, due to the mRNA being a very unstable carrier of genetic information, which is easily destroyed by

mRNAs. Protein concentrations are possible to measure directly, but it is a rather time-consuming process. Some functional groups are known to be regulated by similar transcription-factor proteins, which means that whenever a specific regulating protein is present a certain group of genes are expressed, so-called co-expression. Other genes are never expressed at the same time, anti co-regulated.

The expression levels measured under several conditions in a microarray experiment, generate a unique expression profile, or expression vector, for each gene. The computational methods that are available for predicting gene function, use the relationships and the similarities in the expression profiles. Today, most approaches for analyzing expression patterns in chip data learns expression features in an unsupervised fashion. There is an alternative approach to this problem, which employs the supervised machine learning method. A knowledge-based method learns examples in the presence of a teacher signal, hence called supervised. The two methods mentioned plus some other approaches are described in section 3.4.

There are some experimental conditions that are expected to induce characteristic changes in expression levels of some groups of genes, examples of such conditions are, the diauxic shift [DeRisi et al, 1997] and yeast sporulation [Chu et al, 1998]. The diauxic shift is when the cell goes from the anaerobic state (fermentation) to the aerobic state (respiration). This event is expected to be correlated with widespread changes in expression of genes involved in fundamental processes e.g. carbon metabolism, protein synthesis, and carbohydrate storage. There is a transcription program for sporulation in budding yeast, which is precisely controlled.

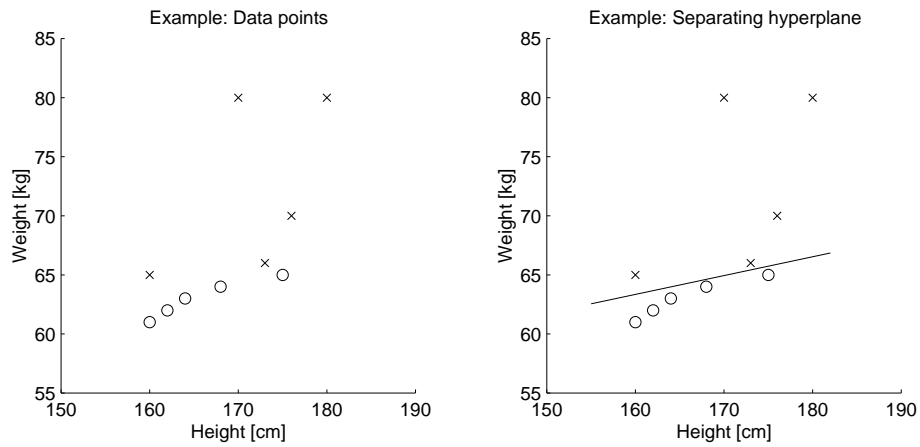
## 2.4 The Separation Problem

**The Problem:** A general description [Joachims, 1999], of the mathematical problem behind predicting functional classes of genes based on the expression data, is to be reviewed here. An example is discussed, in order to simplify the problem and make it more comprehensible. Suppose we are given the weight and height of a person, we want to try to determine their gender. If we are given some examples of weight and height connected to the gender, see Table 1, we can come up with an hypothesis that will help us predict the gender of a person. The hypothesis might be wrong in some cases but hopefully the best we can achieve from the data.

The mathematical problem is best described by plotting the example weights and heights in Table 1 into a two-dimensional coordinate system, see the left hand side in Figure 2. The difficulty lies in drawing a separating line, or hyperplane, such that it divides the points into two regions. One region containing only female points and the other only male points, see the right hand side in Figure 2. Several lines that separate the two genders can be drawn, the one we are looking for is the line that is at the furthest distance from any of the training points. This line or decision surface is also called the optimal separating hyperplane.

Ex.	Height	Weight	Gender
1	180	80	m
2	173	66	m
3	170	80	m
4	176	70	m
5	160	65	m
6	160	61	f
7	162	62	f
8	168	64	f
9	164	63	f
10	175	65	f

**Table 1.** Examples of height and weight corresponding to the gender of ten persons, five male, m, and five female, f.



**Figure 2.** An example of a problem separable by a hyperplane in the input feature space. The data points plotted to the left and the separating hyperplane drawn to the right.

The example described here, is linearly separable at once since it is a relatively simple example. The separating hyperplane is defined as a mathematical function, and using basic geometry, it is possible to determine to which group a point belongs.

In other cases, however, it might be more difficult or even impossible to separate the training examples. If the examples are not separable by a hyperplane in the *input feature space*, it might be possible to transfer the example vectors to a high-dimensional feature space where the vectors are separable, the solution is illustrated in Figure 3.

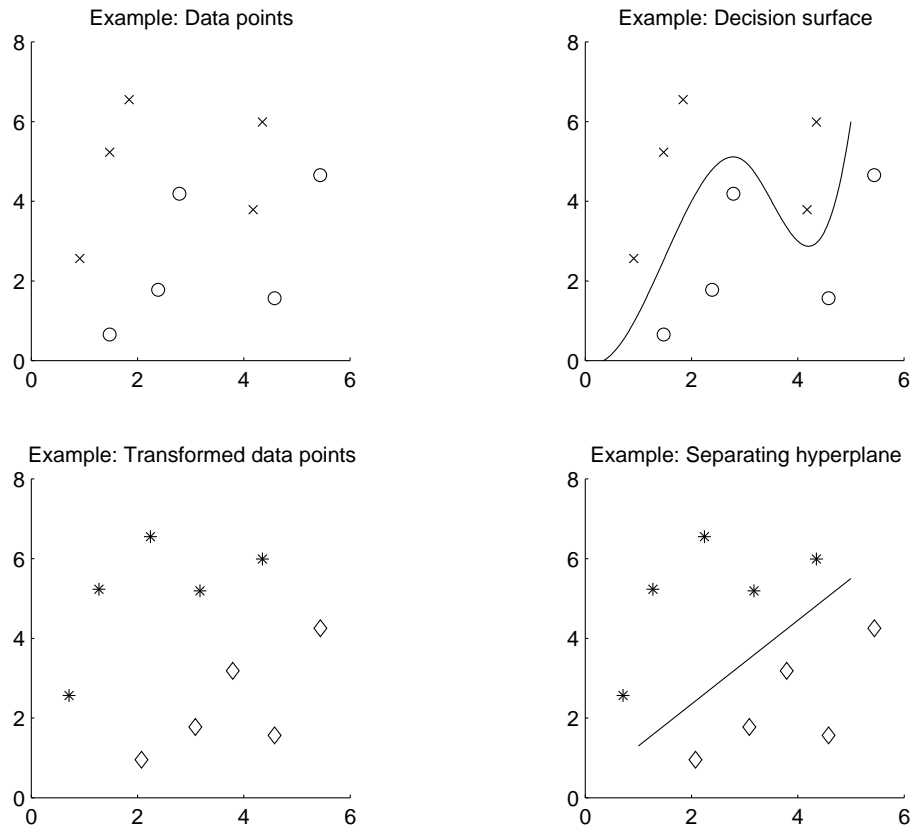
The separation problem in analyzing microarray gene expression data lies in finding the optimal separating hyperplane, which divides the expression vectors of members of a functional group from the vectors of non-members, in the best possible way. Separating microarray expression vectors is a good example of a complex problem, since the vectors are of high dimension and the relationships are not obvious at a first glance.

## 2.5 An Introduction to Support Vector Machines and Other Methods used for the Analysis of Expression Data

There are mainly two different ways to confront the problem of finding the optimal separating hyperplane. An unsupervised learning method does not take previous knowledge of functional classification in to consideration, hence no teacher signal for the method is present. A supervised method, on the other hand, learns in the presence of a teacher signal, e.g. training examples, and is said to be knowledge-based.

### 2.5.1 Unsupervised Learning Methods

Unsupervised learning methods try to predict classification without prior knowledge of functional class, i.e. not trained on previously defined examples, instead these methods begin with a definition of similarity. The similarity between genes in microarray data is often defined as a measure of distance between the expression vectors of the genes, e.g. measured by calculating a correlation coefficient between the vectors. The goal is then to partition or cluster the elements into subsets (clusters), using a clustering algorithm or a search algorithm. A typical clustering problem consists of  $n$  elements and a characteristic  $m$ -vector for each element. In the case of gene expression data, elements are genes and the vector of each gene is the corresponding expression ratio under some pre-defined experimental conditions. The clustering of the vectors should fulfill two criteria. First, homogeneity, the elements in the same cluster should be very similar. Second, separation, elements from different clusters have low similarity to each other. The method of clustering elements into clusters has many areas of application in biology as well as in other disciplines [Eisen et al, 1998]. Clustering is a type of internal analysis, where the analysis can be performed in mainly two ways. Hierarchical methods group data in a 'bottom-up' fashion, joining the most similar profiles into clusters first and then including the more diverse



**Figure 3.** An example of a problem not separable in the input feature space, see top left and top right. The male points are plotted as crosses, and the female points as circles. The input data is transferred into another feature space, where the data is separable by a hyperplane, see bottom left and bottom right. Here, the male points are plotted by stars, and the female by diamonds.

ones. These methods have the advantage that the number of clusters needs not be specified beforehand. These methods are widely used for evolutionary analysis, but the microarray data has no reason for organizing in bifurcating trees. Decisions made early in the process can not be undone and may be devastating for the final result. On the other hand, 'top-down' methods for partitioning data do not assume a tree structure, but these need a specified number of clusters before the algorithm can start. Today, there are algorithms developed to decide the number of cluster themselves. Examples of the 'top-down' method, applied to expression analysis, are the k-means and the self-organizing maps. Other examples of algorithms (not discussed any further here) used in similar types of problems are simulated annealing and graph theory approaches.

### 2.5.2 Supervised Learning Methods

Supervised learning methods, on the other hand, start with a definition of class, which specifies in advance the data elements that should cluster together as members and non-members. An example of a supervised learning technique, is the method using support vector machines (SVMs) for predicting class. Applying SVMs on gene microarray expression data, you begin with defining a group of genes known to be members and a group of genes known not to be members of a specific functional class. The SVMs learn the expression features of these training examples, and construct a model for discriminating members from non-members. If the expression features of the training set are learn-able, the knowledge-based method should be able to predict an unknown example based on the expression data. Supervised learning methods are trained to recognize a set of genes expected to show similarities in expression pattern, based on the knowledge we have today, whereas unsupervised methods are not trained at all.

Another example of a supervised learning method, is the approach based on neural networks [Stitson et al, 1996]. Most neural networks are designed to find a separating hyperplane, not necessarily the optimal one. Solutions using neural networks often start with a random line, which is moved until all points are on the right side of the line. This approach can lead to that many points lie very close to the line, i.e. a small margin classifier. SVMs on the other hand use geometric properties to exactly calculate the optimal separating hyperplane directly from the training data. SVMs also offer methods to deal with non-linearly separable cases, which can be useful when the training data is noisy.

### 2.5.3 Support Vector Machines

Support Vector Machines (SVMs) are considered a supervised computer learning method, see [Joachims, 1999] and [Cristianini et al, 2000], because they exploit prior knowledge of gene function to identify unknown genes of similar function based on the expression data. It is the mathematical features of the SVMs that make them attractive for this type of classification problems. Expression data is hard to separate because of the large data sets, complex expression vectors

and missing data points. Furthermore, SVMs are desirable because of their flexibility in choosing similarity function and ability to identify outliers. SVMs have been shown [Brown et al, 2000] to out-perform other competing machine learning techniques, such as Fisher's linear discriminant, Parzen windows, and two decision tree learners, in prediction the functional class of a gene based on the expression data. Compared with clustering methods, SVMs offer some advantages. First, all methods use distance as a measure of similarity, but SVMs can employ distance functions operating in extremely high-dimensional feature spaces, described in more detail soon. Second, using SVMs for classification is a knowledge-based method, whereas clustering methods are not.

Applying SVMs on expression data, examples of members and non-members of the functional class, have to be defined for the training procedure. The expression vectors might be difficult to separate linearly in the feature space they are mapped in, when the separating decision surface is non-linear. This problem can be solved by mapping the input element vectors into a feature space of higher dimension, where it is possible to construct a linear separating decision surface that corresponds to the surface in the input space, see Figure 3. This problem might seem to be trivial in theory, but it is still a hard computational problem. The hyperplane that is defined as the most optimal classifier with respect to the training data. The optimal hyperplane maximizes the minimum distance between the separating plane and any sample of the training data. The main problem that arises is to figure out how to handle the mapping of the vectors into the feature spaces. The mapping would be a problem since it would generate vectors of high dimensionality, hence require large storage requirements. This is elegantly avoided using the inner product of the vectors in the feature space [Stitson et al, 1996]. There is a set of kernels, or transformation functions, available for the construction of the support vectors. Depending on the kernel function used a different type of support vector machine is constructed. The support vectors are vectors that define the mapping of the input vectors into another feature space. Examples of the different kernels are, the linear, the polynomial and the radial basis function, discussed more in detail below. The two problems one faces when using SVMs for constructing a predictor are: first, to decide what kernels can be used for making the mapping into a higher feature space possible, and second, to choose the optimal set of options related to the kernel and the training examples. The available variables are options for learning and for the kernel. Learning options define in what way the SVMs should learn the training examples, and kernel options define what kernels can be used for constructing the support vectors.

Learning options used, are `-c` (float), which is a trade-off between training error and margin. Default for `-c` is 1000, in this experiment `-c` was varied over the interval [0.05..1000]. Another learning option is `-j` (float), the cost-factor by which training errors on positive examples outweigh errors on negative examples. Default value for `-j` is 1, `-j` was varied in the interval [0.5..10]. An important factor in SVM training is the choice of what kernel to use for the prediction model.

Kernel options are determined by defining the value of `-t`. A `t`-value of 0

uses a linear kernel (default), 1 uses a polynomial kernel, and 2 uses the radial basis function (rbf).

The polynomial kernel function is defined as

$$K(x_i, x_j) = (sx_i * x_j + c)^d$$

where the special case when  $d=1$  defines the linear kernel. The radial basis function kernel is defined as

$$K(x_i, x_j) = \exp\left\{-\frac{\|x_i - x_j\|^2}{\sigma^2}\right\}$$

The polynomial and radial basis kernels were shown [Brown et al, 2000], to be the best kernels for functional classification. Depending on the kernel chosen, there is a set of options to be defined. The -d (int), is the parameter  $d$  in the polynomial kernel, varied over [1,2,3]. The parameters -s and -r (floats) are parameters  $s$  and  $c$ , respectively, in the polynomial kernel and both were varied in the interval [0.25..0.5]. Parameter -g (float) is the parameter  $\gamma$  in the rbf kernel.  $\gamma$  is the inverted value of  $\sigma^2$ , and  $\sigma$  is varied in the interval [133..400].

### 3 Methods and Materials - Experimental Design

#### 3.1 DNA Microarray Data

The yeast DNA microarray data used is available on the Stanford web site (<http://rana.stanford.edu/clustering>). The microarray data represents the expression results obtained in  $m$  expression experiments containing  $n$  genes in each experiment. A gene expression pattern derived from one hybridization presents a picture of the state of a cell at a particular time. The expression pattern of a gene, obtained across a biological process or a collection of biological samples, presents the gene's characteristic expression profile. As described by [Eisen et al, 1998], the actual values in the matrix we used are the normalized logarithms of the difference in expression level, see formula below. Variable  $X_i$  is defined as the logarithm of the ratio of the expression level  $E_i$  for gene  $X$  in the experiment  $i$  divided by the expression level  $R_i$  of gene  $X$  in the reference state. The expression vector is normalized so that  $\vec{X} = (X_1, \dots, X_{79})$  has Euclidian length 1.

$$X_i = \frac{\log(E_i/R_i)}{\sqrt{\sum_{j=1}^{79} \log^2(E_j/R_j)}}$$

Induction of a gene is when a gene's expression level is turned up, in the expression matrix this is indicated by a positive sign. A repressed gene's expression level, on the other hand, is indicated with a negative sign in the matrix. The data points represent hybridization events using spotted arrays of gene fragments and samples from different experiments. The 79-element expression

vectors are generated from samples collected at various time points during the following experimental conditions; a) the diauxic shift [DeRisi et al, 1997], b) the mitotic cell division cycle, c) sporulation [Chu et al, 1998], d) temperature and e) reducing shocks. The Stanford data described above includes 2,467 annotated genes, it was used in the initial experiments and for the evaluation of the functional classification of the support vector machines.

A slightly different set of data was used for the prediction of ORFs of unknown function. The microarray data included both 2,467 annotated genes and 3,754 genes of unknown function, 6,221 in total. This data set spanned over 80 experiments, which included 65 of the 79 experiments used in the initial experiments, temperature and reducing shocks excluded. The last 15 experiments (no. 66 to 80) represent experimental data from the mitotic cell division cycle. The data set is available on the Stanford web site.

Class definitions made by the MIPS Yeast Genome Database that were used to train SVMs include six functional classes: tricarboxylic acid cycle (TCA, respiration, cytoplasmic ribosomes, proteasome, histones and helix-turn-helix proteins). The MYGD classifications are theoretic and based on biochemical and genetic studies of gene function. Many classes in MYGD, especially structural classes such as protein kinases, will be unlearn-able from expression data by any classifier. The first five classes were selected because they are expected, on biological grounds, to exhibit similar expression profiles. Furthermore, [Eisen et al, 1998] suggested that the mRNA expression vectors for these classes cluster well using hierarchical clustering. The sixth group, the helix-turn-helix proteins, was included as a control group. This group is not expected to show expression patterns that could be possible to recognize using any classifier.

There are some microarray expression data sets, representing human gene expression, that are available at the National Human Genome Research Institute's web site [http://www.nhgri.nih.gov/DIR/Microarray/Melanoma\\_Supplement/](http://www.nhgri.nih.gov/DIR/Microarray/Melanoma_Supplement/). The data set used for evaluating the prediction tool on human data, was expression data from human cutaneous melanoma.

### 3.2 The SVM<sup>light</sup>: Support Vector Machine Used for Analysis

The support vector machine, SVM<sup>light</sup>, was used in all experiments. SVM<sup>light</sup> is an implementation of Vapnik's support vector machine (SVM) for the problem of pattern recognition [Vapnik, 1995], available at the University of Dortmund web site ([ftp://ftp-ai.cs.uni-dortmund.de/pub/Users/thorsten/svm\\_light/current/](ftp://ftp-ai.cs.uni-dortmund.de/pub/Users/thorsten/svm_light/current/)). The optimization algorithm used in SVM<sup>light</sup> is described in [Joachims, 1999]. The algorithm has scalable memory requirements and can handle problems with many thousands of support vectors efficiently. The algorithm proceeds by solving a sequence of optimization problems lower-bounding the solution using a form of local search, a cost model may be used. The implementation is of high capacity, hence it is capable of handling many thousands of support vectors and several ten-thousands of training examples. Burges has written an

excellent tutorial on the use of support vector machines for pattern recognition [Burges, 1998].

SVM<sup>light</sup> consists of a training module (`svm_learn`) and a classification module (`svm_classify`). The training module receives a teacher signal and learns (builds a model) to recognize a functional class, the classification module can be applied to both known and not previously encountered examples to predict their functional class.

Using neural networks as a method for prediction functional class was not tested here but is a relevant approach for future work.

### 3.2.1 SVM Training

The training part of SVM<sup>light</sup>, `svm_learn`, creates a prediction model based on the training expression vector data set, described in section 4.1. The prediction model is then used by the classification module, `svm_classify`, on the test set in order to predict members and non-members of the actual functional class.

There are some available and variable options that can be set when calling the learning module, these options determine in what way the prediction model, used in the classification module, is going to be created, see section 3.5.3. The set of options that give the optimal prediction, is unique and has to be optimized individually for each functional class.

Training the support vector machines, with the different options, is performed in nested loops in order to find the optimally specified prediction method. Training is conducted on two thirds of the available data set and testing on the remaining one third. This is done three times using different two thirds for training so that all available data is used.

### 3.2.2 Cross Validation

Performance of the support vector machine classification model was tested using three-way cross validation. Cross validation will, to some extent assure that the prediction performed by the classification model was not just a lucky guess. Step 3 and 4 in Figure 5 illustrates the parts of the program that selects the expression vectors and constructs the training and test sets for the SVMs. The expression vectors of members of the functional class are separated from the non-members. The set containing only member expression vectors is divided into three subsets, the same procedure was applied to the set containing only non-member expression vectors. Each subset of member expression vectors is combined with a subset of non-member expression vectors. The result is three sets of training examples containing both members and non-members of the functional class. Two of the three sets were combined to form training examples, the third set was used for testing the prediction model. This process was repeated three times with a different two thirds, in turn, for training and a different remaining third for testing the model on. Note that the three sets of microarray data are totally disjoint, hence there is no redundancy in the sets of

data, and both training and testing is performed on all data. Steps 5 to 7 in Figure 5 illustrates the three-fold cross-validation process.

### 3.3 The Clustering Algorithm Used for Analysis

In order to be able to compare the performance of the trained SVMs to some other method, a simple algorithm for clustering the expression data in an unsupervised fashion was developed. Unsupervised learning methods are discussed in Section 2.5.1. The measure of similarity was defined as the correlation between a pair of expression vectors. The formula used for calculating correlation (similarity)  $r$  between the expression vectors  $x$  and  $y$  follows here

$$r_{xy} = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_1^n (x_i - \bar{x})^2} \sqrt{\sum_1^n (y_i - \bar{y})^2}}$$

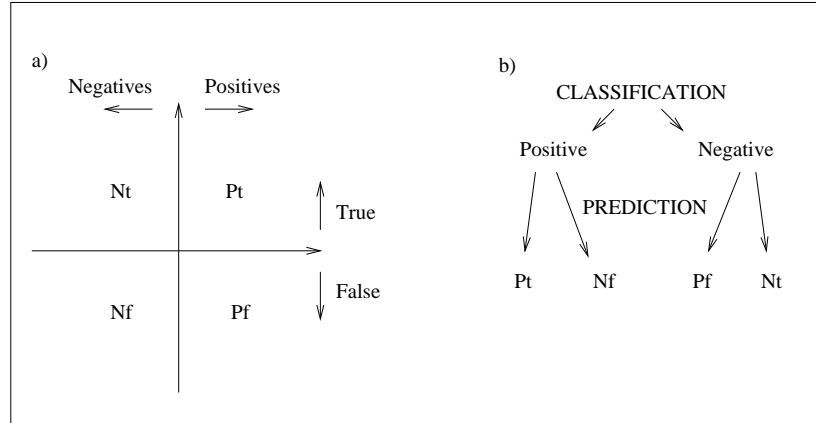
The correlation was calculated between each possible combination of expression vector pair, and expressed as a matrix of correlations. Each possible expression vector pair is the combinations between each vector, member of the functional group, and all the other vectors in the data set. The matrix was then scanned for selecting clusters of high correlation. The scanning of the correlation matrix was performed at different *cutoff* values, which defines the minimum value of correlation. The cluster we were looking for was the cluster containing as many of the genes in the relevant functional group as possible. Performance of the clustering algorithm was evaluated in the same way as was performance of the SVMs.

### 3.4 Performance Measurements

Each training set consists of both members and non-members of the functional class. The functional classifications used are available on the Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) (<http://www.mips.biochem.mpg.de/proj/yeast>). Members of the class are labeled as positive training examples and non-members are labeled as negative examples. A training example for the SVM is the class label and the corresponding expression vector for the particular gene.

As mentioned, the training examples are labeled either as a positive (if member) or a negative (if non-member) training example. Each known test example can be classified as true or false, hence each gene can belong to one of the four following groups; true positives (Pt), false positives (Pf), true negatives (Nt), and false negatives (Nf), see Figure 4 a) and b).

Positive true, or *Pt*, is when the positive (member, according to MYGD) training example is predicted positive (member of the functional class) also by the classifier, and equally the negative true, or *Nt*, is the negative training example predicted as negative (non-member) i.e. MYGD and the classifier agree on functional classification. The two remaining groups are when the MYGD and the classifier disagree of functional classification. The negative false, or *Nf*, is



**Figure 4.** In a) the four possible classifications of a known test gene. In b) a short overview of the classification-prediction procedure, placing the known test genes into one of the classes.

the group that contains the positive training examples predicted by the classifier as negative (non-members), the positive false, or  $Pf$ , which is the group of the negative training examples predicted as positive (members).

Using cross validation, discussed earlier, it is possible to measure how well a certain method performs in the prediction of a test set. To measure the performance of the prediction, on the three-way cross-validated experiment, two different measurements were used: Cost savings for the method used  $S(M)$  and Mathews' correlation coefficient  $C_{Mathews}$ . These two measurements of performance reflects the actual number of correctly and incorrectly classified genes, and give a hint of the performance of the prediction.

### 3.4.1 Cost Savings

By calculating the cost savings, when using a method  $M$ , is possible to compare prediction methods for one specific group. It is not possible to use cost savings for comparing prediction methods between different functional groups. Cost savings  $S(M)$  of using the method  $M$  is defined as

$$S(M) = C(N) - C(M)$$

$C(N)$  is defined as the cost of using no method at all for prediction.

$$C(N) = Pf(N) + 2 * Nf(N)$$

If no method is used for prediction, all test examples will be classified as negative, hence

$$Pf(N) = 0$$

and

$$Nf(N) = Pt(M) + Nf(M)$$

which leads to the relationship

$$C(N) = 2 * [Pt(M) + Nf(M)]$$

Further  $C(M)$  is the cost of using the relevant method  $M$  for prediction.  $C(M)$  is defined as

$$C(M) = Pf(M) + 2 * Nf(M)$$

where  $Pf(M)$  is the number of false positives, and  $Nf(M)$  is the number of false negatives using the method  $M$ . The false negatives are weighted stronger than the false positives, because in this data the fraction of positive examples is considerably smaller than the fraction of negative examples.

Summing all relationships gives the resulting formula for the cost savings

$$S(M) = 2 * [Pt(M) + Nf(M)] - [Pf(M) + 2 * Nf(M)] = 2 * Pt(M) - Pf(M)$$

$S(M)$  could be described as a parameter measuring the performance of a prediction method by evaluating the class of the positively predicted genes.

### 3.4.2 Matthew's Correlation Coefficient

The actual numbers of correctly and incorrectly classified gene expression vectors are calculated and inserted into the formula for Matthew's correlation coefficient,  $C_{Matthews}$ .

$$C_{Matthews} = \frac{(PtNt) - (PfNf)}{\sqrt{(Nt + Nf)(Nt + Pf)(Pt + Nf)(Pt + Pf)}}$$

The value of  $C_{Matthews}$  varies between  $-1$  and  $+1$ , where  $-1$  equals total opposite prediction,  $0$  equals prediction no better than random (untrained SVMs), and  $1$  equals perfect prediction of all tested genes. Matthew's correlation coefficient is an overall measure of the performance of the prediction model, and it is comparable for prediction models for different functional groups as well as prediction models for one functional group. A simple description of the correlation coefficient would be that you subtract the product of the two false groups from the product of the two positive groups, and divide it with a factor that normalizes the difference.

For each method used for prediction, a value of both  $S(M)$  and  $C_{Matthews}$  were calculated. The best performing method i.e. highest  $S(M)$  or  $C_{Matthews}$ , was then applied on the genes of known function in order to evaluate the performance of the classification and then on the genes of unknown function to predict their functional class.

### 3.5 Description of the Program and Experimental Design

One goal of this project was to make an automated method for predicting the functional class of yeast genes, based upon the expression data available. A program that receives in data from user and takes care of all data handling required; such as training the SVMs, selecting the prediction method one would be out looking for, predicts the functional class of a test set, and finally sends the results from the experiment back to the user, was developed. The program was connected to a web page so that anyone interested could access it at any time. The tool for predicting functional class of yeast genes has been used in various kinds of experimental analysis of the expression data for yeast.

#### 3.5.1 Description of the Program

A schematic view of the program, describing the different steps in the procedure of handling the data, is illustrated in Figure 5. Step 1 is when the program receives the user defined input. The user can define the set of microarray expression data that is going to be used in the experimental analysis, both for training and for testing the SVMs, and the user also defines the functional group of genes. When submitting the data in step 2 the program starts the automated prediction. The genes are separated into two sets of expression data in step 3, one set containing the members of the functional group and one set containing the non-members of the functional group. These two sets are further divided into three subsets each (a, b, c, and d, e, f respectively) which are combined to form three sets (a+d, b+e, and c+f) containing both one third of the members and one third of the non-members, step 4.

The first step after separating and combining the data is the training procedure of the support vector machines. Training of the SVMs, step 5 in Figure 5, is conducted in a cross-validation mode, described in Section 3.2.2. In step 6, the SVM prediction model is tested on test data, using the classification module of the SVMs, which is different from the training data. Evaluation of the performance of the prediction model, step 7, is important in finding the desired model for prediction. Performance of a model is measured applying the model on an annotated data set, in the way described in Section 3.4. After having selected the three most optimal prediction models for a specific set of training examples, the model is applied on a different set of test data containing also not previously annotated genes, in order to predict new putative members of the functional group, see step 8 and 9. The three best prediction models were used in order to minimize a weighted result, which the case of only using one model would cause. The data sets were randomized three times, between three consecutive rounds of classifications of the three chosen models, in order to eliminate chance (due to joint order) in the classifications.

A gene constantly classified to a certain group by all three models for classification although the joint order was changed, is classified the same way totally nine times. Hence, the most likely number for a correct classification is nine, whereas the least likely number for classification is zero. The genes classified

three to six times, constitute a group hard to predict what class they belong to.

Using the line of action described here enables the user to find the optimal prediction method, for analyzing a specific combination of functional group and data set. One data analysis experiment gives results about the possible classifications of one functional group, based on the data in one training set and one test set (this data can of course be the same set of data). The user receives the results by e-mail.

### 3.5.2 Experimental Design

The work described in [Brown et al, 2000] was repeated using the automated version of the SVMs. Training and classifying with different sets of options on the same set of data, is it possible to obtain the same performance?

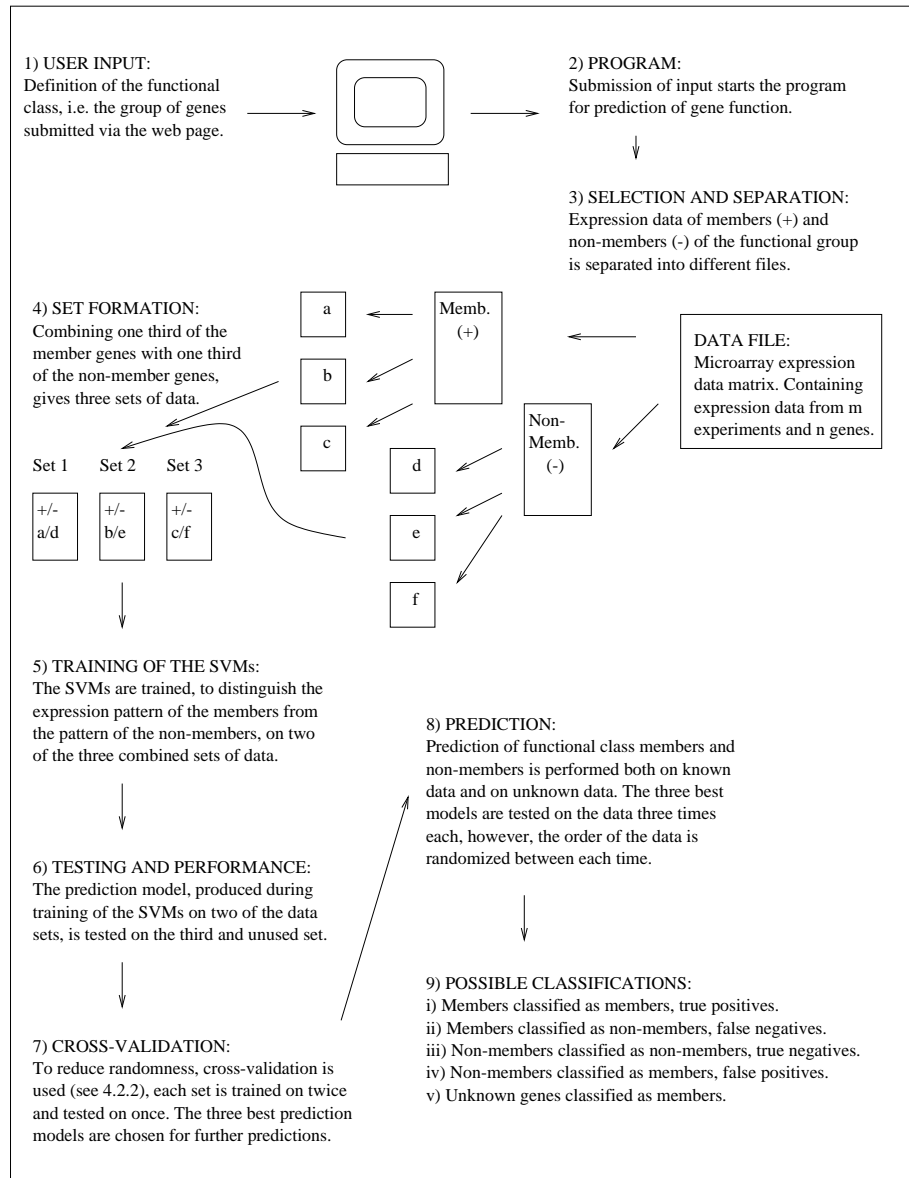
Two different approaches in selecting the members of the functional groups have been used. One, relies simply on the classification made in the Munich Yeast Gene Database (see Section 3.1). The other approach relies on the selection of the member genes as the genes that are present in the highest scoring cluster. The clustering algorithm used, is described in Section 3.3. The clustering algorithm starts off with only one gene, the genes that correlate well to this gene are assigned to belong to that gene's cluster. The whole cluster containing both false and true positives according to the classification made by MYGD, is defined as the group of positive training examples.

The SVMs were trained and applied to different types of yeast data. First, the original data sets obtained from the Stanford web site. Second, the same sets of data but the values of the expression ratios were the absolute values. Third, the original data sets but the expression ratios were squared. The first experiment was conducted to find genes with similar expression patterns, as described before. The two latter experiments were performed with the purpose of testing whether it is possible to find genes that are anti co-regulated to any of the functional groups.

The SVMs were tested in reliability by using only half of the number of members of each functional group. Two things to keep in mind: first, are the SVMs (trained on only the half number of members) able to pick up the other half of the members of the functional group? Second, is the prediction easier or more difficult to do, when the SVMs are trained on only one half, and the other half is labeled as non-members?

Further, the SVMs were trained to recognize other types of functional groups, not previously tested. The aim was to try to identify a "new group" of functionally related genes, which would be possible to make functional predictions on, based on the expression data. The groups tested here were for example: one group of genes was selected by searching in the description of the proteins for the keyword transcription, another was selected using the keyword cell-cycle regulation and a third was a group of genes found using the keyword peroxisome.

A main goal spanning over all the experimental approaches described above, was to try to identify new ways of finding functionally related groups of genes.



**Figure 5.** A schematic description of how the program works, starting with the user input from the web page resulting in the classification of a gene. More details in Section 3.5.1.

## 4 Results and Discussion

Until recently, most computational methods for analyzing genome-wide expression data have focused on unsupervised clustering algorithms. The aim of this project was to further test and evaluate the advantages as well as the disadvantages of using SVMs as a knowledge-based method for pattern recognition in expression data, and develop a tool accessible for the interested scientist. Results that will be presented and discussed in this section are: first, the user interface web page, which is connected to the automated program for predicting functional class, and second, the results obtained in the different experimental approaches.

### 4.1 The User Interface

The user interface, to be used for the tool for prediction of the functional class of genes based on the expression data, needs to be an interactive structure. The type of solution presented here is accessible to everyone interested. The user enters input information, needed in the automated program for function prediction of genes, onto the dynamic web page shown in Figure 6, at <http://www.sbc.su.se/~annette/genesearch.cgi>. This user interface is designed to facilitate the process of predicting the function of genes using the SVMs for knowledge-based analysis of microarray data. This tool has been developed using yeast data but is also applicable on microarray expression data from other organisms like mouse, rat and human.

The main steps in using the tool for predicting gene function are to be reviewed in brief here. At the top of the page, the user is presented with a short guide “How do I do?”, to make a prediction of gene function using this interface. The first thing the user needs to do, is to make sure that the genes to be analyzed are of the right organism. The next thing to do is to define the functional group, this is done simply by entering the *gene codes* (one unique code for each existing gene), of each member of the group, to the text area “Definition of functional group members”. On the left hand side of the text area, there are http-links to web pages with lists of functional groups of genes. The lists can be copied and then pasted into the text area on the main page. The functional groups listed, are the TCA, respiratory, cytoplasmic ribosomal, proteasome and histonal, according to the MYGD classification. Each of the listed genes are also further linked to their own pages of information, found in the MYGD, at [http://vms.mips.biochem.mpg.de/htbin/search\\_code/](http://vms.mips.biochem.mpg.de/htbin/search_code/). If the user wants to define the functional group him- or herself, it is perfectly appropriate to do so. One thing to remember is that, the less co-regulated the functional group members are the longer the training procedure will last, since the expression pattern is not very obvious. When the functional group is defined, the name of the user, and/or a label of the experiment, is to be entered in the text box “Your name”. The last thing before submitting the input is to enter the e-mail address to which the user wishes to receive the results. To submit, press the button “Run the SVM program”. After submission of the input, it might take

some time to check the input data, in order to either confirm the starting of the program, or generate an error message. If incorrect input, such as incorrect gene codes in the text area for defining functional group members, there will be an error message pointing out the incorrectly entered genes. The errors can be corrected by pressing the back button in the web browser, changing what was wrong and the submit again. The program will not start until all errors are corrected. If the input is correct, on the other hand, the program will start and a web page confirming succeeded submission will appear. The user will receive the results from the SVM prediction of gene function via e-mail as soon as the program is ready. The results are sent to the user by e-mail, in which there is a link to a web page containing more detailed list of the results.

Further down the main page but not shown in Figure 6, are relevant links to other databases such as MIPS, PubMed, BLAST, Entrez etc. These links can be useful when searching for information about specific genes, evaluating the functional groups and the results from the prediction.

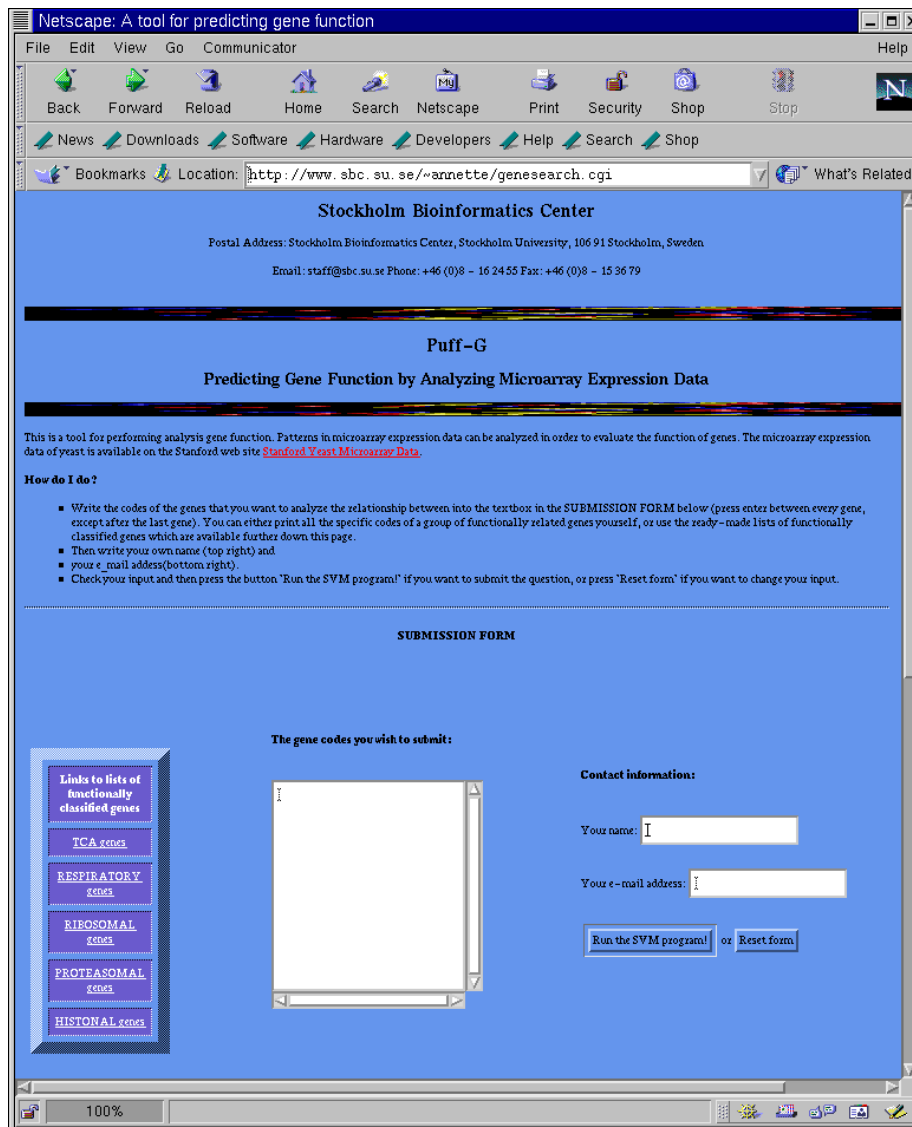
The results sent by e-mail to the user contains information about the prediction models and the classifications made. The three best prediction models are presented, these are defined by information about the kernels and the options used. The performance, as in the values of  $S(M)$  and  $C_{Matthews}$ , and the number of genes classified as Nt, Pf, Nf, or Pt, are presented for each prediction model. The classification data is the lists of classified genes produced, by the prediction models applied on the data with randomized order. The lists that are presented are those containing: a) the members of the functional group (according to MYGD) classified as non-members by the SVMs (Nf), b) the non-members (according to MYGD) classified as members by the SVMs (Pf), and finally c) the previously unannotated genes (MYGD) that are predicted as members by the SVMs.

## 4.2 Results of Functional Classifications

### 4.2.1 Results from Supervised Learning

The results of the experiments show that some of the functional classes can be recognized using SVMs trained on expression data. It was suggested by [Brown et al, 2000] that SVMs provide superior performance in predicting functional class compared to other methods. For all classes, except for the helix-turn-helix class, it was the SVMs using the polynomial or radial basis kernel that made the best predictions in these experiments. The class of proteins containing helix-turn-helix motifs is not expected on biological grounds to be functionally related, hence used as control group.

The performances of the SVMs are presented in Table 2. The values of  $S(M)$  and  $C_{Matthews}$  obtained in these experiments are listed in column 7 and 8, these values are the results obtained using the three best prediction models (column 2) for each functional class (column 1), regardless of the kernel used. The values of  $S(M)^*$  presented in the article, are listed (column 9) to compare with, these results are the three best obtained using different kernels. Note, that the



**Figure 6.** This is the main web page of the user interface, of the tool for prediction of gene function.

value of  $S(M)$  can only be used for comparing the prediction models in a class, whereas the  $C_{Matthews}$  is possible to compare between both prediction models in a class, and prediction models between different classes. The number of genes classified into one of the four categories (column 3 to 6): true negatives (Nt), false positives (Pf), false negatives (Nf), and true positives (Pt), are presented for each model. The Nt, Pf, Nf and Pt values are all summarized over a three-fold cross-validation experiment.

A high score of cost savings ( $S(M)$ ), and a score close to +1 for Matthew's correlation coefficient ( $C_{Matthews}$ ), indicates a good model for prediction. The easier a group of genes are to predict, the higher value of  $C_{Matthews}$  is obtained. The results presented here are in some cases different from the results presented by [Brown et al, 2000], but relatively equal overall. Their method for predicting the genes belonging to the TCA group is slightly better, but the situation is the contrary when it comes to prediction of the ribosomal and proteasome genes.

#### 4.2.2 Results from Unsupervised Learning

Another approach to predict gene function was to cluster the genes using the clustering algorithm described in Section 3.3. Each cluster was obtained by scanning the columns of correlation values, in the correlation matrix, at a certain cutoff value of the correlation. Each gene, member of the chosen functional group, obtains an individual cluster containing genes that have correlation values higher than the cutoff value. The genes in each functional class that achieve the three best clusters, according to the performance measurements, are presented in Table 3. The functional class is presented in column 1, and the genes representing the three best clusters are listed in column 2. The cutoff values are listed in column 3, the four possible classifications (Nt, Pf, Nf, and Pt) in column 4 to 7, and the performance measurements are listed in column 8 and 9.

As we can see, when comparing the two tables for performance of the trained SVMs (Table 2), and the performance of the clustering method (Table 3), the trained SVMs perform better than the clustering method, and do so for every class tested, except for the histone genes that are predicted equally in both cases. The helix-turn-helix proteins are excluded since they are not expected to be correlated.

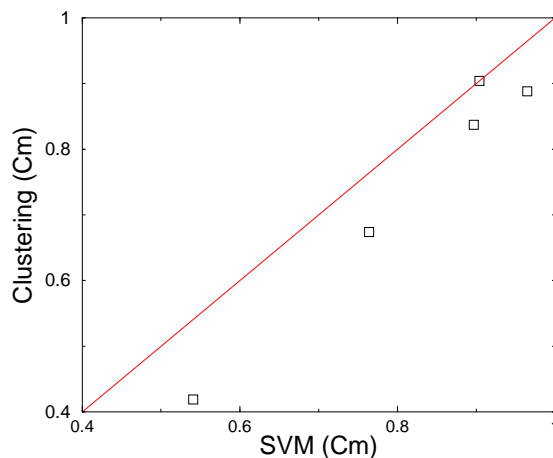
By plotting the performances see Figure 7, measured in  $C_{Matthews}$ , of the supervised prediction method (SVMs) on the x-axis against the unsupervised prediction method (clustering) on the y-axis, it is possible to see a pattern. All plotted points, except for the performance for predicting histone genes, lie below the drawn line. This means that the performance of the prediction models are generally better when using the supervised method based on the SVMs, since the line illustrates equal performance.

Results - SVM prediction								
Class	Model	Nt	Pf	Nf	Pt	$S(M)$	$C_{Matthews}$	$S(M)^*$
TCA	1	2450	0	12	5	10	0.541	12
	2	2446	4	10	7	10	0.509	11
	3	2447	3	11	6	9	0.483	9
RESP	1	2430	7	7	23	39	0.764	39
	2	2433	4	9	21	38	0.764	38
	3	2431	6	9	21	36	0.735	33
RIBOS	1	2342	4	4	117	230	0.965	229
	2	2342	4	4	117	230	0.965	229
	3	2338	8	3	118	228	0.953	226
PROTE	1	2429	3	4	31	59	0.897	52
	2	2431	1	6	29	58	0.894	52
	3	2430	2	6	29	56	0.879	48
HIST	1	2456	0	2	9	18	0.904	18
	2	2456	0	2	9	18	0.904	18
	3	2456	0	2	9	18	0.904	18
HTH	1	2451	0	16	0	0	0.000	0
	2	2451	0	16	0	0	0.000	-1
	3	2450	1	16	0	-1	-0.002	-3

**Table 2.** The performance of the SVMs in predicting the six functional classes: tricarboxylic acid cycle (TCA), respiratory, cytoplasmic ribosomal, proteasome, histonal and helix-turn-helix proteins (first column), using the original expression data set. The three best performing prediction models over all (column 2). Column 3 to 6 are: true negatives, false positives, false negatives and true positives, summed over three splits in cross-validation. Column 7 represents cost savings,  $S(M)$ , and column 8 represents the value of Matthew's correlation coefficient,  $C_{Matthews}$ . The three best values of  $S(M)$ , presented in the article [Brown et al, 2000], are listed (column 9) for comparison ( $S(M)^*$ ), these values are obtained using different kernels.

Results - Clustering								
Class	Cluster	Cutoff	Nt	Pf	Nf	Pt	$S(M)$	$C_{Matthews}$
TCA	YOR136W	0.75	2450	0	14	3	6	0.419
	YNL037C	0.75	2450	0	14	3	6	0.419
	YPL262W	0.75	2449	1	14	3	5	0.362
RESP	YDR529C	0.6	2428	9	10	20	31	0.674
	YHR051W	0.6	2422	15	8	22	29	0.656
	YGL187C	0.6	2422	15	10	20	25	0.612
RIBOS	YJL136C	0.7	2320	26	3	118	210	0.888
	YDR500C	0.7	2327	19	7	114	209	0.893
	YJL190C	0.7	2320	26	4	117	208	0.883
PROTE	YFR050C	0.6	2424	8	4	31	54	0.837
	YJL001W	0.6	2428	4	7	28	52	0.834
	YDR427W	0.6	2426	6	6	29	52	0.826
HIST	YBR010W	0.65	2456	0	2	9	18	0.904
	YNL031C	0.65	2456	0	2	9	18	0.904
	YNL030W	0.65	2456	0	2	9	18	0.904

**Table 3.** Performance results of clustering. The functional class is presented in column 1, the three genes obtaining the best clusters in column 2. The cutoff values used to obtain the optimal clusters, are presented in column 3. The number of genes classified as Nt, Pf, Nf, or Pt are listed in column 4 to 7. The values of  $S(M)$  and  $C_{Matthews}$  are listed in the two last columns.



**Figure 7.** A plot of the performances, measured in  $C_{Matthews}$ , of the two prediction methods. Supervised machine learning on the x-axis and unsupervised on the y-axis.

Results - Different Definitions of Functional Group							
Class	Method	Nt	Pf	Nf	Pt	$S(M)$	$C_{Matthews}$
i) TCA	SVMs on MYGD	2450	0	12	5	10	0.541
ii) TCA	SVMs on Cluster	2465	0	1	1	2	0.707
iii) RIBOS	SVMs on MYGD	2342	4	4	117	230	0.965
iv) RIBOS	SVMs on Cluster	2316	7	4	140	273	0.960

**Table 4.** Different methods are used for defining the functional classes. In row i) and iii), the definition of the class is the MYGD one, whereas in row ii) and iv), the definition is made by clustering the data. The functional class and the method used (column 1 and 2), the number of genes predicted as Nt, Pf, Nf, and Pt (column 3 to 6), and finally the values of the performances  $S(M)$  and  $C_{Matthews}$  (column 7 and 8).

### 4.2.3 Effects of Defining the Functional Group Differently

The cluster can be used as a definition of the group of positive training examples, this approach is an attempt to try to investigate whether the SVMs are able to better recognize and pick up genes expressed in a similar pattern, if the group of training examples is filtered or “tuned” to contain only the genes with the most characteristic expression pattern. The results obtained when training the SVMs on the vectors in the clusters, show that the clusters give a slightly different view of the class definition than do MYGD. For example, the ribosomal cluster contains most of the ribosomal genes but also some non-members according to MYGD, this more widely defined class is relatively easy to predict. Another example is the TCA cluster, which contains virtually no members at all and is almost impossible to predict. Table 4 displays the results for the SVM predictions of the TCA and the ribosomal genes, using both the MYGD functional classifications (see rows i) and iii)), and the definition made by the clustering algorithm (see rows ii) and vi)).

One overall feature is that it is easier for the SVMs to predict the class members if the group is defined by clustering rather than the MYGD definition, which is relatively obvious since the genes in the cluster are highly correlated. For example, compare the  $S(M)$  in row iii) to the  $S(M)$  in row iv), here you can see a higher value of  $S(M)$  if the clustering method defines the functional group. The obvious drawback is of course that the definition obtained by clustering may be more correlated, but it is not necessarily the true or complete one.

This experiment is interesting in the aspect that it is possible to start from only one gene and obtain a quite reasonable cluster representing a functional class, which may be further analyzed in the SVMs. Another valuable aspect is that, when training on the clusters, it is possible in some cases to pick up genes that were excluded from the cluster but included in the MYGD definition of the functional group. There are some examples of the genes excluded from the clusters that are found to belong to the functional group by the SVMs trained on the cluster. The examples are found when training on the clusters

<b>Results - Half Functional Group</b>				
Class	Tot.	Pos.	Neg.	Found
TCA	17	9	8	1
RESP	30	15	15	10
RIBOS	121	61	60	47
PROTE	35	18	17	10
HIST	11	6	5	2

**Table 5.** The SVMs are trained to recognize half of the functional group (MYGD classification), which is labeled positively. The other half is labeled negatively during training, but still the SVMs are capable of recognizing some of the negatively labeled as members of the functional group. Class is defined in column 1, the total number of positives in functional group in column 2, the number of positively labeled in column 3, and the number of negatively labeled (the remaining ones) are listed in column 4. Finally, the number of found i.e. the negatively labeled present in the total number that are recognized by the SVMs.

that are from the classes that are easier to predict. For example, the respiratory genes (MYGD) YKL016C and YPL078C were excluded from the clusters, whereas the SVMs trained on the clusters recognized them as members of the functional class. A similar example is illustrated by the cytoplasmic ribosomal genes (MYGD) YKL180W and YBR048W, which are excluded from the cluster but recognized by the SVMs.

It was also investigated whether the SVMs are able to perform a truthful prediction of class members given only half the group of positive genes according to MYGD. This approach shows that the SVMs are capable of picking up quite a few of the remaining half, which is interesting since these genes were labeled as negative in the training data set. This case is possible, since the SVMs have the ability to construct a classifier with a “soft margin”, which means that it allows for some examples (outliers) to be on the wrong side of the hyperplane. These outliers, when not excluded from the positive training data set, will contribute to a more correct picture of the positive group, hence promote the construction of a more true prediction model. Table 5 shows the functional class (column 1), the total number of genes in the MYGD class (column 2), the number of genes trained on, and the number of MYGD genes labeled negative (column 3 and 4). The number of genes of the negatively labeled that the SVMs are able to pick up, are listed in column 5.

These results allow us to believe that the classifications made by the SVMs are quite true, since the SVMs prove that they manage to find members although the training information is not entirely correct. The prediction becomes more truthful and correct if we have many examples of members of the functional group to train the SVMs on.

Results - Using Absolute Values of Data						
Prediction	Nt	Pf	Nf	Pt	$S(M)$	$C_{Matthews}$
TCA	2449	1	12	5	9	0.493
RESP	2429	8	18	12	16	0.485
RIBOS	2339	7	6	115	223	0.994
PROTE	2423	9	8	27	45	0.757
HIST	2456	0	3	8	16	0.852

**Table 6.** This table shows the results of the predictions using SVMs trained on the absolute values of the expression ratios. The values of  $C_{Matthews}$  are listed for all groups, and so are also the number of genes in each class predicted as Nt, Pf, Nf, or Pt.

#### 4.2.4 Anti Co-Regulated Genes

In order to enable a cell to exhibit a specific function there are some genes that must be expressed, hence co-regulated. Analogous to this, a reasonable thought should be that some specific genes must be silenced (or anti co-regulated) for a cell to be able to carry out a certain function. Plausible ways to discover anti co-regulated genes could be to, either use the absolute values, or the squared values of the expression ratios in the original data sets. Using the absolute or the squared values of the expression ratios means, of course, that no negative values are present and that the expression profiles are somewhat modified.

Using the absolute values of the data set as in-data to the SVMs results in less specific expression patterns for all genes, which in turn results in a prediction not as good as the one using the original data set. The results are shown in Table 6.

It seems like the prediction model becomes less specific when trained on this data. It is harder to find the true positives (the number of false negatives increases), on the other hand, it is less likely for a gene to be predicted as false positive (the number of true negatives increase). In conclusion this means that the expression vectors seem to become more separated, i.e. if an expression vector most certainly belongs to the positive group it will be even more obvious when using the SVMs on the modified data sets.

The approach described here does not specifically find the genes that are entirely anti co-regulated, it may also guide us to the genes that are partly anti co-regulated. When applying this way of thinking to the data set, no obvious pattern in the data is visualized. When looking for genes that are anti co-regulated, these might appear as the genes that are present in the class of false positives using the data sets of absolute or squared values, but not present in the class of false positives using the original data set. Examples of genes that fulfill these requirements are: a) anti TCA e.g. YFL014W, b) anti respiratory e.g. YBL023C, c) anti ribosomal e.g. YPL259C, and d) anti proteasomal e.g. YJR117W (no anti histonal found). These results are quite vague and should be related to both the performance of the prediction model, and to the number

of times a specific gene has been predicted to belong to a certain group.

#### 4.2.5 Other Functional Classes Tested

The SVMs were trained to recognize a few additional functional classes, apart from the six functional classes evaluated in the experiments above. Keywords were used to search the descriptions of the yeast genes. Genes that matched the keywords were gathered to a functional group, which was used for training the SVMs on. Examples of such “new” functional groups that were established using keyword search, are peroxisomal genes (15 genes), and cell cycle (91 genes) [Spellman et al, 1998]. The peroxisomal were not recognizable at all ( $S(M)=0$ ), whereas the pattern of the genes in the cell cycle group was learn-able to some extent, but not very convincing ( $S(M)=6$ ). A good idea, which would help in the search of a co-regulated group of genes, could be to start with a clustering algorithm. Next, investigate the clusters in order to find a cluster likely to be a functional group, and finally use the SVMs for further classifications.

Some functional classes were assembled for human gene expression data. The classes were obtained using a keyword search in the gene annotations. Some experiments were conducted on the human data, but without any results. The simple reason is that the groups were incorrectly defined, a clustering experiment could have helped, which unfortunately was outside the time span of this work.

## 5 Conclusions and Future Improvements

The quite recently developed microarray technology now allows us to conduct genome-scale characterizations of gene expression in one single experiment, and it can be employed in various fields of research. The most prominent areas, are in studies of polymorphisms, screening for mutations, and in investigating transcriptional control factors. Further, microarrays can be used in the screening for disease genes and potential drug targets, by evaluating effective treatments. Evolutionary relationships can be revealed, tissues characterized, and cancer expression patterns can be profiled, these examples are other areas where microarrays can be used. One of the main goals of research in molecular genetics today, is to develop and improve simple and effective means of computational methods for functional analysis of genes. As information and annotations of genes and proteins increase at considerable rate in various databases, it would be favorable if this information could be used for further research.

The expression patterns that are obtained on the microarray, are highly dependent on the type of experiments used to profile the genes with. The experiments described here are all somehow related to the changes occurring during a cell cycle or in response to external factors. There are many functional groups that are almost impossible to recognize in this way. For example, a group of proteins that regulate the cell cycle, such as the cyclins, is not possible to recognize due to the complex and periodic pattern. All cyclins are not expressed at one point in time during the cell cycle, rather, these genes are expressed one

after the other in a precise order. There are some vital questions to answer, which genes should theoretically be co-expressed, and how do you design the experiments in order to find them?

There are a few experimental difficulties associated with using DNA microarray expression data for predicting gene function. The public data obtained from the Stanford web site or from any other microarray expression experiment is not expected to be perfect and show the true and actual expression levels of genes. The fact that the quality and the reproducibility of array experiments is still improving, indicates that the obtained data is not the perfect image of reality. Methods built to extract information from arrays must be engineered to handle extreme amounts of noise, since the resolution of the data tends to be coarse.

The assumption motivating a search for co-expressed genes is that simultaneously expressed genes often share a common function. There are, however, incorrect conclusions that might be drawn based upon this assumption, which is the reason why expression pattern analysis alone cannot address this problem. Theoretically, functionally related genes can show completely opposite expression patterns (or anti correlation). There are many examples in nature of genes that must be suppressed in order to enable the expression of other genes. Another aspect of expression regulation is that genes that are simultaneously expressed do not necessarily share a common function. There is also the possibility that genes expressed at separate times may complement each other in performing one function. The genes that cluster together (are simultaneously expressed), do not tell you anything about the function unless you perform additional background research. Finally, as we all know, nature is ambiguous which is pointed out by one gene possessing more than one function.

In order to improve the prediction methods for determining the functions of proteins, all sorts of available information should be included in the model for classification. One idea could perhaps be to connect the gene expression profile search to other existing databases, containing everything from structural knowledge and cellular localization, to annotated experimental knowledge [Shatkay et al, 2000]. It is thought that some genes sharing similar function lie close to each other in the genome, hence are transcribed at the same time. It would not be difficult to add information about chromosomal localization to improve the method for function prediction. As shown in this work by testing different approaches, such as training the SVMs on the original data, clustered data, parts of the data, and the absolute values of the data, all additional information will help in the development of enhanced prediction models. Almost every little experiment tells the researchers something new. The trained SVMs are able to perform quite trustworthy predictions assuming that the quality of the indata is reasonable. This means that it is a rather robust tool that, if used correctly, which can help researchers to attach functions to genes.

The web based tool for predicting gene function needs to be refined at several points. One idea is to enable an entry where it would be possible to view the expression vectors graphically, another idea is to connect the clustering algorithm to the tool so that it would simplify the assembly of a putative functional group.

When enough microarray expression data accumulate, it is time to link the tool to a database with expression data. The data sets should be simple to handle so that only specific parts of the data sets could be used, if so desired. The database should contain data from several organisms, which would for example enable search of homologue genes.

Needless to say, there are several combinations of clustering algorithms and supervised learning methods not evaluated here, which of course would contribute to a more holistic picture of the situation and guide us in to the future.

## 6 Acknowledgements

There are several people that has meant a lot to me during my Master's Thesis project. In particular, I would like to thank my supervisor Arne Elofsson at the Stockholm Bioinformatics Center (SBC), for his encouraging support and valuable ideas throughout the project. I also want to thank my examiner at the Linköping Institute of Technology/Faculty of Health Sciences Peter Söderkvist, for continuous discussions and an helping hand with the practical details surrounding the project. Many thanks to my opponent, Hans Green, for proofreading and contributing with relevant comments.

I would like to thank the people at the SBC for stimulating and clarifying discussions, technical assistance, proofreading, and precious knowledge. The unique blend of people at the SBC; senior scientists, PhD students, and master students, with varying backgrounds create an inspiring and great atmosphere to work in.

Finally, I would like to thank my boyfriend Pierre, my family, and friends for their doubtless support and for them spurring me to fresh efforts.

## References

- [Alizadeh et al, 2000] A. Alizadeh, M. Eisen, R. Davies, C. Ma, I. Lossons and A. Rosenwald. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 3;403(6769):503-11, Feb 2000.
- [Ashburner et al, 2000] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, and J. Eppig et al: Gene ontology: tool for the unification of biology. *Nat Genet*, 25:25-29, 2000.
- [Bairoch, 2000] A. Bairoch: The ENZYME database in 2000. *Nucleic Acids Res*, 28:304-305, 2000.
- [Bittner et al, 2000] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang and E. Seftor. Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling. *Nature* vol. 406: 546-540, Aug 2000.

- [Brown et al, 2000] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS* Vol. 97 no. 1, 262-267, Jan 2000.
- [Burges, 1998] C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
- [Chu et al, 1998] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P. Brown and I. Herskowitz. The Transcriptional Program of Sporulation in Budding Yeast. *Science*, 282, 699-705, 1998.
- [Cristianini et al, 2000] N. Cristianini and J. Shawe-Taylor. An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000.
- [DeRisi et al, 1997] J. DeRisi, V. Iyer and P. Brown. Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. *Science*, 278, 680-686, 1997.
- [Eisen et al, 1998] M. Eisen, P. Spellman, P. Brown and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS* 95:14863-14868, 1998.
- [Gerold et al, 1999] D. Gerhold, T. Rushmore and T. Caskey. DNA chips: promising toys have become powerful tools. *TIBS* 24, 168-173, May 1999.
- [Gerstein et al, 2000] M. Gerstein and R. Jansen. The current excitement in bioinformatics - analysis of whole-genome expression data: how does it relate to protein structure and function? *Current Opinion in Structural Biology*, 10:574-584, 2000.
- [Joachims, 1999] T. Joachims, 11 in: *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning.*, Edited by B. Schölkopf, C. Burges and A. Smola, MIT Press, 1999.
- [Mewes et al, 2000] H. Mewes, D. Frishman, C. Gruber, B. Geier, D. Haase, A. Kaps, K. Lemcke, G. Mannhaupt, F. Pfeiffer, and C. Schuller et al.: MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*, 28:37-40, 2000.
- [Ogata et al, 1999] H. Ogata, S. Goto, K. Sato, W. Fujibushi, H. Bono, M. Kanehisa: KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*, 27:29-34, 1999.

- [Sharan et al, 2000] R. Sharan and R. Shamir. CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. Proceedings, Eight International Conference on Intelligent Systems for Molecular Biology, ISMB. La Jolla, California, Aug 16-23, 2000.
- [Shatkay et al, 2000] H. Shatkay, S. Edwards, W. Wilbur and M. Boguski. Genes, Themes and Microarrays Using Information Retrieval for Large-Scale Gene Analysis. Proceedings, Eight International Conference on Intelligent Systems for Molecular Biology, ISMB. La Jolla, California, Aug 16-23, 2000.
- [Spellman et al, 1998] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein and B. Futcher. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell*, Vol. 9, 3273-3297, Dec 1998.
- [Stitson et al, 1996] M. Stitson, J. Weston, A. Gammerman, V. Vovk and V. Vapnik. Theory of Support Vector Machines. Technical Report CSD-TR-96-17, Royal Holloway, University of London, Dec 1996.
- [Vapnik, 1995] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer 1995.
- [Velculescu et al, 1995] V. Velculescu, L. Zhang, B. Vogelstein, and K. Kinzler. Serial Analysis of Gene Expression. *Science*, 270:484-487, 1995.
- [Venter, 2000] C. Venter. Personal communication, Nov. 2000.