UPTEC X07 046
Examensarbete 20 p
September 2007

Implementation of an automated model selection and validation algorithm and its application to breast tumour and allergen classification

Daniel Edsgärd

UPPSALA UNIVERSITET

Molecular Biotechnology Programme

Uppsala University School of Engineering

UPTEC X 07 046	**Date of issue 2007-09**

Author

Daniel Edsgärd

Title (English)

Implementation of an automated model selection and validation algorithm and its application to breast tumour and allergen classification

Title (Swedish)

Abstract

This work involves implementation of an automated model selection procedure and implementation of a validation technique giving an unbiased performance estimate of the selected models. The implemented model selection and validation algorithm is applied to a breast tumour and an allergen classification problem. From these studies it can be concluded that proper validation of selected models is critical as to not report biased performance estimates. Second, for both of the applications the composition of the retrieved feature subsets is not stable. However, high performing feature subsets are generally extracted by applying a genetic algorithm implemented in this work. The model selection of breast tumour classifiers indicates that predictors performing better than van't Veer's 70-gene predictor[48] should be possible to construct. Additionally, TSPYL5 is identified as a putative oncogene. Concerning the allergen classification it is concluded that high performing predictors can be constructed using a considerably smaller subset of peptide fragments (FLAPs) than previously reported. However, reliable epitopes may not be directly extracted due to unstable feature lists.

Keywords

statistical learning, cancer classification, microarray, gene expression, allergenicity, epitopes

Supervisors

Mats Gustafsson

Department of Medical Sciences and Department of Engineering Sciences, Uppsala University

Scientific reviewer

Anders Karlén

Department of Medicinal Chemistry, Uppsala University

Project name	Sponsors
Language **English**	Security 2008-09
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **68**

Biology Education Centre Biomedical Center Husargatan 3 Uppsala

Box 592 S-75124 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 555217

Implementation of an automated model selection and validation algorithm and its application to breast tumour and allergen classification

Daniel Edsgärd

Sammanfattning

Detta examensarbete behandlar frågor om hur man kan lära en dator att skapa en modell som kan förutsäga ett visst tillstånd eller en viss egenskap hos ett objekt eller system. Arbetet inleddes med att skapa ett program som automatiskt kan välja ut den modell som verkar göra mest precisa förutsägelser. Programmet utvärderar också automatiskt den utvalda modellen för att försäkra sig om att dess prediktioner är pålitliga.

Detta program tillämpades därefter på två intressanta medicinska frågeställningar. Det första problemet gäller frågan om en bröstcancertumör kommer att bilda metastaser eller ej. Under senare år har så kallade mikroarray experiment kommit att framstå som mycket lovande komplement till mer traditionella kliniska markörer. Genom att tillämpa det i denna studie utvecklade programmet på sådan typ av data återfanns en gen som till stor del bidrar till att kunna särskilja om en bröstcancertumör blir metastasbildande eller ej. Denna gen kan således troligen hjälpa ställandet av diagnos, bidra till förståelsen av bröstcancer, samt utgöra ett potentiellt mål för läkemedel.

Den andra behandlade frågeställningen gäller bedömandet om det finns en risk att ett givet protein framkallar en allergisk reaktion eller ej. Nya proteiner introduceras idag i grödor och andra produkter genom genetiskt modifierade organismer (GMO), varför det finns ett uppenbart behov av att riskbedöma proteiners allergenicitet innan introduktion i näringskedjan. Studien syftade även att finna generella egenskaper som uppbär den allergena funktionen hos proteiner.

Examensarbete 20 p i Molekylär Bioteknikprogrammet
Uppsala universitet september 2007

Contents

1 Introduction

1.1 Data Analysis in Biology and Biomedicine

There is an ever increasing amount of unexplained medical and biological data. The progress of technical development has resulted in highly automated experimental instruments, such as sequencing machines, microarray platforms and robotic systems for performing highly repetitive laboratory tasks. Examples of contemporary projects producing massive amounts of data are metagenomics projects such as that initiated by Craig Venter in the Sargasso Sea[11], or the Human Proteome Atlas project headed by Mattias Uhlén[36], with the aim of systematically map the human proteome to tissue samples by the use of antibodies. This is currently Sweden's largest research project[31].

In the view of this development towards large scale production of data and in the post-genome era of today it should be well established that there is a need for powerful *in silico* methodologies that can extract knowledge from data, such as algorithms for the discovery of gene function.

Seemingly, the advancement of data analysis methods is lagging behind the experimental high throughput methods; there are still much information left to extract from the enormous and fast growing amounts of data. To recite a pertinent quote: *"We are drowning in information and starving for knowledge"*, Robert D. Rutherford[1]. In other words, the possibilities for computational data analysis to reveal important knowledge, today hidden in the mountains of data, is breathtaking.

Additionally, following the increase in availability of computational power, the power of data analysis ought to never have been larger by being supplied the basic means for meeting the demand of large dataset analysis.

These two facts, the demand of data analysis and the increase in computational power, have in fact also spurred the vitality of the field of data analysis. Concepts such as data mining appears to be spreading its roots into more and more fields. For example, a search in the article database PubMed for "machine learning" today returns around 25000 hits, a number which would certainly be considerably lower a few years ago. It is also worth noticing that in the fields of computational biology the problems under study are often of extremely high dimensionality. The nature of the practical problems within computational biology in combination with the advent of computational power will most certainly push certain concepts within statistics to their very edge and thereby spur the development of statistics.

In response to the large scale data generation in biomedicine, computational biology has experienced a considerable growth in recent years and gained acceptance as a necessary tool to accompany experimental biology. Most successful biomedical research centers of today seem to have identified the need of a close collaboration between wet-lab scientists and in-silico analysts. As an example of this duality one may turn to the field of industrial drug development, wherein QSAR[1] modeling today takes a great role in the search of new high-efficacy compounds in the chemical space.

[1] Quantitative Structure Activity Relationship

1.2 Data Analysis in Practice

Despite the aforementioned call for bioinformatics tools and more advanced data analysis applications in particular, it is hard to find completely satisfactory software.

First, applications most widely in use by experimental biologists today are often not advanced enough, commonly only offering univariate statistical analysis. A more sophisticated treatment of data may enable extraction of information and relationships otherwise gone unnoticed. Computational biologists and their tools do here play an important role. For example, a software tool that offers a multivariate perspective may change the way that many experimental scientists view and interpret their data and results.

Second, if one search to pursue a more advanced analysis it often involves a processing-flow of several steps, and commonly the subprocesses are found not to be implemented in a single integrated environment but instead one needs to resort to using a multitude of applications.

Third, one would like to have the possibility to make changes in the algorithms or implement completely novel algorithms. This may be especially important in a research environment context, since the analysis often goes beyond routine tasks.

On the wish list one would therefore place a data analysis software that implements most of the commonly used algorithms in a single interface, and where the source code is open.

Two software tools that gathers the majority of users for higher level statistical analysis are Matlab and R. For the purpose of statistical pattern recognition one of the most complete packages found was Pattern Recognition Tools (PRTools)[41], which is a third-party Matlab toolbox developed at Delft University, The Netherlands, in which most of the code is open to read and edit.

The work in this thesis is mainly based on the PRTools toolbox; functionality of interest has been added and it has been the major working engine behind the classification problems targeted in this thesis.

The toolbox and the work of this thesis treat problems within pattern recognition. Therefore, before moving on to the more detailed specifications of the treated problems, a brief overview of the basics of pattern recognition is given. This outline will provide definitions of the most fundamental terms and some understanding of the problem domains wherein the pattern recognition paradigm is applicable.

1.3 Pattern Recognition Essentials

From a computer science viewpoint, pattern recognition is seen as a subfield of artificial intelligence, concerned with techniques that allow a computer to "learn". The major goal is to extract information from data automatically by computational and statistical methods. A pattern recognition system can be learned to extract patterns from data or to group data that are similar in some sense. It may also be trained to form rules, or construct a map, which is used to predict properties of a data sample. From a mathematical standpoint, pattern recognition may be viewed as the applied field of statistical analysis of large datasets.

Applications of pattern recognition are found in a wide range of areas, such

as, *natural language processing*, *speech and handwriting recognition*, *search engines*, *medical diagnosis*, *bioinformatics* and *chemoinformatics*, *stock market analysis*, *image analysis*, *object recognition in computer vision*, *game playing* and *robot locomotion*. As might be understood from the applications, the data being handled can either be static or temporal in form of time series.

Supervised and Unsupervised learning
On the highest level pattern recognition can be divided into two subtopics, exploratory methods and predictive methods. Exploratory methods deals with finding structures, or groupings, within data. Typical examples are clustering and self organizing maps. The aim of predictive learning systems on the other hand is to predict the value of some property of a given object. The training of systems to gain predictive power may in turn itself be subdivided into two types, supervised learning and unsupervised learning. The distinction between these two learning strategies can clearly be described after having introduced some basic terminology.

Features
A data sample, that may represent a physical object, is described by a set of measured properties. Thus, a vector that contains the values of these properties, $X = [x_1, x_2, ..., x_N]$, termed pattern, describes an object. The properties that the elements in the vector represents are termed features. For example, a fruit may be described by the two features weight and colour. In bioinformatics and chemoinformatics, a common subject for description is a molecule, which may be described in terms of a mass spectroscopy spectrum, an IR spectrum, its physico-chemical properties, three dimensional descriptors, amino acid sequence, etc. Another common dataset in biomedicine is microarray data, where the state of a cell-type is commonly the target to predict, and the expression levels of mRNA are the measured features. One may note that the vector description cannot completely represent all aspects of an object and that the representation also effects the design of algorithms. In other words, there are limitations to the space of problems that may be solved, inherited in the representation used. Other structures, such as trees, may prove to be more powerful in certain applications, but that is currently a field beyond the scope of standard pattern recognition.

Targets
The property of the data sample that one aims to predict based on the measured features is termed target. Taking a molecule as an example, common targets to be predicted are affinity, a biological effect that the molecule exerts such as toxicity, or the functional class that the molecule belongs to.

The domain of the target can either be discrete or continuous. If the domain is discrete then the possible values that the target may have are called classes, or labels, and the prediction is called classification. Returning to the example of a molecule, one may be classifying a protein to a functional class having two possible values such as allergenic or not allergenic. As for microarray studies the target can be disease state with values such as healthy, intermediate and end stage of disease.

To determine which class a given pattern belongs to the classifier divides the feature space into regions corresponding to different classes. In this way the

classifier defines a decision surface in the feature space, see figure 1. Classifiers are typically denoted as being linear or non-linear, which refers to the shape of their decision boundaries.

On the other hand, if the target domain is continuous then the target is commonly denoted response variable and the prediction procedure is called regression instead of classification. Toxicity of a molecule is an example of a continuous target.

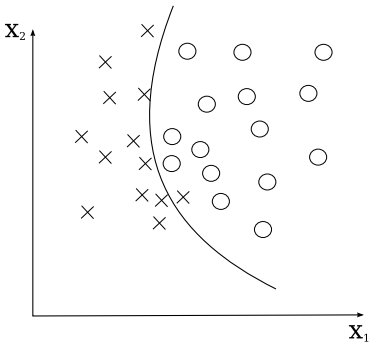


Figure 1: Discrete range of the target. The learning system aims to find a decision surface that separates the classes in an optimal way.

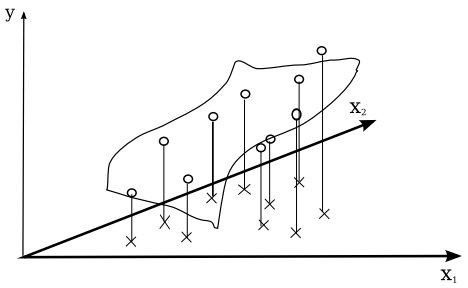


Figure 2: Continuous range of the target. The learning system aims to find a regression surface with predictions as close as possible to the true values.

The learning system

The goal of learning a predictive pattern recognition system may now be defined by help of these terms. Given a pattern, x, find a map, f, that predicts the target, t, as closely as possible, $t \approx \hat{t} = f(x)$. The learning procedure, referred to as training, consists of presenting the learning system with a number of data samples, called the training set, by which help the function f is being shaped.

Returning to the distinction between supervised and unsupervised learning we now have the tools to define the difference. In supervised learning the targets of the samples in the training set are known. In the learning procedure the system is adjusted so that its predictions, $\hat{t}$, are as close to the true targets, t, as possible. By contrast, unsupervised learning involves system training without access to any targets of the training data set.

As an example of supervised learning, we may have a dataset of substances

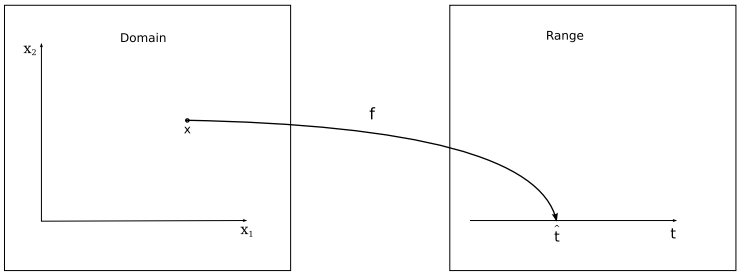


Figure 3: A learning system, f, is a mapping from the domain of patterns to the range of targets.

with measured toxicity in terms of IC_{50}[2] with which the system is trained. When a fresh data sample, which has not been part of the training set that shaped the system and for which the target is unknown, is being presented to the system a prediction of its toxicity is made.

Some examples of common classifiers and regression models used in supervised learning are *neural network*, *support vector machine*, *k-nearest neighbor*, *naive Bayes classifier*, *decision tree*, *linear discriminant analysis* and *logistic regression*.

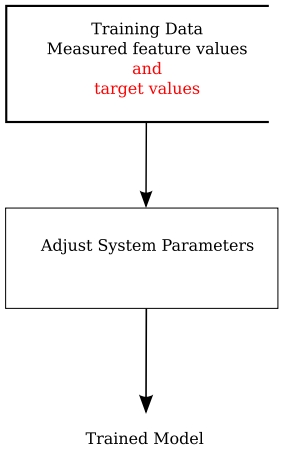


Figure 4: Supervised learning: The model parameters are adjusted by help of true targets of the training set.

The PRTools toolbox

The PRTools toolbox used in this work only implements supervised learning procedures. The two real world datasets studied in this work, a microarray dataset and a dataset of protein sequences, both have discrete targets, and therefore both constitute classification problems. However, the implemented model selection and validation procedures are general, in the sense that both classifiers and regression functions can be input as predictors.

[2]Inhibitory Concentration, the concentration required for 50% inhibition of the target.

Feature selection and representation
From the examples mentioned above, with patterns made up of mRNA levels or three dimensional descriptors, it may be realized that the problems are highly multivariate, as the number of features describing each object are in the size of thousands. This raises an important issue within pattern recognition, which features contain information of predictive capability of the target of interest? This leads to the concepts of feature selection and extraction, which probably is the most critical step in constructing a system with high performance. Feature selection is a central theme in this study, both in the model selection and validation algorithm as well as in the applications on the microarray and allergology datasets. Details about how the feature selection issue is handled is presented in the methods section.

A related issue is whether in the set of features relevant information is present in the first place. Finding a suitable representation of the problem is therefore a core part to its solution. If one manages to design features that perfectly capture the crucial information, then, in a classification scenario, the classes may be cleanly separated in the feature space rendering the problem of finding a decision boundary almost trivial. For example, it is difficult to find a decision boundary that separates clementines and oranges based on colour, but measuring the size it is trivial. The work concerning the allergology problem treated in this thesis mainly concerns the matter of constructing a better representation of protein amino acid sequences than those used by previous studies of the same problem.

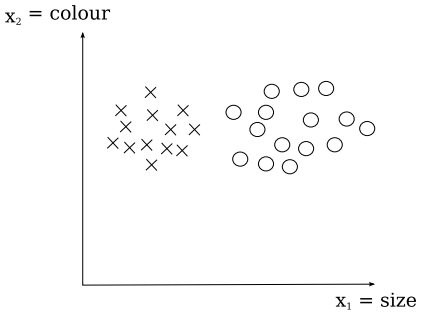


Figure 5: The issue of representing and selecting features that contain relevant information. In the illustrated example it is clear that we should measure and select the feature "size" in order to separate the classes.

1.4 Model Selection and Validation

To wrap up the pattern recognition overview the important issue of model selection and validation must be highlighted.

Model Selection
Given a number of models, in this case being pattern recognition systems, we seek to select the model that performs best. To pursue a rigorous model selection the choice should be based on a performance criteria. Some common criteria used to measure performance of a learning system are holdout tests, bootstrap techniques or cross validation, which all, in more or less elaborate ways, are based on dividing the data into training and test data sets.

Validation: Unbiased performance estimate
After a final model have been selected then it needs to be validated on data that has not been available for any type of model selection.

Here it is important to remind about a common mistake. Since model selection is based on a performance criterion, such as cross validation, it may be tempting to think that the cross validation measure then retrieved gives an unbiased measure of the model's performance. Clearly, this is not the case since the performance of all models have been measured and then the model that performs best on that data is selected. Since a selection of model has been made based on data, that model's performance is biased to that specific dataset. This phenomenon will be clearly illustrated in the application of breast tumour classification in this work.

Thus, apart from the training and test sets used for model selection, a validation set that has been locked into a valve during the model selection procedure is also necessary. This validation set can first be brought into light after the final model has been chosen. Naturally, the validation doesn't necessarily has to be performed by a holdout test, but any other method such as cross validation or bootstrap may be utilized.

Cross-validation error
In this work a prior-weighed cross-validation error is typically used as performance estimate. It is calculated by first computing a prior-weighed holdout error for each of the n holdout tests in a n-fold cross-validation. Each such prior-weighed holdout error, $J_{HoldOut}$, is calculated by dividing the number of misclassifications, e_i, of a class, i, with the total number of samples of that class, N_i. The weighed error for each class is then averaged over the classes by summing them and dividing that sum with the number of classes, see equation 1. The mean of these prior-weighed holdout errors is then calculated to give the final cross-validation error, CV, see equation 2.

Split of a dataset in n pieces for a n-fold cross-validation aims at arriving at as evenly distributed number of samples of each class as possible in the n pieces.

$$J_{jHoldOut} = \frac{1}{k}\sum_{i}^{k}\frac{e_i}{N_i} \quad (1)$$

$$CV(n) = \frac{1}{n}\sum_{j}^{n} J_{jHoldOut} \quad (2)$$

Validation: Which are the selected model parameters?
It is important to realize that all components of a model that are selected based on data are susceptible to overtraining. Notably, features are normally selected based on the data at hand and must therefore be considered as parameters of the model. To repeat the dogma, selected model parameters needs to validated against new data that has been held away from the selection procedure.

Validation: Sensitive performance estimate
Having retrieved an unbiased performance estimate, one should still be aware

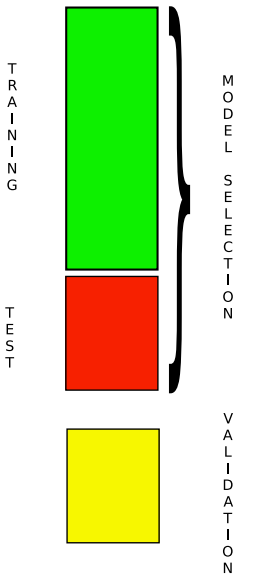


Figure 6: The available data needs to be split two times. An initial split to create a validation set that is to be held out from model selection and a split of the remaining data for tests underlying model selection.

that it is an estimate of the true performance of the model, and as such it is associated with a variance. If the performance estimates have a high variance one should be very cautious. Here the small sample issue needs to be highlighted. The smaller sample size the higher variance of the performance estimate. Cross-validation is probably not the optimal choice of error estimator for small sample sizes since its variance is higher than for a number of other error estimators, such as bootstrap and bolstering[10][7]. Estimating the variance of the performance estimate should be enforced as a standard practice.

Improved model selection criteria and unbiased performance estimate of the selected model are two main incentives behind the implementation of an automated model selection and validation algorithm performed in this work.

1.5 Application to Breast Tumour Classification

Background: Incentive for data collection
The microarray dataset studied was first published in *Nature* by van't Veer *et al.* in 2002[48]. The study aims to design a learning system that predicts whether a patient with breast cancer diagnosis will develop distant metastases or not. Breast cancer patients who don't develop metastasis generally have a good outcome and survive. Chemotherapy or hormonal therapy reduces the risk of distant metastases, but 70-80% of patients recieving this treatment would have survived without such intervention. The motivation behind constructing a classifier is that reliable prediction may allow patient-tailored therapy. In particular, patients for which good prognosis is predicted could be spared chemotherapy. The basic idea behind the prediction procedure is to use gene expression levels of a number of genes selected as clinical markers. Additionally, genes identified as good clinical markers are verly likely crucial actors in the biochemical mech-

anisms of breast cancer progression, and thereby potential drug targets.

Background: Outline of van't Veer's methodology

Next, the procedure used by van't Veer *et al.* for constructing the learning system is summarized, somewhat recast in a language in closer resemblance to pattern recognition terminology.

For each of 97 breast cancer patients a microarray measurement of approximately twentyfive thousand genes from tumour cells was performed. Out of these 97 samples, 78 were chosen for training of the learning system and 19 for a holdout test.

As an initial preprocessing step, genes with at least two-fold change of expression in at least three patients and with a p-value[3] less than 0.1 were selected. The intensity ratio utilized to reflect the change of expression is the intensity signal for a gene divided with the signal of a reference pool containing equal amounts of cRNA from each of the 78 samples. Approximately 5000 genes were found to have a significant two-fold change in expression.

The next feature selection step involved the identification of genes for which the expression level had a correlation to outcome larger than the absolute value of 0.3, $|c| > 0.3$. This resulted in that the set of around 5000 genes was reduced to 231.

The remaining 231 genes were ranked according to their correlation to outcome. A number of top-ranked genes from this set were selected, and used by a classifier for prediction. The top-ranked feature subsets input to the classifier was formed by starting at five features and adding five at a time. Feature addition was halted at 75 features where the performance of the classifier was seen to decrease. Classifier performance was measured in terms of leave one out cross validation error ($LOOCV$) on the 78 samples. The feature subset giving the best performing classifier was denoted the "optimal feature set".

The classifier used calculates the distance of a sample to the mean of the set of training samples belonging to the good prognosis class, where the distance is measured by means of the Pearson's correlation coefficient. If the "correlation" of a sample's feature profile to the mean of the good prognosis class exceeds a certain threshold then the sample is assigned to the good prognosis class. Conversely, a value below this threshold would assign it to the other class. Notably, this is similar to a nearest mean classifier, which calculates the distance to the mean of each class and assigns the sample to the class of the closest mean. The correlation threshold used was 0.4, which was found by optimizing the performance of the classifier. It is not clearly stated in the article how this optimization was done but it is not unlikely that they were making use of the 19 holdout data to tune the threshold parameter.

Using the described algorithm for search for an optimal feature set resulted in a LOOCV error that was monotonously decreasing up to 70 genes, hence, the final model chosen uses a feature set composed of the 70 genes which are most correlated to clinical outcome of breast tumour patients. To validate the 70-gene prognosis predictor the 19 samples of the holdout, in which 7 had good prognosis and 12 bad prognosis, were classified. The prediction resulted in only

[3]In statistical hypothesis testing the p-value is the probability that the value of the random variable t used as test statistic is the same as, or less favourable to the null hypothesis, the observed value, t_{obs}, assuming that the null hypothesis is true. Less favourable can mean greater than or less than depending on test statistic used.

two incorrect classifications, with one misclassification for each category.

Motivation

The aim of this study is two-edged. The main objective is to provide a real-world example dataset which allows the model selection and validation algorithm implementation to be tested. The microarray dataset provided by van't Veer *et al.* makes a good example since the described design and validation is not approached from a conventional pattern recognition framework.

First, the last feature selection step involving correlation to outcome ranked genes, added in groups of five, is far from exhaustive. The procedure is similar to the in pattern recognition terms so called individual feature selection where features are individually ranked according to a criterion and an optimal feature subset is found by adding features taken from the ranked feature set[5]. This is though a procedure that severely restricts the number of subsets covered. It is a so called univariate feature selection method, since each feature is ranked individually. It is therefore of interest to also apply multivariate methods to this problem since there may be combinations of features that could separate the classes better than a subset of individually best features. Correlation between gene expression levels should not be rare since genes belonging to a pathway commonly are co-regulated. Additionally, although if individual feature selection is being employed other ranking criteria apart from than correlation to outcome might also apply.

Second, the classifier used is also rather unconventional. Since the distance to the mean of samples from only one class is used to determine the category of a sample, the classifier only uses information from one of the classes. Commonly, classifiers use the information from samples of all classes involved.

Thus, because only one type of feature selection and only one type of classifier is evaluated it is intuitively sensible to extend the search space of learning systems including other feature selection methods and classifiers. The second objective for the study of the microarray dataset is therefore to find a predictor that performs better than the 70-gene predictor presented by van't Veer *et al* or a predictor that performs equally well but which is based on fewer genes. This is done by utilizing the implemented model selection algorithm.

It may also be noted that the validation of the designed classifier is somewhat unclear. As stated above, it is unclear whether the holdout test data has been used for optimization of the correlation threshold of the classifier. If this is the case, the reported performance estimate is biased. In a subsequent article by the same group published in the *New England Journal of Medicine*[47] the 70-gene predictor was assessed against a test set of 180 tumour samples. The performance was also shown to be markedly lower, with 47% of false positive, *i.e.* good prognosis patients assigned to the bad prognosis category, and 7% of false negative. Finally, one may note that designing a classifier with as much as 70 features based on 78 training samples is questionable in respect to its power of generalization[19]. It is not hard to realize that in the case of as many features as samples the classification may be based on features that are unique to each of the training samples, instead of features that represent a general signal across all training samples.

To conclude, model selection over a range of pattern recognition systems as well as proper validation where the training and model selection is strictly separated from the validation is called for in the examination of the microarray

dataset presented by van't Veer *et al.*[48].

1.6 Allergology and Immunoinformatics: Application to Allergen Classification

Background

Allergy has become a major plague throughout the world[2]. The combined prevalence of all types of allergy amounts to 25% of the population in many industrial nations[35]. The disease involves development of hypersensitivity towards otherwise harmless substances, mainly proteins, triggering an immune reaction. A major form of the disorder is designated IgE-mediated allergy, also known as type I hypersensitivity[44]. The substances causing this disease includes a wide variety of aerial proteins, typically occurring in tree- and weed pollen, as well as proteins present in a wide range of foods[3]. Proteins known to have the potential to cause allergic reactions are denoted allergens.

Acquirement of allergy consists of two separate phases, sensitization and triggering. Sensitization is the education of the immune system to respond to a given antigen. The education requires a series of complex cellular interactions but in essence leads to maturation of T- and B-cells into immunocompetent cells, where an immune response may be triggered by the specific antigen. The triggering is commenced when IgE antibodies anchored to mast cells or basophilic granulocytes bind to the antigen. This binding elicits release of inflammatory substances from the mast cell or basophil, resulting in symptoms of allergy. Typical symptoms are eczema, rhinitis and asthma, but the allergic reaction may also be more severe, causing anaphylactic shock, which is an acute impairment of circulatory and respiratory functions, in some cases leading to death[6].

As should be clear from the brief outline above, the specificity towards an allergen is dictated by the IgE immunoglobulin binding affinity to the allergen. The region of an antigen to which an IgE molecule binds is referred to as an epitope. Clearly, it is of great interest to investigate whether epitopes can be distinctly identified and whether it is possible to determine common features among epitopes. Pursuing such studies may single out a set of allergen-motifs which for instance may serve the purpose of predicting allergenicity. Identification of individual epitopes may be used as diagnostic reagents or as potential synthetic vaccines[38][29]. Accordingly, substantial amounts of research has been done in this field, constituting both experimental as well as bioinformatic studies. Most of our knowledge of protein epitopes has been obtained by so called peptide scanning studies[38], *i.e.* known allergens are split into overlapping peptide fragments, and the fragments are exposed to antibodies that have been raised against the antigen of interest. Such studies have identified many linear epitopes, defined as a contiguous array of amino acids in the polypeptide chain that do not require formation of secondary or tertiary structure for IgE to bind[21]. In contrast, conformational epitopes require secondary or tertiary structure before IgE binds, and are typically discontinuous, *i.e.* occurring as scattered patches along the linear amino acid sequence of the protein. In recent years an appreciable number of conformational epitopes have been at least partieally identified through random peptide libraries expressed in filamentous phages. This is also known as the mimotope concept[14]. Although discontinuous epitopes are probably more prevalent, linear epitopes are also believed to occur. In food allergens for example, linear epitopes may be overrepresented

since the immune system encounter allergens first after they have been partially denatured and digested by the human gastro-intestinal tract[21].

Motivation
The allergology problem under study in this thesis is to predict whether a given protein is allergenic or not. A second aim is to simultaneously find peptides, or epitopes, that may carry the allergenic function of the allergens. This study was conducted in collaboration with the *National Food Administration*, *Division of Toxicology*, which handles risk assessment of food products from genetically modified organisms and novel food products. Genetically modified organisms involves the risk of adventitious introduction of allergenic proteins in crop plants. A typical example is the 2S albumen from Brazil nut that in 1996 was transferred into soybean to improve the nutritional value. It turned out that patients allergic to Brazil nut but not to soybean showed an IgE mediated immune response towards the genetically modified soybean[27]. It is clear that the introduction of novel or new-in-context proteins in food crops requires proper allergenic assessment before being placed on the market. A FAO/WHO[4] special work group has devised a guideline for such risk evaluation, where the first step of the assessment involves the use of computational screening to evaluate the potential allergenicity of a protein. There are though considerable room for improvements of the predictive capability of the methods outlined in the protocol. For these reasons, research is being performed that concerns the design of higher performing predictors. Several methods superior to the FAO/WHO guidelines have been described recent years. Stadler and Stadler have proposed a sequence motifs based approach[37], Soeria-Atmadja *et al.* presented an algorithm named DFLAP[5][15], and most recently, Cui *et al.* reports a high performing method using an encoding of amino acids called CTD[6] combined with support vector machine classification[26].

To conclude, there are two objectives with the immunoinformatics study of this thesis. The primary objective is to design a learning system with good performance for detection of allergens. The secondary objective is to design a learning system which allows selection of peptide-fragments, epitopes, that may have importance for allergenicity.

1.7 Aims

The issues of creating improved and automated model selection and retrieving an unbiased performance estimate of the selected model are two main motivations behind the implementation of an automated model selection and validation algorithm done in this work.

The model selection and validation algorithm is applied to a real-world microarray dataset, pursuing classification of breast-cancer patients to prognostic category. The reason to approaching this problem is that a 70-gene predictor found to perform well on this dataset has been given considerable scientific and clinical attention[8][28][32][47][48]. The study may however be seen to be far from exhaustive and the validation of the model may also be questioned.

[4]Food and Agricultural Organization/World Health Organization
[5]Detection based on Filtered Length-adjusted Allergen Peptides
[6]CTD: Composition, Transition, Distribution

As a second real-world application, which also is the major part of this work, a problem within allergology is dealt with that aims to perform functional classification of protein amino acid sequences as being allergenic or not. The reason for approaching this problem is the growing problem of allergenicity and a growing attention to allergenicity among regulatory toxicologists, crop breeders and manufacturers of protein products for the consumer market. Accordingly, there is an increasing need of screening for proteins that may pose a risk. Additionally, a good model for an allergen-motif is yet to be found.

The add-on package developed as part of this thesis may also become useful in future computational biology research. Research utilizing this package has already been conducted within the group *Computational Medicine* belonging to the program of *Cancer Pharmacology and Informatics* at the department of *Medical Sciences*, *Uppsala University*. It may also be used as a pedagogic tool, by making up the base for student computer laborations in courses in Statistical Learning. In fact, it has already been used for that purpose, in the course *"Learning Systems for Molecular Data Analysis"* offered by the department of *Engineering Sciences*, *Uppsala University*. The aims of the thesis can be summarized as follows:

- **Pattern Recognition Toolbox**
  - Implement an algorithm that automates model selection and performs proper validation of the selected model.
- **Applications**

  Classification of prognosis for breast-cancer patients
  - Apply the model selection algorithm to elucidate whether a better performing predictor than van't Veer's 70-gene predictor can be found.
  - Apply the model selection algorithm to elucidate whether a smaller and more stable feature (gene) list may be obtained.

  Classification of protein sequences with respect to allergenicity
  - Apply the model selection algorithm to attempt to improve the performance of classification of proteins as being allergenic or not.
  - Perform feature selection on the given set of peptide fragments (FLAPs) to try to find epitopes.
  - Implement a recently proposed encoding (CTD) of amino acid sequences as to elucidate whether it can give predictors with higher performance than conventional alignment based approaches.

2 Methods

2.1 Model Selection and Composed Pattern Recognition Systems

The implementation of the model selection procedure implements the view that a pattern recognition system is not simply a classifier or regression function, but is made up of all components that processes the raw data before a prediction is being output. Accordingly, preprocessing and feature selection should be added to the predictor as to form a complete learning system, as illustrated in figure 7.

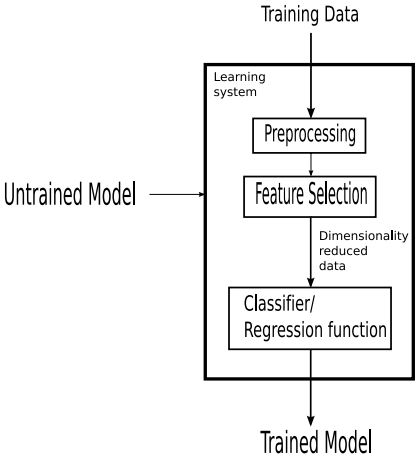


Figure 7: A complete learning system is composed of several components. The three main parts are preprocessing, feature selection and classifier/regression function. These components are seen as (meta) model parameters.

One advantage of such an approach is that a strict separation between training and validation is made more clear since it emphasizes that not only selection of classifier is based on the training data but also selection of all other model parameters, notably which features to include in the final model.

The model selection is viewed as an optimization problem, where the model parameters being varied as to maximize the optimization criterion involves feature selection parameters, preprocessing methods as well as classifiers and their corresponding parameters. The optimization criterion is the performance of the complete learning system.

Feature selection and preprocessing
Typically, the feature selection and the training of predictor together with assessment of the performance of the complete model are separated into two stages. This occurs when the feature selection is based on an information content criterion that is not a classifier or regression function. The feature selection process is then commonly termed filtering.

Examples of such information content criteria are euclidean distance between the means of the classes, mahalanobis distance and inter-intra distance. These criteria measures the degree of separation of the class distributions, which is

used as a metric to assess the discriminatory power of the feature subset. These criteria are in this work formally denoted I.

The feature subsets evaluated may be constructed in several ways. As the total number of possible subsets satisfies, $\frac{p!}{(p-n)!n!}$, where p is the total number of features available for selection and n is the number of features one wishes to select, an exhaustive search is only possible when a very limited number of features is at hand. Instead there are several algorithms that in an intelligent way tries to select and evaluate subsets that ought to be more promising than other subsets. Common methods are individual feature selection that ranks each feature individually and greedy forward feature selection that builds up the subset by adding the feature that increases the information of the subset being built the most. A genetic algorithm can also be used for feature selection purposes and its implementation and employment is described in section 2.5.6 below. These feature subset search path methods are denoted σ in this work.

The number of features that the search path method is instructed to find is denoted n. The optimal feature subset of size n that the feature selection search path method σ finds by help of the information content criterion I is denoted n^*.

The information of a feature subset is assessed by calculating the information criteria I based on all available training data, giving a so called apparent estimate, or by using less biased error estimators such as holdout or cross-validation. The error estimators used to retrieve a criterion for feature subset assessment are here denoted θ_ϕ.

The complete feature selection process, ϕ, is summarized in equation 3 below. In the equation is also present a parameter ψ, which symbolizes preprocessing of the data D. The preprocessing employed in this work constitutes the use of more coarse-grained feature selection methods.

$$n^* \quad = \quad \phi(D, n, \sigma, I, \theta_\phi, \psi) \tag{3}$$

Model assessment and predictor component

Having retrieved a promising feature subset the data is reduced to n^* features and is thereby ready to be input to a predictor being either a classifier or a regression function, here denoted ω. Common classifiers used in this work are k-nearest neighbor, which assigns a sample to the class which the majority of its k neighbors belongs to, and the parzen density classifier, which is a kernel density estimator with a normal distribution as kernel. The complete model is then assessed by an error estimator, θ_Ω, such as cross-validation or holdout, giving a performance estimate, J, of the model. This is formally stated in equation 4 below.

$$J \quad = \quad \Omega(D, n^*, \omega, \theta_\Omega) \tag{4}$$

Equation 3 and 4 are fused when the feature selection and assessment of model performance are integrated. This happens when the feature subset information content criteria, I, is a classifier or regression function, thereby measuring the information content of the feature subsets by a more direct evaluation of their predictive capability. In equation 3 above I is then replaced by ω and

the optimization criterion, J, in terms of classification performance, is output as well as an optimal feature subset, n^*.

Varying the parameter values, such as the feature selection search path method used or informativenss criterion used, a set of models is generated, each associated with a certain value of the model selection criterion, J. Typically one chooses parameter domains spanning a few possible values for each of the input parameters, and then forms all possible combinations of the parameter values as to search through some part of the model parameter space. Among these models the one with best performance is preferably chosen, directed by the criterion, J.

The functions and output parameters as well as the eight input parameters described above are summarized in table 1 below. This description, utilizing eight input parameters to specify, also reflects the implementation.

Functions and output parameters			**Eq.**
ϕ	Feature selection process		(2)
n^*	An optimal feature subset of n features. This subset may be determined separately from the classification performance, if the information content criterion used is not a classifier but rather some distance measure that evaluates the separation of the classes in the feature space. The dataset input to Ω is reduced to the n^* identified features by a trivial function ρ, $D^* = \rho(D, n^*)$.		(2)
Ω	Training of predictor and performance assessment of the complete model.		(1)
J	Optimization criterion. The error estimator that outputs J is specified by θ_Ω.		(1)
Input parameters		**Model Example**	
D	Dataset	microarray: 78 samples. Classes: 34 good prognosis, 44 bad prognosis.	(1),(2)
n	Number of desired features to reduce the dataset to.	70	(2)
σ	Feature selection search path, determines the way to traverse the space of possible feature subsets.	forward	(2)
I	Feature subset information content criterion, specifies how to evaluate the amount of information contained in the feature subset of interest.	euclidean distance	(2)
θ_ϕ	Error estimator for feature subset selection.	cross-validation(10-fold)	(2)
ψ	Preprocessing method.	feature selection: 2-fold change in expression levels	(2)
ω	Classifier and parameter values for that classifier.	kNN(3)	(1)
θ_Ω	Error estimator for model selection. May for example be 10-fold cross-validation.	cross-validation(5-fold)	(1)

Table 1: Input and output parameters specifying the model selection procedure. Examples of values of the input parameters specifying a single model are presented.

D	blosum62	Sequences represented purely by their amino acids. BLAST alignment used as similarity measure.
	CTD-7	CTD-encoded amino acid sequences using seven physicochemical attributes. Euclidean distance between encoded sequences was used as similarity measure.
	CTD-ZZ	CTD-encoded amino acid sequences using five multidimensional scaling latent variables. Euclidean distance between encoded sequences was used as similarity measure.
σ	individual	Individual feature selection. Features are individually evaluated according to their information content and then ranked.
	forward	Greedy forward feature selection builds up the subset by adding the feature that increases the information of the subset being built the most.
	Genetic Algorithm	A feature subset is represented as an individual of a population. The population is then evolved as to find an optimal individual. Further details are found in section 2.5.6 below.
I	euclidean	Euclidean distance between the means of the classes.
	mahalanobis distance	$(m_2 - m_1)^T S_W^{-1}(m_2 - m_1)$, where S_W is the pooled within-class sample covariance matrix, $S_W = \sum_{i=1}^{C} \frac{n_i}{n}\hat{\Sigma}_i$, where m_i and $\hat{\Sigma}_i$ are the estimated means and covariance matrices of each class. Assuming that the distributions are normally distributed and with equal covariance matrices this distance is equal to the divergence measure, $\int [p(x\|w_1) - p(x\|w_2)] log(\frac{p(x\|w_1)}{p(x\|w_2)}) dx$
	inter-intra	This distance calculates the distance between the means of the classes and also takes into account how the classes internally are distributed.
	correlation with outcome(c)	Features are individually ranked according to how well they correlate with the target variable. A correlation coefficent threshold, c, is set so that only features with the absolute value of the correlation being higher than c is selected.
ω	distance to mean(class, c)	The distance to the mean of the specified class is calculated. The distance is measured in terms of the "correlation" between the sample and the mean. If the correlation is above the specified threshold, c, then the sample is assigned to the class to which the distance is calculated.
	fisher	fisher's discriminant
	nmc	nearest mean classifier
	lda	linear discriminant analysis
	qda	quadratic discriminant analysis
	kNN	k Nearest Neighbors classifier
	parzendc	parzen density classifier
	decisiontree (pruning)	'maxcrit' and 'fishcrit' specifies pruning parameters of the decision tree.
ψ	expression change (intensity ratio, p-value, n samples)	Selects genes that are differentially expressed. Three criteria need to be fulfilled for a gene to be selected. The intensity ratio specifies the minimum ratio with which a gene needs to be expressed in relation to a reference mean as to be selected. The p-value specifies the probability that the observed intensity ratio may have occurred due to random noise, see section 1.5 for further explanation. A p-value threshold is set which is not allowed to be trespassed. The number of samples specifies in how many samples the gene needs to fulfill the two first criteria as to be selected.
	derivedFLAPs	Selects peptide fragments (FLAPs) which are derived from the allergens of the training set.
θ_ϕ and θ_Ω	apparent	Apparent error rate, results from a test based on the training data, which has not been held out from the training procedure.
	cv(n)	Cross-validation, where n stands for the number of folds.

Table 2: Reference table of values used in this work of the model input parameters. For further details regarding the parameters see the references[5][1].

2.2 Validation

In the former subsection the implemented model selection procedure was described, which leaves us to the issue of validating the selected model. The model selection described above utilizes a dataset, D, to find the best model. As to perform an unbiased validation of the selected model a separate dataset, V, which has not been part of the model selection procedure is needed. Hence, among all models the model with optimal value of the optimization criterion, J, is selected and tested against the validation set, V. The validation can be done by any conventional error estimator, and those implemented in this work were cross-validation and holdout test.

As validation methods also are used for retrieving performance criteria, J, upon which model selection is based, these methods are being applied twice. This results in that the the data needs to be split twice. First in an outer split, dedicating one piece of the data to validation and the other for model selection. Then the piece of data dedicated to model selection is further split to train and test the set of models as to give performance measures which provide guidance for model selection. Thus, a common scenario is that a cross-validation procedure is utilized twice, both in an inner loop for model selection as well as an outer for validation. This type of double cross-validation loop procedure has been reported in several articles[9][16][18]. Such a "double-loop" is also realized by the implementation in this work.

2.3 Model Selection and Validation Algorithm: Test of Implementation on Simulated Mixed Gaussian Data

A test problem was designed as to verify the correctness of the implementation. A simulated dataset of two classes was generated, where each class was generated by a separate 3-dimensional Gaussian distribution. The mean values of the distributions were (0,0,0) and (2,1,0) respectively, and with correlation matrices

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix} \textit{and} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}.$$

The models were allowed to vary in terms of two different feature selection methods and four different classifiers. Among the classifiers was quadratic discriminant analysis (qda) which should perform well since it assumes that the data to be classified is generated from Gaussian distributions, which in this experiment also is the case. The model selection was performed on two different datasets, one smaller with 10 samples from each class, and a larger one with 100 samples from each class. The final selected models for each dataset were validated against a validation set of 100 fresh samples from each class.

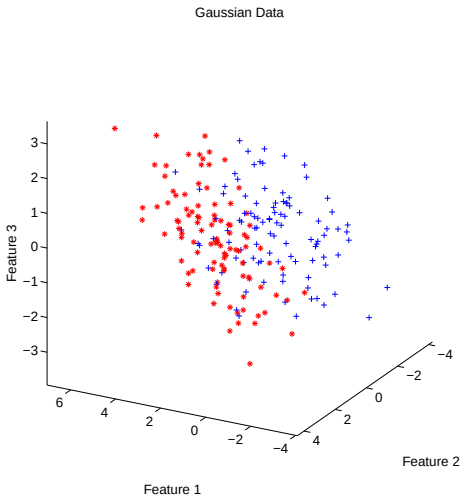


Figure 8: Scatterplot of two 3-dimensional Gaussians. 100 samples from each Gaussian.

Experimental Setup: Two 3-dimensional Gaussians.	
Parameter	**Value**
D	2 Gaussians. Small dataset: 20 samples, 3 features. 10 samples from each class. Large dataset: 200 samples, 3 features. 100 samples from each class.
n	2
σ	Individual, Correlation with outcome
I	euclidean distance
θ_ϕ	crossvalidation(5-fold)
ψ	none
ω	distance_to_mean(distance to mean of class 1, correlation threshold=0.0), fisher, knn(3), qda
θ_Ω	crossvalidation(10-fold)
V	200 sample holdout, 100 samples from each class.

Table 3: Parameter settings of learning systems for classification of simulated Gaussian data. Two different feature selection search paths and four different classifiers were used. fisher: fisher's discrimant, kNN: k Nearest Neighbors, qda: quadratic discriminant analysis.

2.4 Application to Breast Tumour Classification

The 78 tumour samples used by van't Veer *et al.*[48] were employed in an attempt to find a predictor yielding higher performance. The range of learning systems searched was formed by taking all possible combinations of the input parameters listed in table 4 below. Notably, the feature selection is of filter type as the information content criteria are not classifiers. Another considerable difference to the procedure presented by van't Veer *et al.* is that three different classifiers are evaluated here, in contrast to a single one only. All three are though relatively simple in the sense that they all have linear decision boundaries. Three different validations of the model selection were performed. The 19 holdout samples used in the *Nature* publication[48] and the 180 samples from the publication in *New England Journal of Medicine*[47], were applied in two separate holdout tests as to enable comparison of the selected model with the 70-gene predictor presented in these articles. Additionally, a 5-fold outer cross-validation was also used, hence forming a double loop since the model selection is based on 10-fold cross-validation. In this last case of outer cross-validation the 78 and 19 samples were pooled before input to the double-loop procedure. The reason for choosing 5-fold cross-validation is that the number of samples available for model selection are then approximately 78, allowing better comparison with the holdout performance estimates. The 5-fold external validation procedure was repeated 5 times with different splits of the dataset used for the cross-validation. This results in that the model selection procedure is being called 25 times in total. The experimental setup describing all parameter settings is presented in table 4 below.

Experimental Setup: 19 sample and 180 sample holdout validation	
Parameter	**Value**
D	Microarray dataset of breast cancer tumours. Dimensions: 78 samples, 24 481 features. Classes: 44 good prognosis, 34 bad prognosis.
n	5, 10, ..., 75
σ	Individual FS, greedy forward FS
I	mahalonobis distance, euclidean distance, inter-intra distance
θ_ϕ	apparent
ψ	correlation_with_outcome_FS(correlation threshold = 0.23) expression_change_FS(intensity ratio ≥ 2, $p_{value} < 0.01$, ≥ 3 samples)
ω	distance_to_mean(distance to mean of good prognosis class, correlation threshold=0.4), fisher, nmc
θ_Ω	crossvalidation(10 fold)
V	19 sample holdout: Classes: 7 good prognosis, 12 bad prognosis. 180 sample holdout: Classes: 138 good prognosis, 42 bad prognosis.

Table 4: Parameter settings of learning systems for breast tumour classification with microarray data using holdout test as validation error estimator. apparent: performance is evaluated on the training data, nmc: nearest mean classifier.

Experimental Setup: 10-fold crossvalidation	
Parameter	**Value**
D	Micro-array dataset of breast cancer tumours. Dimensions: 78+19=97 samples, 24 481 features. Classes: 51 good prognosis, 46 bad prognosis.
θ_Ω	crossvalidation(10 fold)
V	crossvalidation(5-fold). Repeated 5 times.

Table 5: Parameter settings of learning systems for breast tumour classification with microarray data, using cross-validation as validation error estimator. Only the dataset differs from model selection parameters used in the holdout validation. θ_Ω is shown as to clarify that a double cross-validation loop is being run.

2.5 Application to Allergen Classification

A novel representation of the protein sequences was designed. The design was mainly inspired from two articles, Dubchak *et al*[23] and Soeria-Atmadja *et al.*[15], the former introduces a novel representation of protein sequences and the latter presents a novel method for extraction of peptide fragments from allergens. However, none of the methods presented in these articles can without modification be directly applied to select peptide fragments that contain a general allergen-motif. In the article by Soeria-Atmadja *et al.*, a set of slightly less than 5000 peptide fragments forms the basis for classification, which is a too large set as to enable interpretations of biological relevance. The goal was therefore to design a learning system that may reduce the set of peptide fragments input to the trained learning system. Apart from the possibility that one may find a set of peptides that carry information about allergenicity, such a learning system may also be able to achieve higher performance as less features lessens the noise and thereby the risk for overtraining on the training data.

2.5.1 Summary of DFLAP algorithm

The DFLAP algorithm described by Soeria-Atmadja *et al*[15] is here outlined since the algorithm designed in this study is a modification and extension of that algorithm. The algorithm is illustrated in figure 9.

A database of 762 allergens is split into a training set of 500 allergens and a holdout validation set of 262 allergens. The allergens are cut by a sliding window of length, l. The generated fragments are then aligned, by the application of a BLAST algorithm using the BLOSUM 62 substitution matrix, against a set of non-allergens. The non-allergen set is composed of all sequences of the human proteome that do not have annotations related to allergenicity as well as non-allergen sequences from skin prick preparations. A fragment derived from the allergens having too high sequence similarity to any of the non-allergens is discarded, where the critical level of similarity is determined by an alignment score threshold, f. The peptides left after this filtering process of the allergen generated peptide fragments is referred to as *FLAPs*, *Filtered Length-adjusted Allergen Peptides*. The length-adjustment is performed whenever two or more consecutive peptides, where consecutive peptides are peptides where the sliding window have slided only one amino acid, all are deemed being dissimilar enough to the set of non-allergens and ought to be assigned as FLAPs. These overlapping peptides are then concatenated into one single unit, resulting in a set holding peptides of variable length.

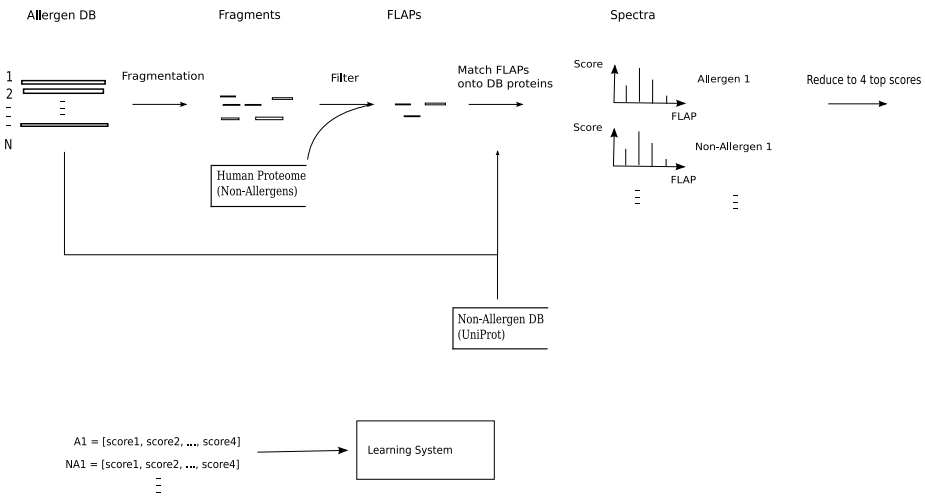


Figure 9: The DFLAP algorithm. FLAPs are extracted from a set of allergens. The FLAPs are filtered and then aligned against a set of allergens and a set of non-allergens. The four highest scores of each alignment entails a feature vector for each protein sequence. These features are used as input to a learning system.

The idea behind this filtering process is that the set of FLAPs ought to form a base of peptides with a higher density of allergenicity information than that of a non-filtered set of allergen peptide fragments.

Out of the training set of 500 allergens a set of several thousands FLAPs is generated. Next, the features describing a protein sequence is defined by aligning each peptide from the FLAP set against a protein sequence, retrieving an alignment score for each FLAP. The similarity of a FLAP against a sequence is defined as a feature, hence giving a feature set of several thousands of similarity scores. Since this feature space is of too large dimension to be of sensible use for classification, a somewhat unconventional method for reduction was invented. The alignment scores were sorted in descending order followed by selection of the top n similarity scores for each sequence. These n scores make up the final feature vector describing the protein sequence.

Subsequently a linear kernel support vector machine (SVM) classification algorithm was trained utilizing the 500 allergens of the training set as well as 1000 non-allergen training samples obtained by randomly sampling proteins from Swiss-Prot (followed by removal of possible allergens). Additionally, 5 vertebrate tropomyosins, regarded as non-allergens, were included in the training data phase. Tropomyosins originating from non-vertebrates are commonly found to be allergenic, whereas tropomyosins found in vertebrates is not known to induce an allergic reaction. Since sequences of the tropomyosin family are homologous across species and contains both allergenic as well as non-allergenic representatives, the members of this family are difficult to predict with respect to allergenicity.

Using this training data the four parameters of the learning system, that is, the sliding window length, l, the similarity threshold for filtration of FLAPs, f, the number of alignments scores selected as features, n, and the SVM cost parameter, C, were optimized using 3-fold cross validation. The optimal model selected was found to have the following values of its parameters, $l = 22$, $f = 48$,

$n = 4$ and $C = 100$, which results in a filter that returns 4776 FLAPs.

The performance of the selected system optimized and trained on the above described training data was evaluated by a holdout test of the remaining 262 allergens. It was also tested on 193 non-allergens sampled from 3 "difficult" protein families, each of them known to contain both allergens and non-allergens. The three "difficult" families represented were tropomyosins, parvalbumins and profilins. The performance achieved by the DFLAP classification system is one of the best ever published, and ought therefore to make a good starting point for design of a new detector of allergenic proteins.

2.5.2 Modification of DFLAP: from Feature Extraction into Feature Selection

As aforementioned, one drawback of the DFLAP algorithm is that the trained learning system requires as much as 4776 features as input. The reduction into four features made before input to the SVM classifier is done uniquely for each query protein, resulting in that the four peptides selected may be unique for each query. This procedure clearly makes it difficult to find peptides that can represent generalized epitopes. It is also probable that a substantial fraction of all 4776 features contains information of little relevance to allergenicity and such uninformative features may therefore cause the DFLAP predictor to perform worse than hypothetically possible.

To overcome this problem a reduced feature set that is the same no matter what the query sequence is would be preferable. Designing an algorithm that allows the same selected features to describe all sequences and using those same features for classification, could potentially result in the selection of generalized epitopes.

A method where a learning system requires a larger set of features as input, but then the feature set is reduced before classification is generally called feature extraction. A well-known method used for feature extraction is principal component analysis. In this view, it is clear that the DFLAP algorithm involves a feature extraction step, since the original set of 4776 features is required as input but the dimensionality is reduced to four before classification. In contrast, the method called for, where it is set beforehand what subset of peptides is to be input, is an instance of what is known as feature selection.

The DFLAP algorithm was therefore modified as to allow feature selection, see figure 10.

This feature selection process may be more clearly understood by viewing the FLAP alignment scores against a certain protein sequence in a spectrum. Since all features have the same unit of measure it makes sense to put them in a spectrum, having FLAPs on the x-axis and their corresponding alignment scores on the y-axis. Such a spectrum can be regarded as a fingerprint of a protein sequence. Such a fingerprint of a sample are also commonly denoted profile. As described in the section "Model Selection and Composed Pattern Recognition Systems" above, we let the feature selection procedure be a component of a learning system, and in this way we may also utilize the methods implemented in the PRTools toolbox. The set of model parameter values evaluated is described under "Experimental Setup" in the end of this section.

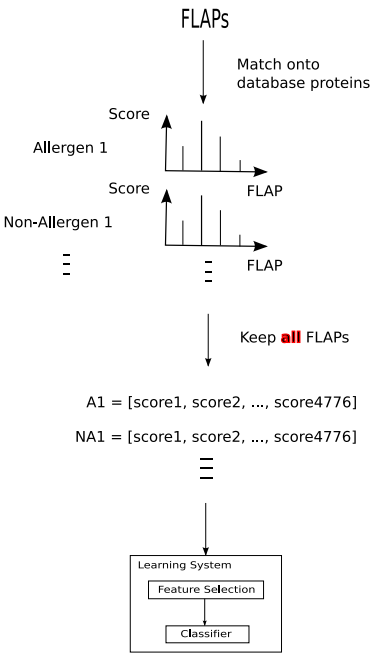


Figure 10: Modified DFLAP algorithm as to perform feature selection instead of feature extraction. All FLAPs are kept instead of only taking the four top-scoring features for each protein sequence. All these features (FLAPs) are instead selected among by conventional feature selection processes, thereby enabling the possibility to select general patterns.

2.5.3 Modification of DFLAP: Feature Representation

To aquire ideas of possible improvements to the feature representation used by the DFLAP algorithm, the problem of peptide and protein representation in terms of features is given an overview. This may help to distinguish and facilitate reflections on pros and cons of the DFLAP representation.

To describe a protein in terms of a peptide, we assign a similarity score that describes the similarity of the peptide to the protein sequence. To be able to define this similarity two components are needed, a representation of the peptide and protein, as well as a "similarity measure", that defines the distance between the two in a given representation.

An intuitive way to represent peptides and proteins is in terms of their amino acid sequences. Using this representation, the most widely employed technique to measure similarity is to apply a dynamic programming algorithm, such as BLAST, using a substitution matrix based on evolutionary amino acid substitution frequencies. As an oversimplified generalization it may be said that in this similarity measure two amino acids are more similar if they are statistically found to be more frequently substituted with one another. One disadvantage of this type of conventional alignment is that the representation is "local", in the sense that the substitution of amino acids may not always reflect substitution of other more global properties of a sequence. We can for example imagine that there can be two sequences that may differ on the amino acid sequence level, having rather rare substitutions, but that they form a tertiary structure which is very similar. Such conserved structural motifs may be hard to

represent with just information about the conservation of amino acid residues, at least if the substitution matrix is based on sequences from a range of different functional and structural classes.

As a way to alleviate the issue of "locality" that the algorithms employing substitution matrices based on amino acid distances suffers from in this thesis project, a more global representation was implemented. This representation was originally presented by Dubchak *et al.*[23] for the purpose of protein fold classification, and has more recently been implemented in a web server called SVM-Prot[13] and which has been successfully applied to a number of classification problems[33][34][22] one being allergen classification[26]. The high performance of the classifiers based on this representation made it an interesting target for implementation. This more global representation of polypeptide chains will here be denoted CTD-encoding, and is described below.

We may observe that in studies by Dubchak *et al.*[23][24][12] and subsequent publications utilizing essentially the same encoding, notably by employment of the web servers SVM-Prot[13] and PROFEAT[49], the complete amino acid sequence of a protein has been the direct subject for CTD-encoding, resulting in a CTD feature vector representing the whole protein sequence. The representation of a protein is in other words not based on a set of peptides, which is the focus of the work presented here. The representation presented here is therefore novel, forming a fusion and extension of the ideas contained in the DFLAP algorithm and the CTD-encoding.

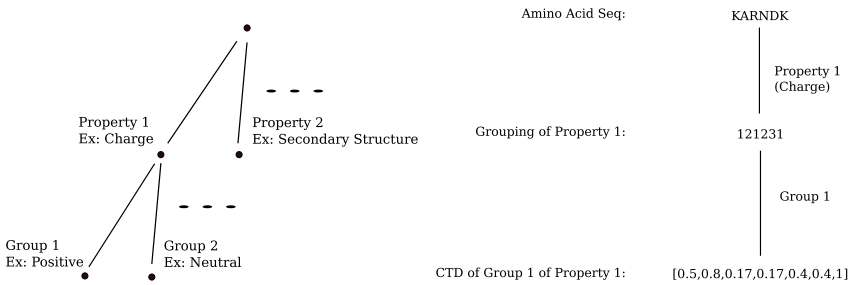
2.5.4 CTD-encoding

Algorithm

The basic units used in CTD-encoding are amino acid properties. Straightforward amino acid properties are physico-chemical attributes, such as weight, charge or hydrophobicity. Let's consider one such amino acid property to be subject for CTD-encoding.

An amino acid property gives rise to a set of 20 values, one for each amino acid. The initial step of the encoding is to categorize the amino acids into one of several groups with respect to the value of that property. In other words, a number of amino acids that have a similar value for that property should be mapped to one discrete group. Dubchak[23] as well as Li *et al.*[49] group the amino acids into three categories for each specific amino acid property. The sequence of amino acid letters is by this procedure replaced by a sequence of numbers representing each amino acid's category assignment, with respect to the amino acid property being encoded, see figure 11.

Taking charge as an example, there are three natural categories: positive(K,R), negative(D,E) and neutral(remaining amino acids). Next, three descriptors, composition (C), transitions (T), and distribution (D), are introduced to describe the global composition of the property. The C, T and D descriptors are calculated for each category of the property. Composition is the number of amino acids belonging to one category divided, with the total number of amino acids in a protein sequence. Transitions characterizes the frequency with which one category is followed by any other category of the property. For example, for a sequence of 1112113, where the digits represent categories of the property, there are three transitions from or to category 1, resulting in a transition frequency, $T = 3/(7-1) = 0.5$, since the number of possible transitions is the



(a) The amino acid sequence is translated into groupings within each property. Each group for a property is then CTD encoded.

(b) Example of coding of one group of one property.

Figure 11: The CTD-encoding algorithm is based on amino acid properties, grouping of these properties and finally calculation of the "CTD" feature vector for each group.

number of amino acids in the sequence minus one. Distribution refers to the proportion of the chain length within the very first, 25%, 50%, 75% and 100% of the occurences of the particular category is covered. In the above example, 100% of the first category is covered after six of the seven amino acids, hence, $D(100\%) = 6/7$. CTD-encoding of one category of one property thereby results in $1 + 1 + 5 = 7$ features. For a property of three categories, such as charge, $3 * 7 = 21$, features is generated. The complete feature vector is formed by using a multitude of amino acid properties, and the size of the final feature vector representing a sequence is therefore $21 * n$, where n is the number of amino acid properties, assuming that all properties were divided in three categories.

Properties Subject for Encoding

Physico-chemical attributes are a straightforward choice for the type of amino acid properties suitable for CTD-encoding. A sensible question to ask is which physico-chemical attributes are most appropriate to encode. Preferentially, one would like to restrict the number of amino acid properties encoded to a minimal set of properties that contains the most physico-chemical information about the amino acids.

Seven physico-chemical attributes

In a later article by Dubchak *et al.*[24], the number of CTD-encoded amino acid attributes had increased from three to six. They are: hydrophobicity, normalized van der Waals volume, polarity, polarizability, secondary structure and solvent accessibility. References to the original measurements of each of these attributes are found in their article. To quote the authors, the reason behind choosing these six properties is motivated by that they are "representing the main clusters of the amino acid indices of Tomii and Kanehisa"[30]. In the article by Tomii and Kanehisa a set of 402 so called amino acid indices was clustered. An amino acid index is defined as a set of 20 numerical values representing any of the different physico-chemical attributes of amino acids. These amino acid indices have been mined from publications and gathered in a database. There are more aa-indices than amino acid properties represented in the database, since several measurements of the same amino acid attribute may have been reported in separate publications, and also because there may be several definitions of

the amino acid property, such as is the case for hydrophobicity. Interestingly, Tomii and Kanehisa state that the volume and hydrophobicity can be used to reproduce the PAM substitution matrix to a very high degree, and that these two properties are the major factors influencing the amino acid substitution during evolution. Dubchak *et al.*[24] divided the above mentioned six attributes into three categories for CTD-encoding. Li *et al*[49] used the same six attributes as Dubchak and the same division into categories but also added the attribute charge, though without any motivation found being stated in the article. It is obvious that the information contained in the hydrophobicity attribute ought to be highly correlated to charge.

In this work, these same seven attributes were employed for CTD-encoding and with the same division of categories, see table 6. The seven attributes, each divided into three categories, results in a feature vector of length $3 * 7 * 7 = 147$, since each category gives rise to 7 features as described above.

Attribute	Category 1	Category 2	Category 3
Hydrophobicity	Polar	Neutral	Hydrophobic
	R,K,E,D,Q,N	G, A, S,T,P,H,Y	C,L,V,I,M,F,W
Normalized van der Waals volume	Small	Medium	Large
	G,A,S,C,T,P,D	N,V,E,Q,I,L	M,H,K,F,R,Y,W
Polarity	Low	Medium	High
	L,I,F,W,C,M,V,Y	P,A,T,G,S	H,Q,R,K,N,E,D
Polarizability	Low	Medium	High
	G,A,S,D,T	C,P,N,V,E,Q,I,L	K,M,H,F,R,Y,W
Charge	Positive	Neutral	Negative
	KR	ANCQGHILMFPSTWYV	DE
Secondary structure	Helix	Strand	Coil
	EALMQKRH	VIYCWFT	GNPSD
Solvent accessibility	Buried	Exposed	Intermediate
	ALFCGIVW	RKQEND	MPSTHY

Table 6: Amino acid attributes and the division of the amino acids into three groups for each attribute used in this work. For such attributes as secondary structure and solvent accessibility the division is based on statistical appearance of each amino acid in a specific state.

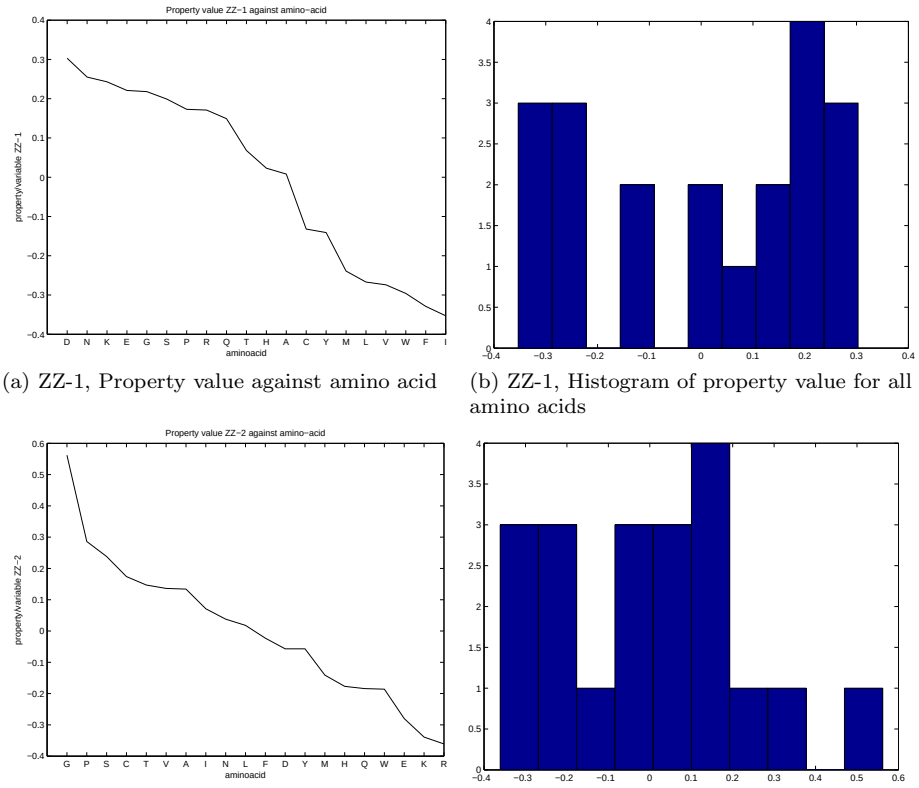
Five latent variables from multidimensional scaling of 237 physico-chemical attributes

The issue of finding a minimal set of properties that represents the difference between the amino acids in the physico-chemical space, can be approached in other ways than by means of clustering. By viewing the amino acids as data points in a space spanned by physico-chemical features, we can identify that this is a matter of reduction of feature space where one seeks to keep the information contained in the distribution of the dataset. This type of reduction is commonly done by feature extraction methods such as PCA[7] or multidimensional scaling, which both extracts latent variables that are ranked according to how much information they contain[5]. In multidimensional scaling latent variables are sought and ranked in respect to how well they are able to reconstruct the geometrical configuration of the point set, that is, it seeks to conserve the distances between the points when represented in a lower dimensional space. The perhaps most comprehensive analysis utilizing a method similar to multidimensional scaling

[7]Principal Component Analysis. Orthogonal latent variables that maximizes variance are extracted.

for the problem of reducing the large redundancy in physico-chemical properties was reported in year 2001 by Venkatarajan and Braun[45]. They report five latent variables that were found to be able to represent the distances between the amino acids in an original space of 237 physico-chemical properties.

These five variables were in this work identified to be good candidates for CTD-encoding, and will henceforth be referred to as ZZ-variable one to five. As mentioned above, the first step of CTD-encoding is to categorize the amino acids into one of several groups for each property. It was decided that each of the five ZZ-variables were to be divided into three categories, and this was done by partitioning the value range into three equally sized intervals. This decision was taken after having sorted the 20 values in descending order for each of the variables and inspecting the plots of these values against their corresponding amino acids. No plateaus were seen in the plots, which for example would arise if all amino acids can take only one of either of two values, see figure 12.



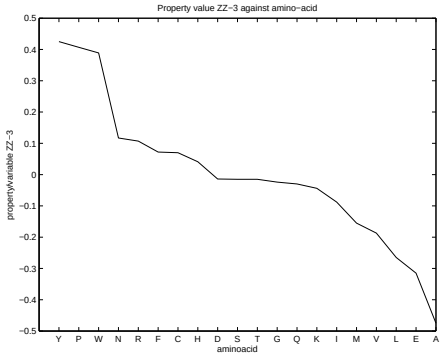
(a) ZZ-1, Property value against amino acid

(b) ZZ-1, Histogram of property value for all amino acids

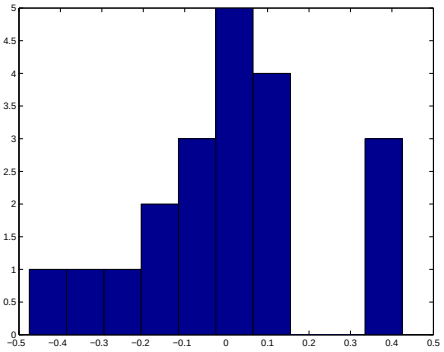
(c) ZZ-2, Property value against amino acid

(d) ZZ-2, Histogram of property value for all amino acids

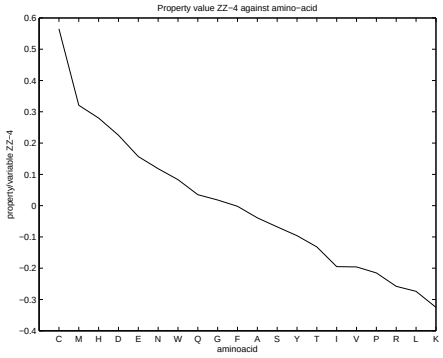
Figure 12: Amino acid values on the five most prominent multidimensional scaling vectors. No specific group of amino acids is identified.



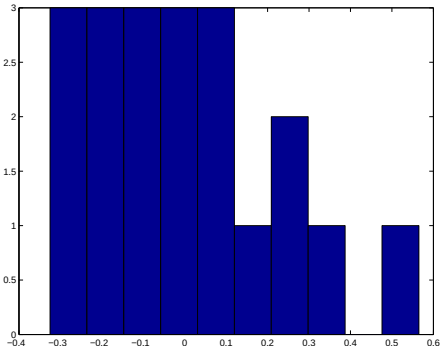
(e) ZZ-3, Property value against amino acid



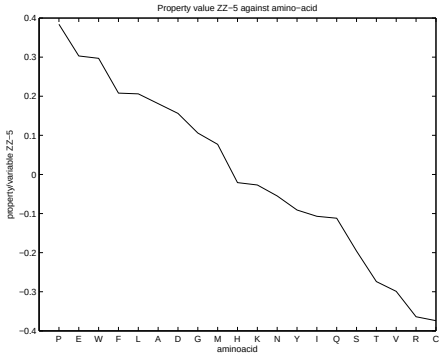
(f) ZZ-3, Histogram of property value for all amino acids



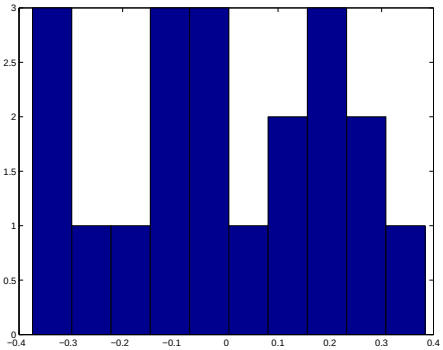
(g) ZZ-4, Property value against amino acid



(h) ZZ-4, Histogram of property value for all amino acids



(i) ZZ-5, Property value against amino acid



(j) ZZ-5, Histogram of property value for all amino acids

Figure 12: (cont'd) Amino acid values on the five most prominent multidimensional scaling eigenvectors. No specific group of amino acids is identified.

Similarity Measure

Given a representation in CTD-encoding, how then to measure the similarity between a peptide and a protein sequence? There are no substitution matrices based on that representation so a dynamic programming algorithm would not be directly applicable. A straightforward idea is to take the euclidean distance between CTD feature vectors representing the sequences. In analogy with conventional local alignment it would then be reasonable to compare the peptide for which similarity is to be determined with peptide fragments from within the query protein sequence. This is implemented such that the peptide fragments of a protein sequence is generated by a sliding window of fixed length l. The euclidean distance between the peptide and each of the sliding window generated peptides are then generated. The shortest distance among these is chosen as the similarity-score between peptide and protein. An overview of this algorithm is given in figure 13.

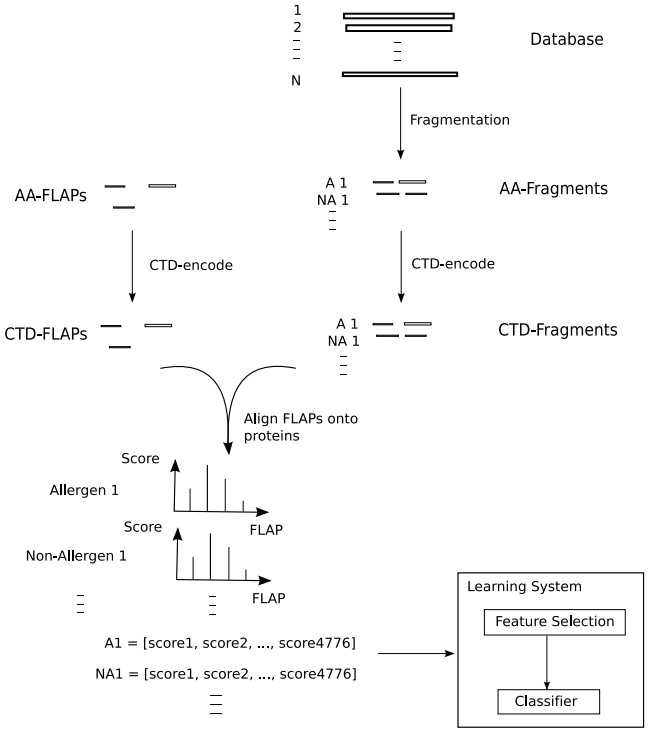


Figure 13: DFLAP algorithm with the representation of protein sequences modified as to be based on CTD-vectors instead of amino acid vectors. Alignment of CTD-encoded sequences is performed in an euclidean distance metric instead of alignment based on an evolutionary substitution matrix. AA: amino acid sequences. AA-FLAPs are the same 4776 FLAPs extracted by Soeria-Atmadja *et al.*[15]. Allergens and non-allergens from a database are fragmented into peptides by a sliding window method. The FLAPs and the peptide fragments are then CTD-encoded. In this CTD-representation the FLAPs are aligned against the peptides of each protein sequence, with alignment based on euclidean distance between FLAPs and fragmented peptides. This gives rise to a pattern of alignment scores for each protein from the database. Using this representation the protein samples of the database are input to a learning system, which selects informative features, corresponding to informative peptide fragments.

CTD-encoding Issues

During implementation of the CTD-encoding algorithm three issues arose that required special handling.

- Undetermined amino acids in the given sequences, abbreviated as 'X'. These amino acids are not being assigned to any of the categories that the amino acids are being divided into. However, they are not totally ignored in the CTD-encoding, since they are included in the calculation of the total length of the amino acid sequence.

- Another issue that can arise is that for a given peptide there are no representatives to be found for a certain category of a property to be encoded. For example, it may be that there are no positively charged amino acids present in a sequence. Omitting such a property in the encoding of a sequence would cause the feature vector to be shorter as compared to completely encoded sequences without missing representatives for a category. The euclidean distance comparison between sequences is then difficult to implement in a way that gives comparable similarity scores. For this reason, such sequences were completely excluded. Many FLAPs were considerably shorter than the peptide fragments since the fragments were generated by a sliding window of length 28, which was the mean FLAP-length. Since many FLAPs are shorter than the peptide fragments it makes them more prone to miss representatives. Therefore mainly FLAPs were subjects for exclusion.

- The sliding window length used to cut the query protein sequences into peptide fragments for comparison against the FLAPs, also requires special attention. A straight forward length to employ would be to let the sliding window length be equal to that of the FLAP against which similarity-score is to be computed. In the FLAP set of 4776 FLAPs there are though peptides of 96 different lengths represented. Due to limitations in computational resources and time, the sliding window length was instead set to the mean length of all 4776 FLAPs, which was found to be 28. Thus, protein sequences were split in peptide fragments of length 28, irrespective of length of the FLAP against which comparison was made. See figure 14 for a histogram of the lengths of the 4776 FLAPs.

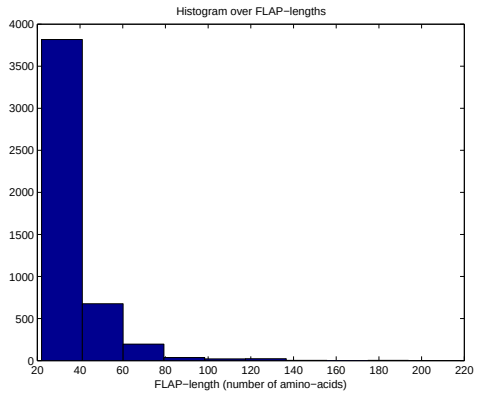


Figure 14: Histogram of FLAP-lengths for the set of 4776 FLAPs. The mean value of 28 amino acids is chosen as the sliding window length.

2.5.5 Overview of Implemented Feature Selection and Representation Methods

A summary of the above described methods for feature selection and feature representation of peptides as well as similarity measures used in this work is found in table 7 below. Two pieces of information, both concerning the similarity measure, and neither of which have yet been described, are included in the table.

First, conventional local alignment algorithms based on dynamical programming allows for adjustment of the alignment with respect to insertions or deletions that may have occurred by evolution of the DNA. For example, if the only difference between two sequences is a minor segment having been inserted into one of the sequences, then a so called gap is introduced by the alignment algorithm, and the alignment score is penalized for the introduced gap. When applying the alignment algorithm based on euclidean distance presented above, this kind of gap-handling is not featured.

Second, substitution frequencies for a substitution matrix such as BLOSUM62 is based on a large set of sequences covering many structural and functional groups. For the purpose of functional classification, it may be though imagined that it might be better to base the substitution frequencies on how common certain substitutions are between two functional classes of interest. Using physico-chemical attributes directly, such as is the case for CTD-encoding, avoids such bias of the substitution frequencies towards the sequence set upon which they are based.

Feature space reduction method	
Feature Extraction	Implemented in DFLAP-algorithm. Requires input of a large set of FLAPs. No reduction to epitopes.
Feature Selection	Reduction of FLAP set to a set of possible epitopes.
Feature representation of sequences	
Pure amino acid sequence representation	Local representation.
CTD-encoding of amino acid sequence using seven physico-chemical amino acid attributes	Global representation. Possible to evaluate if selected physico-chemical attributes are relevant.
CTD-encoding of amino acid sequence using five variables extracted by multidimensional scaling of 237 physico-chemical amino acid attributes	Global representation. The variables subject for encoding are known to be highly informative.
Similarity measure for feature vectors	
BLAST alignment	Gap-handling. Substitution frequencies based on a large set of sequences. BLAST alignment was only performed on pure amino acid representation of sequences.
Euclidean distance	No handling of gaps. Unbiased to any particular sequence set. Euclidean distance was used to measure distance between CTD-encoded amino acid sequences.

Table 7: Comparison of implemented feature selection and extraction techniques, of sequence representations and finally of the similarity measures measuring the distance between feature vectors. The feature extraction in the DFLAP algorithm is compared to the selection technique implemented in this work. The three different representations of protein sequences used in this work are compared, pure amino acid representation, CTD-encoding of seven physico-chemical features and CTD-encoding of five multidimensional scaling eigenvectors. Finally the utilized similarity measures are compared, that is, conventional BLAST alignment and Euclidean distance.

2.5.6 Genetic Algorithm

Motivation and Overview of Algorithm

Running feature selection methods in order to reduce the FLAP set of 4776 peptides to a size of up to 200 we learnt that only the relatively simple feature selection search path methods (σ) implemented in PRTools are reasonably expedient, while the more sophisticated methods are too slow, taking several weeks to finish on a 2GHz 64-bit AMD processor. More specifically, only individual and greedy forward feature selection are fast enough. Individual feature selection evaluates each feature by itself and then ranks them all, while forward selection builds up the feature set by repeated addition of one feature at a time selected to maximize increase of the information content[5]. A drawback is that none of these two methods starts out with investigating a large feature subset. The forward selection method may take an early decision in its search path that is non-optimal in the search of finding an optimal subset of a certain final size. The individual feature selection is even more simple than forward selection since it does not consider existence of correlations between features.

To alleviate this it was decided to apply a genetic algorithm as a feature selection method. One advantage with a genetic algorithm is that it can start out with a large feature subset. Moreover, it is not as computationally intensive as the more sophisticated feature selection search path methods implemented in PRTools. Additionally, a genetic algorithm may be able to find a global optimum among several local optima, which also is shown on benchmarking functions below.

The basic idea behind the genetic algorithm is to solve an optimization problem by simulating evolution. The implemented version is similar to the basic genetic algorithm presented by Goldberg[17]. This algorithm may be summarized as follows, and is illustrated in figure 15.

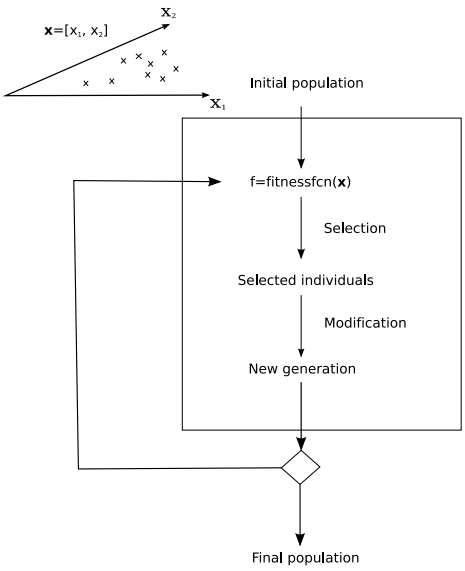


Figure 15: Genetic algorithm overview. A population of individuals is evolved to find an optimum on the fitness surface. In this work the individuals represents feature subsets, so that an optimal individual found corresponds to an optimal feature subset.

The goal of the genetic algorithm is to find the global optimum of an objective function. The initial step involves the creation of a population where each individual is randomly placed in the feature space, that is, the domain of the objective function. The objective score of each individual is then computed by the objective function that takes the feature vector of an individual as input. The objective scores is then transformed into fitness values, by the application of a fitness scaling function. To move the population towards the global optimum of the objective function a selection of fit individuals is performed, by the help of a selection function. The selection function assigns a higher probability of selection to individuals with higher fitness. The set of fitness-selected individuals are the parents allowed to contribute with genetic material to the next generation. From the parent set a new population is created by the application of genetic operators. Their purpose is to introduce diversity in the population, so that it doesn't get stuck in a local optimum. The genetic operators used are replication, mutation and cross-over. By applying the genetic operators to the parents a new generation emerges. It is set to have the same number of individuals as the parent generation as to keep the population size constant. This process is iterated until a stop criterion is reached, such as a maximal number of generations.

Some of the genetic algorithm parameters employed are given a more detailed explanation below.

Rank is the only fitness scaling function applied in this work. It first ranks individuals according to their objective scores. The fitness of an individual with rank n is then set to $\frac{1}{\sqrt{n}}$. The reason for using fitness scaling of raw objective function values is to smoothen the difference between probabilities of being chosen to contribute to the next generation. A sharp decline in probability for lesser fit individuals would lessen the chance of introducing diversity into the next generation and may therefore cause premature convergence where the population settles at a local optimum.

The application of the genetic operators is done in such a way that a parent is subject to only one of them, so that the children are either created by replication of a parent, by mutation of a parent or by cross-over between two parents. The proportions among these is set by two parameters, $nElite$ and Pc. The parameter $nElite$ sets the number of parents that are to be replicated. The selection of the parents subject for replication is done by ranking all parents according to fitness and choosing the $nElite$ top-ranked ones. Among the remaining parents the variable Pc sets the probability that a parent is chosen for cross-over. Parents neither selected for replication nor for cross-over are mutated. The mutational impact applied to each parent selected for mutation is set by a variable denoted Pm. It sets the probability that a bit along the bitstring is flipped.

Test of Implementation on Benchmarking Objective Functions

Since no suitable implementation of a genetic algorithm that would be appropriate to integrate into PRTools within a reasonable amount of time was found, it was decided to make my own implementation. Mathworks Inc. does have a genetic algorithm toolbox for Matlab, but Uppsala University did not have a license for it at the time.

To test the implementation of the genetic algorithm (GA) and to get a rough

idea of appropriate parameter value ranges where an optimal parameter setting may be found, the implemented GA was tested on two objective functions commonly used for benchmarking optimization search tools. These two objective functions, Rastrigin's function and Schwefel's function are defined below, and are also presented in figure 16. They both use a domain of real values. In the implementation, each real is represented by a bitstring of 60 bits.

- Rastrigin's function. It is a spherical model with added cosines. It contains many local optima.

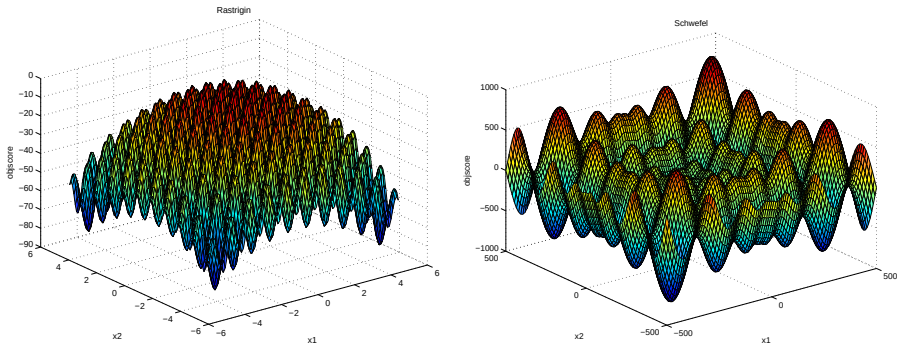
$$f(x) = -(10n + \sum_{i=1}^{n} x_i^2 - 10\cos 2\pi x), \quad -5.12 \leq x_i \leq 5.12 \quad \forall x_i$$

$$\max f(x) = 0, \quad x_i = 0 \quad \forall x_i$$

- Schwefel's function. In this function the second best optimum is not close to the global optimum. It contains many local optima.

$$f(x) = -(\sum_{i=1}^{n} -x_i \sin\sqrt{|x_i|}), \quad -500 \leq x_i \leq 500 \quad \forall x_i$$

$$\max f(x) = -n418.9829, \quad x_i = 420.9687 \quad \forall x_i$$



(a) Surfplot of Rastrigin's function. (b) Surfplot of Schwefel's function

Figure 16: Surfplots of benchmarking objective functions.

Application to Feature Selection

To apply the genetic algorithm as a feature selection method each individual feature subset candidate is represented by a binary vector where each bit represents one feature. The two states of the bit corresponds to whether the feature is selected as being part of the reduced feature subset or not. For example, the bitstring 10100 can represent five features where feature one and three is being selected. Each individual of the population in this way represents a candidate feature subset. The objective function is the information criterion specified by the parameter I, see section 2.1 about composed pattern recognition systems above. In this study a range of different classifiers were used to evaluate the selected features, entailing a so called wrapper selection since classifiers were used as information criteria.

2.5.7 Experimental Setup

Suboptimal model selection: Semi-finals before final competition
It would be preferable to inspect a wide range of different values for each input parameter defining a learning system, and try all possible combinations of these values. Considering the relatively large amount of input parameters to combine as to specify a learning system, see section 2.1 about composed pattern recognition systems above, it is clear that the number of learning systems to evaluate would amount to very large numbers if all possible parameter settings in the parameter space were to be evaluated. Due to limitations in computational resources a search for the optimal model in such a large space of parameters settings could not be done in one run, letting all models compete against all. Instead, and to get an indication of the most promising regions of the parameter space, the set of parameter settings investigated was split into separate runs of the model selection procedure. For each such subset of parameter settings an optimal learning system was selected to qualify to the final. In the final competetion, these "suboptimal" models were assessed on a holdout test using data that had been held away from the semi-finals. The final model that prevailed in the competition between the various suboptimal models, was eventually evaluated by a validation data set kept away from all model selection procedures.

Data splitting
The available data was split in the following way. First the 762 allergens available in the "in-house" database at the *Swedish National Food Administration* was split into a training set of 500 allergens and a validation set of 262 allergens, thereby equalling the data sets described in an article by Soeria-Atmadja *et al.*[15]. To the training set of 500 allergens was added the same 1000 non-allergens as used by Soeria-Atmadja *et al.*. This set of 500 allergens and 1000 non-allergens was further split into an inner training set of 400 allergens and 900 non-allergens used for the suboptimal model selection and an outer set of 100 allergens and 100 non-allergens used for selection of the final model among the suboptimal models.

Specification of model parameter space
A large range of different feature selection parameters as well as a wide range of classifiers were evaluated in the model selection procedure. Investigated values of the input parameters are listed in tables 8, 9 and 10 below, and almost all possible parameter combinations of these values were evaluated.

In the case of using the genetic algorithm as feature selection search path, only classifiers that proved to perform well on the 100 allergens plus 100 non-allergens test set when using the other feature selection methods were evaluated.

Regarding the list of classifiers, a very limited number of learning systems containing a support vector machine or a artificial neural network were also used, but they were prohibitively slow to train and those completed showed poor performance.

Concerning the preprocessing mainly correlation with outcome was used, but a number of other feature selection methods were also tried as preprocessors in conjunction with forward feature selection with $kNN(3)$ as wrapper but none of these returned better performance than correlation with outcome.

An explanation of all classifiers and all parameters of the feature selection

methods employed is outside the scope of this thesis, and the interested reader is encouraged to read the references[5][1].

Concerning the number of features to reduce to a maximum of 200 features was set. The reason for this is that the number of allergens available for training was only 400 and the features (FLAPs) are all derived from these 400 training allergens. To be able to extract FLAPs that had some general allergen-motif 200 peptides therefore seemed to be a reasonable limit.

Most inner parameters of the genetic algorithm was optimized on completely different data sets namely on Rastrigin's function and Schwefel's function. These parameter settings was though found to perform well also on the allergology dataset, so no major optimization of the genetic algorithm parameters was done on this data. Mainly two parameters that were tuned in the application on the allergology dataset, the number of generations until stall and the number of individuals in the population, while the remaining parameters were kept fixed. Parameter values of the genetic algorithm are listed in table 11 below. A detailed explanation of all genetic algorithm parameters is beyond the scope of this thesis, the interested reader is encouraged to consult the litterature[17].

Experimental Setup: No Pre-processing and No Genetic Algorithm	
Parameter	**Value**
D	1300 protein sequences. Classes: 400 allergens, 900 non-allergens. 3 different representations: 1. blosum62, 2. CTD-7 3. CTD-ZZ
n	{10, 50, 100, 150, 200}
σ	Individual FS, Forward FS
I	Euclidean distance, Correlation with outcome, kNN(1), kNN(3), kNN(5)
θ_ϕ	apparent, crossvalidation(5-fold)
ψ	derivedFLAPs FS
ω	fisher, nmc, lda, qda, kNN(1), kNN(3), parzendc
θ_Ω	crossvalidation(5-fold)
$Subopt.Sel.$	100 allergens and 100 non-allergens
V	262 allergens, 1000 non-allergens

Table 8: Parameter settings of learning systems for allergenicity classification of protein sequences where no preprocessing was done. blosum62: pure amino acid sequence representation with BLAST alignment, CTD-7: ctd-encoded amino acid sequences using seven physico-chemical attributes, CTD-ZZ: ctd-encoded amino acid sequences using five multidimensional scaling latent variables, apparent: test based on training data, derivedFLAPs: selects FLAPs derived from the allergens of the training set, nmc: nearest mean classifier, lda: linear discriminant analysis, qda: quadratic discriminant analysis, kNN: k nearest neighbor, parzendc: parzen density classifier. Subopt.Sel: Test data to assess performance of suboptimal models that were retrieved from each separate model selection run.

Experimental Setup: Pre-processing to ∼450 features	
Parameter	**Value**
D	1300 protein sequences. Classes: 400 allergens, 900 non-allergens. 3 different representations: 1. blosum62, 2. CTD-7 3. CTD-ZZ
n	{10, 50, 100, 150, 200}
σ	Individual FS, Forward FS
I	kNN(1), kNN(3), parzendc, decisiontree('maxcrit', 3), decisiontree('maxcrit', 10), decisiontree('fishcrit', 3)
θ_ϕ	crossvalidation(5-fold)
ψ	derivedFLAPs FS * Correlation_with_outcome FS(Coefficient threshold set as to reduce the set to approximately 450 features. blos62: c=0.20, ctd-7: c=0.14 ctd-zz: c=0.175)
ω	kNN(1), kNN(3), parzendc, decisiontree('maxcrit', 3), decisiontree('maxcrit', 10), decisiontree('fishcrit', 3)
θ_Ω	- (all wrappers)
$Subopt.Sel.$	100 allergens and 100 non-allergens
V	262 allergens, 1000 non-allergens

Table 9: Parameter settings of learning systems for allergenicity classification of protein sequences, where preprocessing to around 450 features was performed. blosum62: pure amino acid sequence representation with BLAST alignment, CTD-7: ctd-encoded amino acid sequences using seven physico-chemical attributes, CTD-ZZ: ctd-encoded amino acid sequences using five multidimensional scaling latent variables. 'maxcrit' and 'fishcrit' specifies pruning parameters of the decision tree.

Experimental Setup: Genetic Algorithm as Feature Selection Search Path	
Parameter	**Value**
D	1300 protein sequences. Classes: 400 allergens, 900 non-allergens. 3 different representations: 1. blosum62, 2. CTD-7 3. CTD-ZZ
n	{25, 75, 100, 150, 200}
σ	Genetic Algorithm
I	fisher, lda, qda, kNN(1), kNN(3), kNN(5), parzendc
θ_ϕ	crossvalidation(5-fold)
ψ	derivedFLAPs FS
ω	fisher, lda, qda, kNN(1), kNN(3), kNN(5), parzendc
θ_Ω	- (all wrappers)
$Subopt.Sel.$	100 allergens and 100 non-allergens
V	262 allergens, 1000 non-allergens

Table 10: Parameter settings of learning systems for allergenicity classification of protein sequences, where a genetic algorithm was used for feature selection. blosum62: pure amino acid sequence representation with BLAST alignment, CTD-7: ctd-encoded amino acid sequences using seven physico-chemical attributes, CTD-ZZ: ctd-encoded amino acid sequences using five multidimensional scaling latent variables.

Experimental Setup: Genetic Algorithm Parameters	
Parameter	**Value**
Number of generations (ngen)	100-1000
Population size	30-200 individuals
Number of elite parents	2
Probability for crossover selection	0.6
Probability of mutation along bitstring	0.05
Fitness scaling function	Rank
Selection function	Roulette-wheel selection
Crossover function	Single point crossover
Mutation function	Uniform mutation

Table 11: Parameter settings of the genetic algorithm.

3 Results

3.1 Model Selection and Validation Algorithm: Test of Implementation on Simulated Mixed Gaussian Data

For both the smaller dataset of 20 samples from each of the two classes, as well as for the larger dataset of 200 samples from each of the two classes, individual feature selection was chosen ten times of ten in the 10-fold external cross validation. In ten times of ten feature number one and two was chosen. This is reasonable since the two Gaussian distributions are most separated when projected onto the plane of feature one and two. Regarding selection of classifier of the learning systems, it varies depending on the size of the dataset and also across repeated runs, where new data is drawn from the distributions in each run. For the larger dataset the quadratic discrimant analysis (qda) classifier seems to be selected most often. This is expected since the quadratic discriminant function is based on modeling distributions as normal distributions, which the underlying distributions in this case really are. For the smaller dataset all four classifiers were selected with none of them having an outstanding frequency of selection. The model selection criterion in terms of the prior-weighed external cross validation error (see section 1.4) is estimated to be approximately 0.2.

As a final model individual feature selection and "*qda*" were chosen. Training this model on 100 samples from each class, and making a validation test on 200 fresh samples also gives a performance of approximately 0.2. See table 12 for a summary of the selected model and its performance.

Parameters and Performance of Selected Model	
n^*	2
σ	individual feature selection
I	Euclidean distance
ψ	none
ω	qda
θ_Ω	crossvalidation(10-fold)
J	~ 0.2
V	~ 0.2

Table 12: Parameters and performance of selected model for classification of mixed Gaussians. The "correct" model is selected. The two features known to be most informative is consistently selected. The classifier supposed to be best, quadratic discriminant analysis (qda), is chosen most often. Finally, the model selection criterion is seen to be the same as the validation criterion.

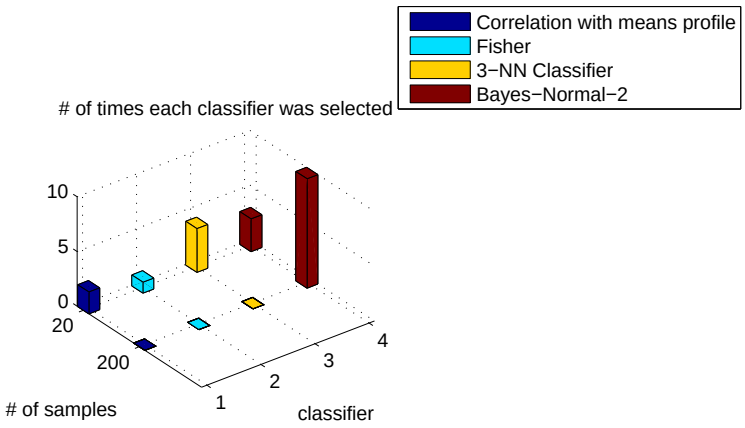


Figure 17: 10-fold external crossvalidation on two simulated Gaussians. 4 classifiers are chosen among. Runs on two datasets of different sizes were performed. For the larger dataset qda is seen to be chosen most often. For the smaller dataset no classifier is consistently selected. "*Bayes-Normal-2*" is identical to quadratic discriminant analysis (qda), "*Correlation with means profile*" is identical to the "*distance to mean classifier*" described above, see 2.1.

3.2 Application to Breast Tumour Classification

Two approaches were realized, model selection followed by holdout validation and model selection combined with external cross validation.

3.2.1 Holdout Validation

Model selection

In the case of model selection followed by holdout validation, the model selection is only made once and returns top-scoring models comprised of forward feature selection as search path, *inter-intra* distance measure as information criterion and "*Fisher*" as classifier.

The two best performing models selected have 30 and 40 genes respectively, both showing equal performance. The performance is remarkably good, the 10-fold cross validation which the model selection is based upon reported zero errors. The parameters of the selected model along with the value of the model selection criterion J are summarized in table 13 below.

Validation

Validation of the very promising model with 30 genes gave negative results. In the case of holdout validation using the 19 holdout samples from van't Veer *et al.*[48] it misclassifies one out of seven good prognosis patient and eight out of twelve bad prognosis patients. Classification of the 180 patients from van de Vijveer *et al.*[47] results in 66 out of 138 good prognosis patients being misclassified

Model Parameters and Performance of Selected Model for Breast Cancer Data	
Parameter	**Value**
n^*	30
σ	Forward FS
I	Inter-intra distance
θ_ϕ	apparent
ψ	correlation_with_outcome_FS(correlation threshold = 0.23) $*$ expression_change_FS(intensity ratio ≥ 2, $p_{value} < 0.01$, ≥ 3 samples)
ω	fisher
θ_Ω	crossvalidation(10 fold)
J	0
V	See confusion matrices in table 14 above.

Table 13: Parameters and performance of selected model for breast tumour classification. apparent: test is based on training data.

and 16 out of 42 bad prognosis patients being misclassified. This corresponds to prior-weighed errors (see section 1.4) of 0.47 and 0.46 respectively. The results of the holdout validation are summarized in the confusion matrices in table 14 below.

	Good prognosis	Bad prognosis
Good prognosis	44	0
Bad prognosis	0	34

(a) Confusion matrix for selected model on 78 training samples from breast cancer data.

	Good prognosis	Bad prognosis
Good prognosis	6	1
Bad prognosis	8	4

(b) Confusion matrix for selected model on 19 holdout samples.

	Good prognosis	Bad prognosis
Good prognosis	72	66
Bad prognosis	16	26

(c) Confusion matrix for selected model on 180 holdout samples.

Table 14: Confusion matrices showing the holdout validation results for the selected model. Each row represents actual class and each column predicted class. On training data the model is seen to perform exceptionally well, but on validation data the results are less positive.

3.2.2 External Cross-Validation

Model selection

In the case of the external cross validation, model selection was conducted 25 times, and in these 25 competitions forward feature selection, inter-intra distance measure and the "Fisher" classifier prevailed 25 times out of 25. This is the same model as that selected in the case of holdout validation described above.

The size of the feature subsets varied between 20, 30 and 40 genes, being

selected approximately 25, 50 and 25 percent of the times respectively. Unfortunately, as illustrated by figure 18 the composition of these subsets varied. There are approximately 10 genes that were selected in at least 10 of the 25 selected models, but almost all of the 221 genes were selected at least once. However, there is one outstanding peak. Gene number 97 was chosen 25 times out of 25. A litterature study revealed that the official symbol of this gene is *TSPYL5*. Details about this gene are outlined in the discussion. The 10-fold cross-validation which was used as model selection criterion also in this case reported an error of zero.

Validation
A histogram composed of the 25 prior-weighed holdout errors that the external cross validation gives rise to (see section 1.4) was made. This results in a histogram that is not especially narrow and which is centered somewhere around an error of 0.35, see figure 19.

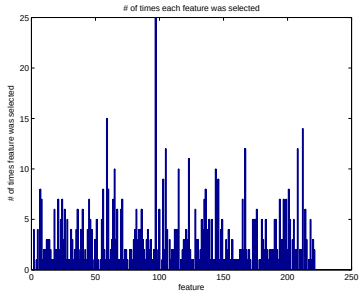


Figure 18: Features used by the 25 selected models in the external cross validation procedure on breast cancer data. The selected feature subsets are seen to be relatively unstable.

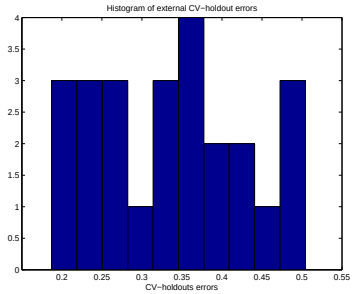
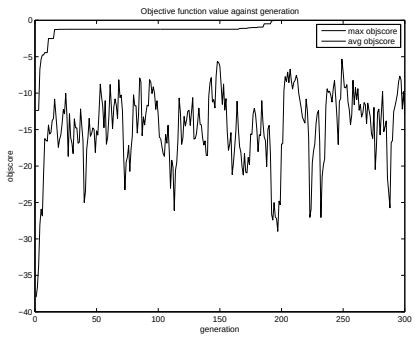


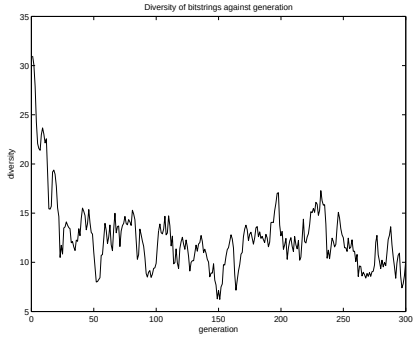
Figure 19: Histogram of 25 crossvalidation-holdout errors from the external cross validation procedure. The error distribution is seen to have relatively high variance.

3.3 Genetic Algorithm: Test of Implementation on Benchmarking Objective Functions

The global optimum was successfully found for both test-functions. The max- and average objective score of the generations can be seen in plots below, as well as the development over the generations in terms of histograms. Interestingly, in the case of Rastrigin's function the population is first settled in a local optimum next to the global one and no individual has yet entered the pit where the global optimum resides. Later on however, some individuals find their way to the global pit causing the majority of the population to move there. In the case of Schwefel's function a similar scenario should be harder, since if the population first settles at the second best optimum it is then relatively far away from the global one. The population was though found to also in this case settle at the global optimum. To conclude, application of the GA on test functions with a real-valued domain indicates that this algorithm seems to behave well.

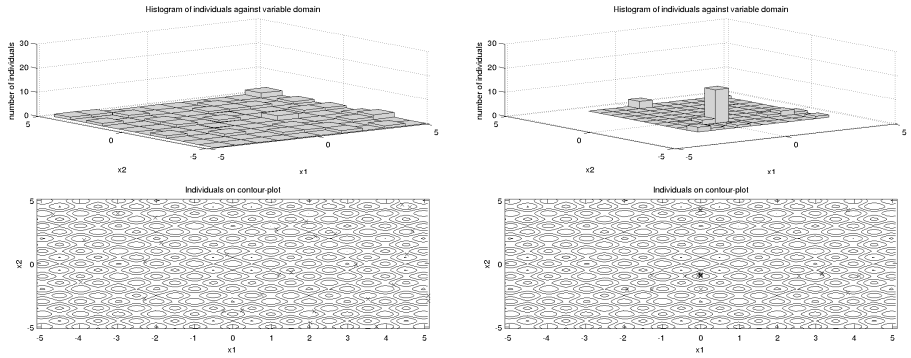


(a) Max and average objective scores of population against generation.



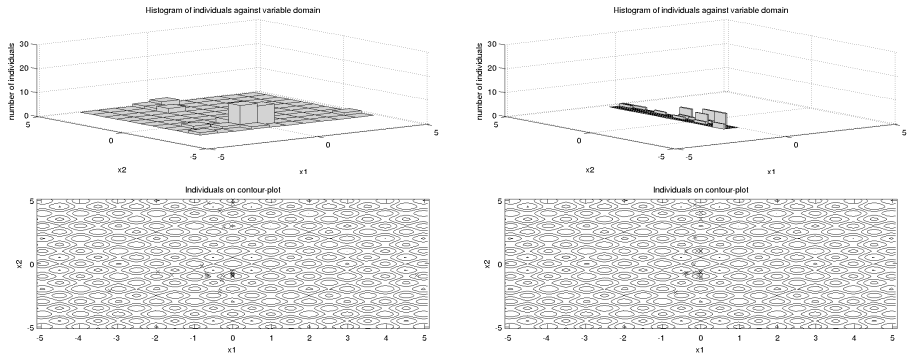
(b) Diversity of bitstrings of population against generation.

Figure 20: Rastrigin's function. The best individual is seen to find the global optimum, having an objective score of zero, after around 200 generations. The GA was set to use 300 generations and a population size of 30 individuals.



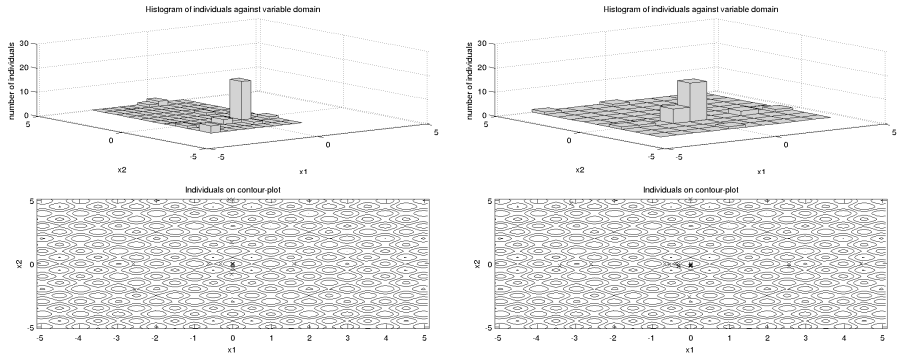
(a) gen=1

(b) gen=60



(c) gen=120

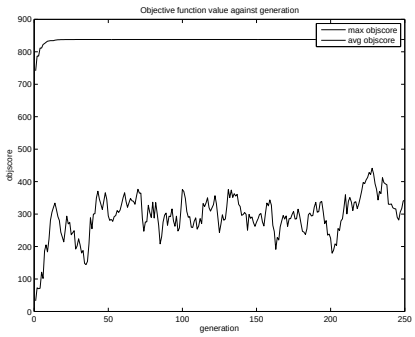
(d) gen=180



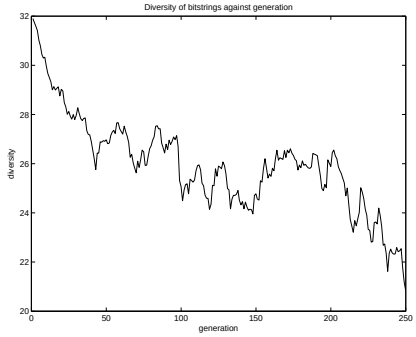
(e) gen=240

(f) gen=300

Figure 21: Rastrigin's function. Snapshots of histogram and scatterplot of individuals of a population for certain generations. The population is seen to evolve as to move to the global optimum at (0,0). The GA used 300 generations and 30 individuals as population size.

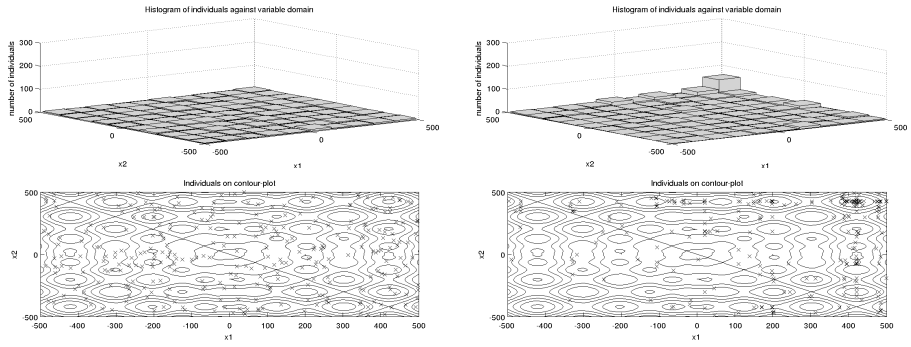


(a) Max and average objective scores of population against generation



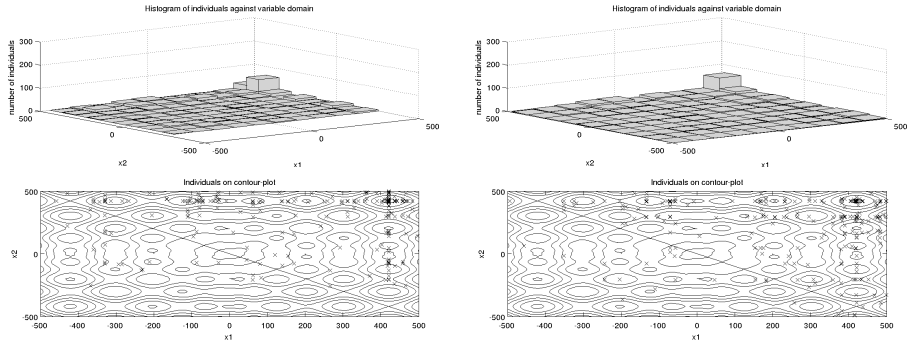
(b) Diversity of bitstrings of population against generation

Figure 22: Schwefel's function. The best individual is seen to find the global optimum after around 25 generations. As seen from the population diversity plot the population are at this stage still highly inhomogeneuous. The inertia of the evolution of the population is desired as to not give premature convergence. The GA was set to use 250 generations and a population size of 30 individuals.



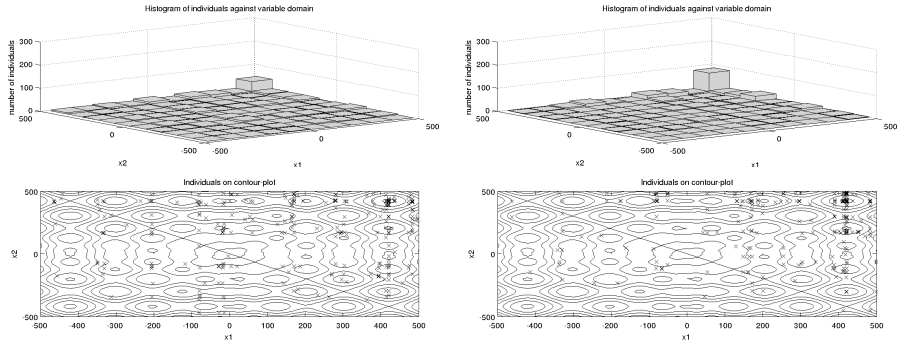
(a) gen=1

(b) gen=50



(c) gen=100

(d) gen=150



(e) gen=200

(f) gen=250

Figure 23: Schwefel's function. Snapshots of histogram and scatterplot of individuals of a population for certain generations. The population is seen to evolve as to move to the global optimum. The GA used 250 generations and 30 individuals as population size.

3.4 Application to Allergen Classification

For each of the three dataset representations, amino acid alignment with blosum 62 substitution matrix, alignment of CTD-encoded sequences based on 7 physico-chemical properties and third, CTD-encoding based on the 5 most informative eigenvectors from multidimensional scaling of physico-chemical properties, the two best performing systems retrieved from running the GA and the best system retrieved from using all other feature selection methods are presented in table 15. With best performing is meant the highest accuracy on the test dataset which is used as a model selection criterion to select among the "suboptimally selected" models. The accuracy is the total number of misclassifications on this test dataset comprised of 100 allergens and 100 non-allergens. Performance of the models are displayed in table 16. Concerning the genetic algorithms the max- and average objective scores of the generations are presented in figures 25 to 30 below and also the average diversity of the bitstrings of the generations.

Composition of selected models
The performance is generally highest when using blosum 62 representation. This might be explained by the fact that a relatively large number of FLAPs were lost due to missing group representatives in the CTD-encoding. More specifically, for CTD-encoding using 7 physico-chemical features only 4365 out of the 4776 FLAPs could be encoded, corresponding to a loss of 9% of the FLAPs. For CTD-encoding using the 5 multidimensional scaling variables it is even worse, only being able to completely encode 3364 FLAPs, corresponding to a loss of 30%.

Regarding the feature selection parameters the genetic algorithm is generally seen to populate as good as all top performing positions, but in blosum 62 representation it is found to be surpassed by an individual feature selection method that gives the very best performance achieved across all parameters. This best performing system uses a one-nearest-neighbor classifier as feature selection criterion, I, and it applies 200 features. Somewhat surprisingly, it uses the parzen density classifier as classifier ω instead of the the one-nearest-neighbor classifier. One would expect that the classifier for which the feature subset is optimized would be chosen as classifier. The reason for this phenomenon of a different classifier being chosen as ω compared to what is used as I is not unlikely overtraining. It may also be that the one-nearest-neighbor and the parzen density classifier have relatively similar decision boundaries on this data. Generally the parzen density classifier appears most often among the top selected models, and the one-nearest-neighbor classifier the second most.

Feature subset size
Regarding the number of features selected it can from table 15 be seen that it is consistently in the neighborhood of 200 features. Notably, the best performing system with GA as feature selection method starts with all individuals having 75 features, but the best individual of the last generation have 205 features, see table 15.

Performance and Validation
The best performing model only gives 13 misclassifications on the 100 allergens

$D:id$	ψ	σ	I	θ_ϕ	ngen	n	n^*	ω	θ_Ω
blos62:1	derived	i	knn-1	app	-	50:50:200	200	parzendc	100A+100NA
blos62:2	derived	GA	knn-1	app	1000	75	205	parzendc	100A+100NA
blos62:3	derived	GA	parzendc	cv(5)	300	200	215	parzendc	100A+100NA
ctd-zz:1	derived	GA	lda	cv(5)	200	200	189	knn-1	100A+100NA
ctd-zz:2	derived	GA	parzendc	cv(5)	200	200	272	parzendc	100A+100NA
ctd-zz:3	derived* corroutc (0.175)	f	parzendc	cv(5)	-	50:50:200	200	parzendc	100A+100NA
ctd-7:1	derived	i	knn-1	cv(5)	-	50:50:200	200	knn-1	100A+100NA
ctd-7:2	derived	GA	knn-3	cv(5)	200	200	232	parzenc	100A+100NA
ctd-7:3	derived	GA	parzendc	cv(5)	200	200	232	knn-3	100A+100NA

Table 15: Parameter values of 3 top ranked models in each representation. id: identifier of model, blos62: representation using pure amino acid sequence and blosum62 substitution matrix as dissimilarity measure, ctd-zz: ctd-encoding based on 5 multidimensional scaling eigenvectors, ctd-7: ctd-encoding based on 7 physico-chemical attributes, derived: selects FLAPs derived from the allergens in the training set, corroutc: correlation with outcome feature selection, i: individual, f: forward, GA: Genetic Algorithm, knn: k nearest neighbor, lda: linear discriminant analysis, parzendc: parzen density classifier, app: apparent test based on training data, cv: crossvalidation, ngen: number of generations of GA, 50:50:200: {50,100,150,150,200}, A:allergens, NA:non-allergens.

and 3 misclassifications on the 100 non-allergens, corresponding to a sensitivity and specificity of 87% and 97% respectively, see table 16. The performance on the validation data is somewhat lower, with sensitiviy and specificity of 82% and 94%. Unfortunately, the model gives 121 misclassifications on the 193 tropomyosins which is one of the "difficult" protein families containing both allergens and non-allergens.

All retrieved classification systems have very poor performance on the difficult dataset. No protein from any of the three difficult protein families was however incorporated into the training dataset of these models. Hypothetically, including some of these into the training set could significantly improve performance on the difficult dataset. Alternatively, performance on the difficult proteins could be included as part of the model selection criterion, which is also the approach adopted by Soeria-Atmadja et al[15]. There the number of misclassifications on four of the non-allergen tropomyosins was part of the model selection criterion, J.

Training including 52 non-allergen tropomyosins

To investigate whether improved performance on the difficult non-allergens could be conferred to the selected model a new training of models were performed where 52 out of 104 tropomyosins were included in the training dataset. The same parameters as those of the three systems with blosum62 representation with the best performance when trained on the previous training set without difficult proteins were applied for this study. The three models were then tested against a set of 26 other tropomyosins making up part the model selection criteria, J, together with the 100 allergens and 100 non-allergens previously used. Finally the models were validated against the same validation set previously used composed of 262 allergens and 1000 non-allergens, as well as the remaining 26 tropomyosins, 39 profilins and 14 parvalbumins. The performance of these models on the 262 allergens and 1000 non-allergens are comparable to the models not having tropomyosins in their training set, see table 17, but notably higher on the difficult proteins, see table 18. All three models classifies all

tropomyosins correctly. The 39 profilins are also relatively well classified with sensitivities of 85%, 87% and 100%. However, the 14 parvalbumins still persist as being problematic. One model correctly assigns 4 of the parvalbumins as non-allergens while the other two models classifies all of them as being allergens.

$D : id$	J_{100A}	J_{100NA}	J_{tot}	J_{sens}	J_{spec}	V_{262A}	V_{1000NA}	V_{sens}	V_{spec}	$V_{193\ diff}$
blos62:1	13	3	16	0.87	0.97	47	56	0.82	0.94	121
blos62:2	18	1	19	0.82	0.99	46	21	0.82	0.98	114
blos62:3	18	3	21	0.82	0.97	44	40	0.83	0.96	110
ctd-zz:4	28	12	40	0.72	0.88	91	144	0.65	0.86	85
ctd-zz:5	41	0	41	0.59	1.00	140	19	0.46	0.98	75
ctd-zz:6	38	5	43	0.62	0.95	139	54	0.46	0.95	77
ctd-7:7	26	16	42	0.74	0.84	95	115	0.64	0.88	127
ctd-7:8	36	8	44	0.64	0.92	104	47	0.60	0.95	106
ctd-7:9	36	9	45	sens	spec	121	51	0.54	0.95	112

Table 16: Value of model selection criteria, J, and validated performance of three top performing learning systems in each representation. The integer values states the number of misclassifications. id: identifier of model, A:allergens, NA:non-allergens, $J_{tot} = J_{100A} + J_{100NA}$, sens: sensitivity, spec: specificity, diff: 193 proteins from "difficult" protein families, *i.e.* 140 tropomyosins, 39 profilins and 14 parvalbumins

$D : id$	J_{100A}	J_{100NA}	J_{tot}	J_{sens}	J_{spec}	V_{262A}	V_{1000NA}	V_{sens}	V_{spec}	$V_{193\ diffNA}$
blos62:1 + 52 Tropo	20	8	28	0.80	0.92	40	101	0.85	0.90	20
blos62:2 + 52 Tropo	19	2	21	0.81	0.98	43	20	0.84	0.98	37
blos62:3 + 52 Tropo	21	1	22	0.79	0.99	49	12	0.81	0.99	23

Table 17: Value of model selection criteria, J, and validated performance of three top performing learning systems in each. These models have been trained with 52 of the tropomyosins included in the training set. The integer values states the number of misclassifications.

$D : id$	$J_{26Tropo}$	$V_{26Tropo}$	V_{39Prof}	V_{14Parv}	$Spec_{26Tropo}$	$Spec_{39Prof}$	$Spec_{14Parv}$
blos62:1 + 52 Tropo	0	0	5	10	1.00	0.87	0.29
blos62:2 + 52 Tropo	0	0	6	14	1.00	0.85	0.00
blos62:3 + 52 Tropo	0	0	0	14	1.00	1.00	0.00

Table 18: Value of model selection criteria, J, and validated performance of three top performing models on non-allergens from "difficult" families. The families are, Tropo: Tropomyosins, Prof: Profilins and Parv: Parvalbumins. The models are the same as above, see table 15, except that the training dataset included 52 non-allergen tropomyosins. The integer values states the number of misclassifications.

Overfitting of GA

In the objective score plots, the maximum and average classification performances of a population of feature subsets are plotted against generation, see figures 25 to 30. From these plots is evident that the model selection on the training data of 400 allergens and 900 non-allergens performed by the GA results in overfitted models. Looking for example at figure 25 which corresponds

to blosum 62 representation and parzen density classifier, cross validation performance estimates of over 0.97 for the best system of the generation is seen and rather stabilized averages of around 0.95. However, the performance estimates on the holdout of 100 allergens and 100 non-allergens used for selection among the suboptimal models, only give an accuracy of around 0.9, see table 16. The same phenomenon is observed for the CTD representations where the systems have higher performance estimates within the GA optimization, but performs significantly worse on the model selection holdout.

Robustness of feature subset composition

Robustness of subset compositions was evaluated by means of two procedures. First, by checking the number of peptides in the intersection between the subsets of the top scoring models, and second, by in the same way checking the overlap of subsets returned from repeated runs of the top scoring model having GA as feature selection search path. The subsets all have sizes of approximately 200 features, but the intersections of pairs of these subsets only contain around 3 features as illustrated in figure 24.

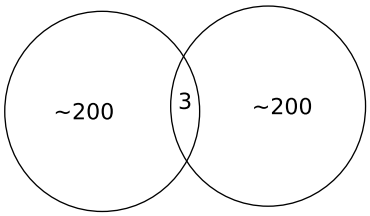
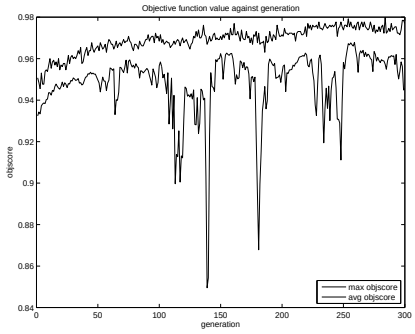
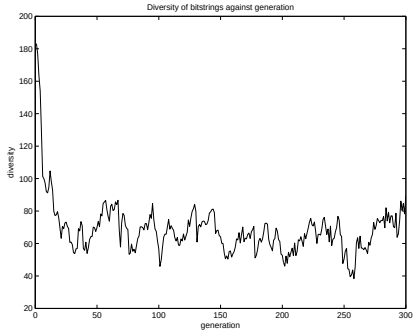


Figure 24: Overlap between feature sets of top-ranked models for allergenicity prediction.

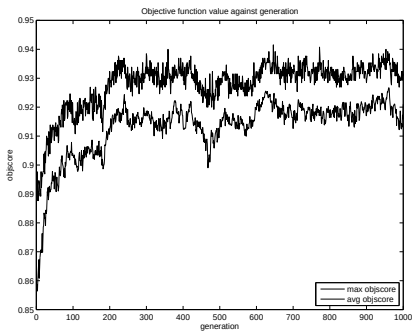


(a) Max and average objective scores of population against generation.

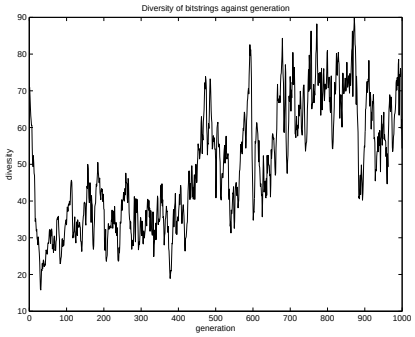


(b) Diversity of bitstrings of population against generation.

Figure 25: D: Blosum 62. I: parzendc. The performance is the highest achieved among models employing the genetic algorithm. The best feature subset reaches a value of 98% of the "suboptimal" model selection criterion, J. The GA used 300 generations and a population size of 30 individuals.

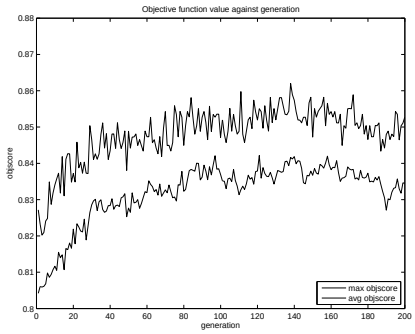


(a) Max and average objective scores of population against generation.

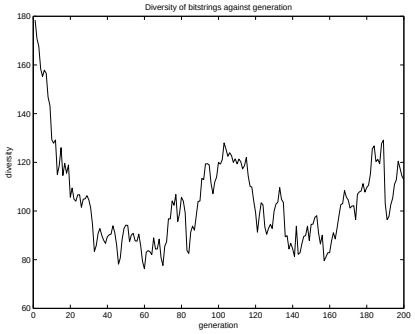


(b) Diversity of bitstrings of population against generation.

Figure 26: D: Blosum 62. I: knn-1. 1000 generations, 30 individuals

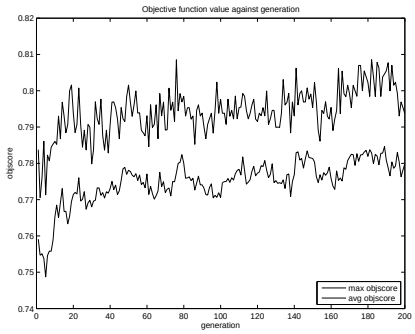


(a) Max and average objective scores of population against generation.

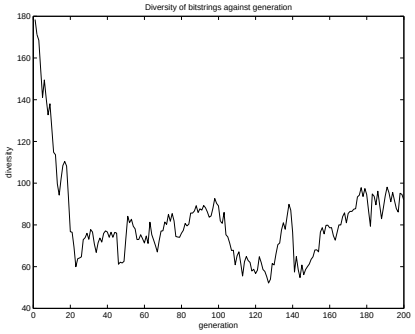


(b) Diversity of bitstrings of population against generation.

Figure 27: D: CTD-ZZ. I: parzendc. 200 generations, 30 individuals

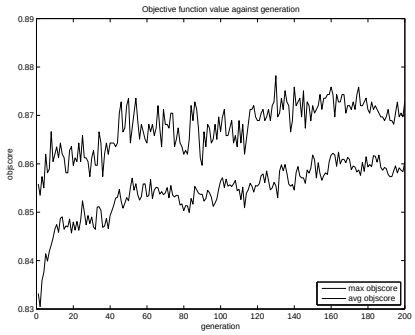


(a) Max and average objective scores of population against generation.

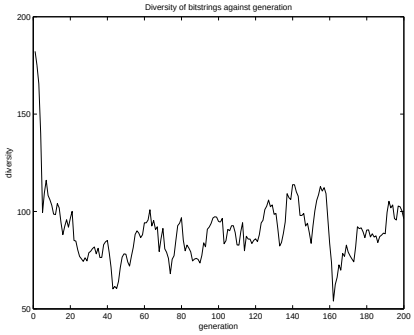


(b) Diversity of bitstrings of population against generation.

Figure 28: D: CTD-ZZ. I: linear discriminant analysis. 200 generations, 30 individuals

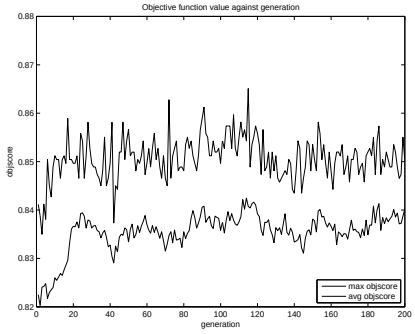


(a) Max and average objective scores of population against generation.

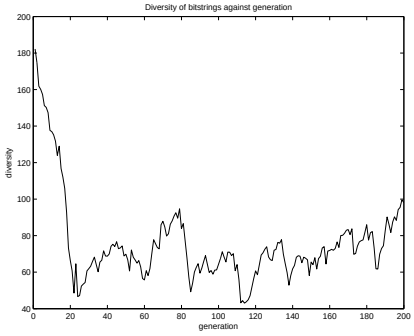


(b) Diversity of bitstrings of population against generation.

Figure 29: D: CTD-7 physchem. I: parzendc. 200 generations, 30 individuals



(a) Max and average objective scores of population against generation.



(b) Diversity of bitstrings of population against generation.

Figure 30: D: CTD-7 physchem. I: knn-3. 200 generations, 30 individuals

4 Discussion

4.1 Model Selection and Composed Pattern Recognition Systems

This work implements the viewpoint that a learning system is comprised of several components such as preprocessing, feature selection and a classifier, each component itself having a number of parameters. Such a viewpoint facilitates a more properly performed model selection and validation. Automation of model selection across all learning system parameters considerably facilitates a search across wide ranges of different learning systems, and thereby lessens the need to beforehand select a single learning system to apply for a given problem. The relative ease with which one can search for optimal learning systems is also illustrated in this study by the application of the implementation to two rather complex real world problems, breast tumour classification using microarray data and functional classification of proteins based on their amino acid sequences.

The implementation of automated model selection in conjunction with validation techniques was successful. This is illustrated by the application of the algorithm on simulated Gaussian data, where learning systems that ought to perform best on such data also are selected by the algorithm.

4.2 Application to Breast Tumour Classification

Performance
The application of the implemented automated model selection and validation on the breast tumour classification problem clearly points out the need for proper validation of the selected model. This is evident from a model found which classifies all patients in the model selection holdout test data correctly, but that same model performs significantly worse than van't Veers 70-gene classifier on validation data held out from the model selection procedure, see table 19 and 14. The reason to this is that with a large learning system parameter space it is highly probably that one finds a model that performs very well on the model selection data, thereby resulting in an overfitted model on data available for model selection.

Selected feature selection method: Univariate vs Multivariate
It is interesting to note that forward feature selection was the method of choice among the various feature selection methods, σ, tested. This is a multivariate feature selection method whereas that of van't Veer's work[48] is univariate. Since the selected model in this study was overfitted, it is not unlikely that univariate feature selection may perform better on holdout data. Higher performance of univariate feature selection methods, relative to that of multivariate methods on microarray cancer datasets is also corroborated by a study by Lai *et al.* in 2006[9]. In their conclusions they state that correlation structures, if present, are difficult to extract due to the small number of samples, and that consequently, overly-complex gene selection algorithms that attempt to extract these structures are prone to overtraining. As a counterargument one may though claim that their conclusions may be due to limitations in the methods applied. The search methods they use investigate a limited search path in the large set of possible feature combinations, and the application of a method such

	Good prognosis	Bad prognosis
Good prognosis	36	8
Bad prognosis	5	29

(a) Confusion matrix for van't Veer 70-gene predictor on 78 training samples.

	Good prognosis	Bad prognosis
Good prognosis	6	1
Bad prognosis	1	11

(b) Confusion matrix for van't Veer 70-gene predictor on 19 holdout samples.

	Good prognosis	Bad prognosis
Good prognosis	73	65
Bad prognosis	3	39

(c) Confusion matrix for van't Veer 70-gene predictor on 180 holdout samples.

Table 19: Confusion matrices showing the holdout validation results for van't Veer's 70 gene predictor. Each row represents actual class and each column predicted class. The performance on the 180 sample holdout is considerably worse than previously reported.

as a genetic algorithm may have produced good results.

Selected classifier
The classifier used by van't Veer *et al.* was never selected in the computer experiments of this study. This gives rise to questions as to whether there are better performing classifiers than that of their subjective choice. With modification of the model selection procedure as to select systems that are less overfitted, results presented here indicate that it may be possible to find a model that have better performance.

Data comments
The van de Vijveer data published in *New England Journal of Medicine*[47] need some additional comment. It should be stated that it was difficult to select patients fulfilling the same class criteria as those used by van't Veer *et al.*. It may even be questioned whether metastasis within 5 years is a reasonable target, or if it would have been more appropriate to use cell biological markers such as histological data or estrogen receptor levels. The micro-array data from van't Veer is also somewhat different since it is based on two-channel microarrays whereas data studied by van de Vijveer *et al.* is produced by one-channel microarrays. The data used in this study was downloaded in an already normalized format, interpretations should be made with caution since it is unclear if the normalizations are done such that the datasets are directly comparable.

Gene-list robustness and *TSPYL5*
As is clear from external cross-validation the composition of gene subsets is not robust, since there are few genes that are frequently selected to be part of the "optimal" gene subset, see figure 18 on page 44. This is a well-known phenomenon in microarray datasets[42][32]. A recent study that employ the 70-gene predictor and also other models reported to perform well on the cancer microarray domain, finds that although the gene-list is not stable, the perfor-

mance is found to be stable[8]. This is encouraging for their use as predictors although it can't enable interpretations of the genes in the gene-list.

As an exception, one gene in my study, *i.e.* number 97 among the 221 genes shown in figure 18, repeatedly occurs in the gene list of models selected in the external cross validation procedure. This gene is named TSPY-like 5 (*TSPYL5*). Interestingly, a search in the litterature finds numerous recent articles where this gene is a member of a list of top-ranked genes extracted from expression data aimed to predict cancer status. Alexe *et al.*[20] investigates the same van't Veer microarray dataset under study in this thesis and *TSPYL5* is found to be ranked as number 8 in a list of 17 genes. In an article by Niijima and Kuhara it is ranked as number 2 in a list of 10 top-ranked genes[43]. In an article by Vachani *et al.*[4] it appears in a panel of 10 genes, used to distinguish whether a lung tumour is primary or a case of metastasis of head and neck carcinoma. The Gene Ontology (GO) states that *TSPYL5* is located in the nucleus and concerning the biological process it is said to be involved in the aggregation and bonding of the nucleosome. It is therefore hypothetically a critical factor for epigenetic control and can thereby be controlling the transcription of many genes. This is found to be corroborated in a study by Kim *et al.*[46], aiming to find targets of epigenetic silencing in malignant glioma, which is the most common central nervous system tumour of adults. The study identifies *TSPYL5* as one of a handful of genes that are epigenetically silenced in glioma, and that the expression of *TSPYL5* suppress growth of glioma in cell culture. These findings indicate that *TSPYL5* might be part of a core set of oncogenes, since it is found to be involved in several kinds of carcinoma.

The genes that were found to be selected more than ten times of the twentyfive external cross validation holdouts are naturally also interesting subjects for further investigation. Inspired by the success of finding *TSPYL5*, it could be of interest to create a filter that selects genes that are repeatedly selected in the retrieved subsets. Related ideas of robustifying the selected feature subsets by looking at a population of models have been summarized elsewhere[25].

4.3 Application to Allergen Classification

Performance

The results from the study of allergen classification clearly indicate that a considerable reduction of the peptide feature subset can be done, still withholding high classification performance. It is also clear that as to achieve high performance on proteins from protein families containing both allergens and non-allergens at least some part of it need to be taken into account in the training phase. However, the parvalbumins still pose a problem, since a clear majority of them are incorrectly classified even though models were trained on tropomyosins which are also a protein family containing both allergens and non-allergens. This indicates that the retrieved models do not have the capability to generalize to "novel" protein families outside the training domain that contain both allergens and non-allergens.

Selected classifier

Regarding the composition of the top-scoring models it is interesting to note that the one nearest neighbor and the parzen density classifiers frequently are selected. One may speculate that this could be due to that the sequences form

many separate groups in the feature space and these groups of allergens and groups of non-allergens are inhomogeneously mixed with one another. Such an explanation could reflect that there is no single linear epitope allergen-motif, but instead one motif for each allergen group.

Feature selection method: GA vs others
Regarding the feature selection component of the selected models it is clear that a non-linear information criterion, I, in the form of a wrapper is needed. Concerning the feature selection search method, σ, it is not obvious which one is best. The very best model utilizes individual feature selection, but most of the selected models utilize a genetic algorithm.

Overfitting of feature selection
It is interesting to note that the classifier used as information criterion, I, in the feature selection, often differs from the classifier component, ω, of the selected models. A plausible explanation to this is overtraining in the feature selection step. Such an explanation is also supported by the results of this study that clearly show an overfitting of the GA in the feature selection step.

Representations
Concerning the representations it is clear that similarity based on alignment score using blosum 62 substitution matrix gives best classification. Two possible reasons to this is first the lack of gap-handling in the alignment algorithms not utilizing the dynamic programming algorithm and second, a quite considerable loss of FLAPs when using the CTD-encoding. Many FLAPs are lost due to the second CTD-encoding issue mentioned above, *i.e.* missing amino acid representatives for certain physico-chemical properties. Hypothetically, the information contained in the CTD-encoding representation should otherwise cover the information in the substitution matrices and extend it by its more global representation. This statement is corroborated by independent studies by Venkatarajan *et al.*[45] and Tomii and Kanehisa[30] that both report high correlation between their selected physico-chemical features and the BLOSUM and PAM substitution matrices.

Feature subset size
The selected feature subset is seen to have around 200 features across all representations of the dataset and across all compositions of the learning system. This is illustrated by the second best model which uses a genetic algorithm that starts out with all individuals having 75 features but ends up with a top scoring individual of 205 features.

Feature subset robustness
The small overlap between the feature subsets may be explained as follows: Assume that there are two very close homologues of each allergen within the given dataset of 400 allergens. Then selecting 200 peptides is enough to retrieve unique peptides that are specific for each homology-pair. Extracting epitopes that are specific for each homologous allergen-pair, it would then not matter which peptide for each such pair we choose except that such a peptide is not allowed to be highly homologous to any of the non-allergens in the test set. Since low homology to non-allergens was the very criterion for constructing the

set of FLAPs it is unlikely that any of the 4776 peptides selected among are highly homologuous to any non-allergen. The result is that the selected feature subset can be highly variable. This hypothesis would also explain why we need approximately 200 features since the number of peptides would reflect the extent of homology within the allergen set. It is also in line with the fact that the one-nearest-neighbor classifier and the parzen density classifier perform well. This is expected in a feature space with many separate groups of allergens with high homology within each allergen group.

Trying to classify a family of proteins that is homologous but contains both allergens and non-allergens would then give poor performance, since the classification would not be based on the signature of the whole spectrum of allergen-peptide alignment-scores but rather on similarity to a few specific peptides. This may be the reason to the poor results for classification of the "difficult dataset". The 200 peptides do not represent a general allergen-motif but rather 200 unique allergen sequences. It would be interesting to attempt circumventing this problem by an initial homology filtering of the dataset. Such a homology filtered dataset may then be split into a training and test set. It might then be possible to extract a pattern of peptides that is not unique to specific allergens.

To investigate whether the selected peptides are relevant epitopes, one may also compare the retrieved subsets against epitope databases.

As a final note on this topic, as mentioned in the discussion of the microarray predictors the difficulty of achieving robust feature lists in high dimensional feature spaces is well-established, but the retrieved predictor found in this study may still be useful in terms of its predictive power.

Performance comparison

In table 20 below the performance of the selected predictor is compared against two of the highest performing predictors found in the literature concerning classification of allergenicity of protein sequences. The sensitivity and specificity is seen to be comparable for the predictor found in this study. However, the predictor presented in this work only uses 200 peptide fragments (FLAPs) for its classification as compared with Soeria-Atmadja *et al.* who use 4776 FLAPs. The approach by Cui *et al.* does not have the possibility to identify possible epitopes, but instead use the whole protein sequence of a query protein as input to a classifier. The bad specificity on the "difficult" protein family of parvalbumins is unfortunate, but might be due to the fact that optimization of the model parameters was not done as thoroughly when the "difficult" protein family of tropomyosins were included in the training data as when it was not. Doing that could improve the predictive power, making it competitive with that of Soeria-Atmadja *et al.* Notably, in their design they did also use samples from the tropomyosins in the training phase, similarly to this study.

Study	Number of FLAPs	Sensitivity	Specificity	$Sens_{Tropo}$	$Sens_{Prof}$	$Sens_{Parv}$
Stadler	-	86%	95%	-	-	-
Cui	-	93%	99%	-	-	-
Soeria-Atmadja	4776	87%	98%	95%	82%	96%
Edsgärd	200	80%	92%	100%	87%	26%

Table 20: Performance comparison of top performing allergenicity predictors found in litterature. "Difficult" protein families validated against are tropomyosins, profilins and parvalbumins. Sens: sensitivity.

4.4 Summary of Conducted Work

The conducted work may be divided into work on a toolbox which may be reused in other applications, and the application of the toolbox for analysis of microarray and protein sequence data.

- **Pattern Recognition Toolbox**
  - Implementation of algorithms for automated model selection and validation
  - Implementation of a Genetic Algorithm
  - Implemenentation of minor methods for the applications, *e.g.* filter genes with at least a two-fold change in expression level.
- **Applications**
  Microarray data
  - Classification of prognosis for breast-cancer patients.

  Protein sequence data
  - Implementation of CTD-encoding.
  - Classification of proteins to functional class: Allergen or Non-Allergen

4.5 Conclusions

The conclusions listed below answers whether the aims of this thesis, listed on page 13, have been fulfilled and summarizes the findings of this study.

- General
  - The implementation of the algorithm that automates model selection and validation was successful. Its correctness was illustrated on simulated datasets of normal distributions. Its usefulness was illustrated by the application on two real world problems under study in this thesis.
  - Proper validation of selected models is critical as to not report biased performance estimates. The reason is that promising models have a tendency of overfitting to data when having been selected from a large range of models.
  - The composition of retrieved feature lists is unstable in both of the applications.

- Application to Breast Tumour Classification
  - Model selection indicates that predictors performing better than van't Veer's 70 gene predictor should be possible to construct.
  - The selected models also indicate that predictors using fewer genes, down to as few as 30, can potentially reach levels of comparable performance. However, the composition of the retrieved subsets is not robust with respect to perturbations of the training data.
  - *TSPYL5* is identified as a putative oncogene. This finding is corroborated by the litterature.
- Application to Allergen Classification
  - High performing predictors can be constructed with a considerably smaller subset of FLAPs than previously reported.
  - Epitopes may not be directly extracted due to unstable feature lists.
  - Issues with CTD encoding need to be overcome as to give comparable results with the BLOSUM62 substitution matrix representation.
  - GA is a robust and high performing feature selection method.

4.6 Future

Model Selection
From this work it is clear that a robustification of the model selection is needed. When models are searched across a large space of learning system parameters there is great risk for overfitting. Naturally, one would therefore wish to design a method that allows a search across a large set of models, but which still selects models that are not overfitted. To solve this, a change of the criterion used for model selection is needed. Apparently the current criteria give optimistic performance estimates of the model. One way to get around this is to take into account the variance of the performance estimate which the model selection is based upon. Such work is currently in progress. In this work the main selection criterion has been the cross validation error. This error estimator may not be optimal since it can have relatively high variability, especially on small sample sets using a high number of folds.

The problem of unstable feature lists remains unsolved. There are methods that for example takes the intersection of feature lists retrieved from repeated training on perturbed datasets, such as the the sets retrieved from cross-validation folds, and such approaches might be promising[25]. The finding of *TSPYL5* in this study by looking at repeatedly selected features indicates that such approaches can be rewarding.

Application to breast tumour classification
In the case of the microarray dataset it would be interesting to use the genetic algorithm as a feature selection search method as this method seems to perform well.

Apart from *TSPYL5* several genes were identified to be repeatedly selected in the external cross validation procedure more than 10 times out of 25. Litterature and gene database searches would be of high interest to conduct also for these.

Application to allergen classification

Regarding the allergen classification which was the main focus of this work several possible improvements may be envisioned, and which are listed below.

- Further optimization of model parameters when "difficult" proteins are included in the training phase.
- Improved allergen database. The *National Food Administration* of Sweden recently replaced the content of their allergen database, thereby extending it from 762 to 1251 allergens, by importing all sequences from "AllergenOnline" (`http://www.allergenonline.com`)[40], one of the worlds largest and best curated allergen databases, .
- It would be of interest to test other substitution matrices for measuring sequence similarity. It may for example be of interest to try physicochemical substitution matrices instead of evolutionary ones.
- The FLAP-set was now generated based on alignment scores using the BLOSUM62 substitution matrix. It would be more consistent in the case of CTD-encoding representation to generate a FLAP-set based on CTD-encoding.
- It would be of interest to study the extent of overlap of the retrieved feature subsets with sets of sequences from epitope databases.
- It would be of great interest to modify the algorithm as to be able to identify discontinuous epitopes. One idea is to use 3-D descriptors as features. The algorithm presented in this work could then be used by simply substituting the representation, and it would then be able to identify discontinuous epitopes. The amount of 3-dimensional data for proteins are limited but recently there have been successful reports on prediction of discontinuous epitopes using 3D structures[39] indicating that the use of 3D data for epitope identification is feasible.

As a final note I hope that the features added to the toolbox will come to be of future benefit in both education and research. From this work it should be apparent that the toolbox is a powerful analysis tool that helps in making more complete and thoroughly validated studies of the multivariate data at hand.

5 Acknowledgments

With much gratitude I must thank Mats Gustafsson, Ulf Hammerling and Daniel Soeria-Atmadja for superb supervising. I would also like to thank Jonathan Alvarsson for our every day collaboration and discussions. The *Division of Toxicology* at the *National Food Administration* provided me with computational resources without which my algorithms would stand still. Finally I thank Rolf Larsson and the *Clinical Pharmacology* group at the department of *Medical Sciences* for providing the daily office space and an exciting research environment.

References

[1] Hastie A., Tibshirani R., and Friedman J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2001.

[2] Hjern A. Major public health problems - allergic disorders. *Scandinavian Journal of Public Health*, 67:125–131, 2006.

[3] Martinez Barrio A. et al. Evaller: A web-server for *in silico* assessment of potential protein allergenicity. *Nucleic Acids Research*, 35:W694–W700, 2007.

[4] Vachani A. et al. A 10-gene classifier for distinguishing head and neck squamous cell carcinoma and lung squamous cell carcinoma. *Clinical Cancer Research*, 13:2905–2915, 2007.

[5] Webb A. *Statistical Pattern Recognition*. Wiley, second edition, 2002.

[6] Kay A.B. Allergy and allergic diseases. second of two parts. *New England Journal of Medicine*, 344:109–113, 2001.

[7] Efron B. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of American Statistical Society*, 78:316–331, 1983.

[8] Fan C. et al. Concordance among gene-expression based predictors for breast cancer. *New England Journal of Medicine*, 355:560–569, 2006.

[9] Lai C. et al. A comparison of univariate and multivariate gene selection techniques for classificaiton of cancer datasets. *BMC Bioinformatics*, 7, 2006.

[10] Sima C. et al. Superior feature-set ranking for small samples using bolstered error estimation. *Bioinformatics*, 21:1046–1054, 2005.

[11] Venter C. et al. Environmental genome shotgun sequencing of the sargasso sea. *Science*, 304:66–74, 2004.

[12] Ding C.H. and Dubchak I. Multi-class protein fold recognition using support vector machines and neural nnetworks. *Bioinformatics*, 17:349–358, 2001.

[13] Cai C.Z. et al. Svm-prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Research*, 31:3692–3697, 2003.

[14] Mittag D. et al. A novel approach for investigation of specific and cross-reactive ige epitopes on bet v 1 and homologous food allergens in individual patients. *Molecular Immunology*, 43:268–278, 2006.

[15] Soeria-Atmadja D. et al. Computational detection of allergenic proteins attains a new level of accuracy with *in silico* variable-length peptide extraction and machine learning. *Nucleic Acids Research*, 34:3779–3793, 2006.

[16] Soeria-Atmajda D. et al. External cross-validation for unbiased evaluation of protein family detectors: Application to allergens. *PROTEINS: Structure, Function and Bioinformatics*, 61:918–925, 2005.

[17] Goldberg D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[18] Freyhult E. et al. Unbiased descriptor and parameter selection confirms the potential of proteochemometric modelling. *BMC Bioinformatics*, 6, 2005.

[19] Dougherty E.R. The fundamental role of pattern recognition for gene-expression/microarray data in bioinformatics. *Pattern Recognition*, 38:2226–2228, 2005.

[20] Alexe G. et al. Breast cancer prognosis by combinatorial analysis of gene expression data. *Breast Cancer Research*, 8:R41, 2006.

[21] Bannon G.A. and Ogawa T. Evaluatoin of available ige-binding epitope data and its utility in bioinformatics. *Molecular Nutritional Food Research*, 50, 2006.

[22] Lin H.H. et al. Prediction of transporter family from protein sequence by support vector machine approach. *Proteins: Structure, Function and Bioinformatics*, 62:218–231, 2006.

[23] Dubchak I. et al. Prediction of protein folding class using global description of amino acid sequence. *Proceedings of National Academic Science*, 92:8700–8704, 1995.

[24] Dubchak I. et al. Recognition of a protein fold in the context of the scop classification. *Proteins: Structure, Function, and Genetics*, 35:401–407, 1999.

[25] Guyon I. and Elisseeff A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[26] Cui J. et al. Computer prediction of allergen proteins from sequence-derived protein structural and physcochemcial properties. *Molecular Immunology*, 44:514–520, 2007.

[27] Nordlee J.A. et al. Identification of a brazil-nut allergen in transgenic soybeans. *New England Journal of Medicine*, 334:688–692, 1996.

[28] Nevins J.R. and Potti A. Mining gene expression profiles: Expression signatures as cancer phenotypes. *Nature Reviews Genetics*, 8:601–609, 2007.

[29] Stanley J.S. et al. Identification and mutational analysis of the immunodominant ige binding epitopes of the major peanut allergen ara h 2. *Archives of Biochemistry and Biophysics*, 342:244–253, 1997.

[30] Tomii K. and Kanehisa M. Analysis of amino acid indices and mutation matrices fro sequence comparison and structure prediction of proteins. *Protein Engineering*, 9:27–36, 1996.

[31] KTH. `http://www.kth.se/aktuellt/1.2480?offset=25`, 2006.

[32] Ein-Dor L. et al. Outcome signature genes in breast cancer: Is there a unique set? *Bioinformatics*, 21:171–178, 2005.

[33] Han L. et al. Prediction of rna-binding proteins from primary sequence by a support vector machine approach. *RNA*, 10:355–368, 2004.

[34] Han L. et al. Prediction of functional class of novel viral proteins by a statistical learning method irrespective of sequence similarity. *Virology*, 331:136–143, 2005.

[35] Jackson M. Allergy: the making of a modern plague. *Clinical & Experimental Allergy*, 31:1665–1671, 2001.

[36] Uhlén M. et al. A human protein atlas for normal and cancer tissues based on antibody proteomics. *Molecular Cell Proteomics*, 4:1920–1932, 2005.

[37] Stadler MB and Stadler BM. Allergenicity prediction by protein sequence. *Federation of American Societies for Experimental Biology Journal*, 17:1141–1143, 2003.

[38] Regenmortel M.H.V. Immunoinformatics may lead to a reapraisal of the nature of b-cell epitopes and of the feasibility of synthetic peptide vaccines. *Journal of Molecular Recognition*, 19:183–187, 2006.

[39] Haste Pedersen P. et al. Prediction of residues in discontinous b-cell epitopes using 3d structures. *Protein Science*, 15:2558–2567, 2006.

[40] Hileman R.E. et al. Bioinformatic methods for allergenicity assessment using a comprehensive allergen database. *International Archives of Allergy and Immunology*, 128:280–291, 2002.

[41] Duin RPW. et al. Pr-tools 4.0, a matlab toolbox for pattern recognition. Technical report, IGT Group, Technical University Delft, The Netherlands, 2004.

[42] Michiels S. et al. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *The Lancet*, 365:488–492, 2005.

[43] Niijima S. and Kuhara S. Recursive gene selection based on maximum margin criterion: a comparison with svm-rfe. *BMC Bioinformatics*, 7:543, 2006.

[44] Johansson S.G. et al. Revised nomenclature for allergy for global use: Report of the nomenclature review committee of the world allergy organization. *Journal of Allergy and Clinical Immunology*, 113:832–836, 2004.

[45] Venkatarajan S.M and Braun W. New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical-chemical properties. *Journal of Molecular Modelling*, 7:445–453, 2001.

[46] Kim T.Y. et al. Epigenomic profiling reveals novel and frequent targets of aberrant dna methylation-mediated silencing in malignant glioma. *Cancer Research*, 66:7490–7501, 2006.

[47] van de Vijver M.J. et al. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347:1999–2009, 2002.

[48] van't Veer L.J. et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

[49] Li Z.R. et al. Profeat: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequences. *Nucleic Acids Research*, 34:W32–W37, 2006.