NTNU

Norwegian University of
Science and Technology

Prediction of Protein Function using semantic Fingerprints

Multivariate Data Classification by Artificial Neural Networks involving dimensional Reduction

Christoph Linse

MSc in Physics
Submission date: May 2016
Supervisor: Rita de Sousa Dias, IFY
Co-supervisor: Martin Kuiper, IBI

Norwegian University of Science and Technology
Department of Physics

Preface

This Master's thesis is submitted in fulfilment of the requirements of the international Master's program in Physics at the Norwegian University of Science and Technology (NTNU). The work developed into a procreative cooperation between the Department of Physics and the Department of Biology, focusing on semantic systems biology. I am grateful to my supervisor Rita de Sousa Dias from the Department of Physics for her confidence in my work and many fruitful discussions. I would also like to particularly thank my supervisor Martin Kuiper for his prolific way of non confining guidance, his dedication when answering my many questions, his considerations and assistance. I would also like to thank Harald Martens for his experienced advice related to dimensional reduction techniques and soft multivariate data analysis.

Christoph Linse, Trondheim, May 2016

Abstract

In many fields of today's scientific research the amount of knowledge is far smaller than the amount of accessible data and often too limited to make meaningful analyses and draw reasonable conclusions. Hence, statistical knowledge inference becomes more and more popular in multivariate data analysis and machine learning tools from artificial intelligence are applied. However, there are issues when dealing with increased sparsity, high-dimensionality and a lack of training samples which pose new requirements on data analysis tools. Particularly the experimental determination of protein function is challenging and cumbersome. Theoretical predictions are challenging due to the variety and complexity of macromolecules. However, recent massive knowledge integration approaches in systems biology resulted in curated semantic knowledge-based systems that could make relevant problems more and more feasible.

One of these problems is the identification of specific DNA-binding RNA polymerase II transcription factors (DbTFs). In this work semantic knowledge-based systems are exploited for DbTF prediction and the feasibility of the approach is explored. This master's project involves the design, implementation, rigorous testing and optimisation of a specific methodology to classify putative DbTFs by an artificial neural network approach. From 2655 candidate proteins of the TFcheckpoint database [5] a selection of 54 proteins is classified as DbTFs with a relative classification error of less than 10%.

Abbreviations

ANN	Artificial neural network
DbTF	Specific DNA binding RNA polymerase II transcription factor
GexKB	Gene Expression Knowledge Base
GO	Gene Ontology
GOC	Gene Ontology Consortium
MCC	Matthews correlation coefficient
PCA	Principal component analysis
PLS, PLSR	Partial least squares, regression
ROC	Receiver operating characteristic
SVD	Singular value decomposition

A selection of the most frequently used abbreviations and concepts.

Contents

Chapter 1

Introduction

Background

In today's information age machine learning has become a versatile tool in multivariate data analysis and classification. The idea is to determine a set of instances that is represented by descriptive data sequences and dependable key features. These training instances are subsequently used in an iterative learning process, that analyses patterns of similarity in order to find a mapping from the descriptive data on the key features. The trained models can subsequently be applied to statistically infer knowledge about other data sets of the same data structure. In addition, the approach realises variable relevance estimation and offers better understanding of the underlying patterns that explain the key features.

A crucial step in multivariate data analysis is the choice of methodology that suits the structure of the data, provides well interpretable results and allows accuracy estimation. However, there are distinct limitations. The problem arises if the data needs automated treatment but no meaningful statistics can be applied due to the following aspects:
High-dimensional space poses a challenge to machine learning, especially if the algorithm scales badly with the number of variables and if there are many variables (several thousands) compared to a low number of samples (several hundreds). This problem received the catchy name 'curse of dimensionality'.
Sometimes this challenge occurs together with sparse data. In this case noise is difficult to distinguish from relevant patterns. Machine learning becomes prone to over-training, a phenomenon that relates to the loss of ability to generalise and is bound to a significant decrease in prediction accuracy. Training instance selection and dimensional reduction become delicate steps because they might change the outcome completely.
For many classification problems in natural science there exists only a small number of pre-known training samples. In binary classification there is a particular issue if the number of available true positive and true negative instances is unbalanced. Preferably a one-to-one ratio is used for machine learning. If one performs instance selection there will be the risk that the subset does not represent the domain of classification.

One very important part of this project is to address these challenges in respect of the use case that is presented in the following section.

Specialisation and problem description

The use case aims at the classification of specific DNA binding RNA polymerase II transcription factors (DbTFs) exploiting the feature space provided by semantic knowledge-based systems. Semantic knowledge-based systems combine the organisation of complex structured data with an inference engine. These offsprings of information technology realise a vast feature space that might make the use case problem more and more feasible.
The project involves the development, implementation, rigorous testing and optimisation of a specific approach to regress, classify and analyse sparse, high-dimensional data sets by an artificial neural network (ANN) approach. A set of candidate proteins, that is extracted from the TFcheckpoint database [5], is ranked according to the classification results.

A DbTF is a protein that is crucial for regulation of gene expression in eukaryotic cells. Cells adapt dynamically to changes in their surrounding in order to survive or to provide differentiated function in a multicellular organism. Matching the new requirements may involve a biochemical change of metabolic pathway activity - triggered by highly interactive and complicated protein regulation processes. Protein regulation can be achieved by inhibition or activation of subunits by allosteric or cooperative binding to other proteins or chemical ligands, as well as covalent modifications. However, it can also depend on the control of protein concentration, which is a lower level regulation and more resource efficient.
Protein synthesis in eukaryotic cells is achieved in a multi-step process including transcription, translation and further modification. During transcription a pre-initiation complex binds to a specific promoter region on DNA. DbTFs bind non-covalently to very specific DNA sequences that are part of the core promoter region of protein-coding genes and might also interact selectively with other proteins or complexes. They modulate the level of activity of the RNA polymerase II complex and thereby affect gene expression. RNA polymerase II uses the DNA as a template and creates an inverted copy of the four base sequence. A single RNA strand is obtained and further modified. Subsequently, the RNA strand is transferred from the nucleus to the cytosol. At the rough endoplasmic reticulum the ribosome, a multiprotein-RNA complex, synthesises a polypeptide chain utilising the RNA strand as a template. Subsequently the nascent protein is folded and possibly covalently modified in order to obtain its final 3D conformation and biological functionality. The rate of protein synthesis can therefore be directed by transcription control during the formation of the pre-initiation complex.
A very detailed introduction to fundamental molecular cell biology can be found in the book [25].

Comprehending the process of gene expression regulation is important for a diverse range of research, including cell function, drug research and cancer treatment. The growing knowledge about molecular, cellular processes together with the data analysis tool, that is developed in this thesis, might enable further improvement in exploiting limited data and will hopefully make the use of semantic knowledge bases generally more popular in current research.

The scientific work focuses on the following questions:

Semantic knowledge based systems How much useful information does gene ontology data contain for specific DNA binding transcription factor classification? To what extend do semantic knowledge-based systems suit protein property prediction?

Dimensional reduction in sparse data How well perform different dimensional reduction tools like principal component analysis (PCA), partial least squares (PLS) and a variant LPLS in sparse multivariate space? Can a transparent and well scalable variable relevance estimator be found? For studying the last question a method called separability analysis is proposed, tested and compared to other approaches.

Specific DNA binding transcription factors (DbTFs) What biological conclusions can be drawn related to DbTFs?

The approach

The resulting work flow is illustrated in figure 1.1. The flowchart is structured into four subtasks, including data retrieval, preprocessing, machine learning and postprocessing. For preprocessing there are several methods proposed. These methods are tested and discussed in chapter 4 and a particular methodology will be chosen that suits the data. This decision process is illustrated in the flowchart as blue diamonds.
In the following the procedure is stepwise introduced.

Data retrieval

During data retrieval a selection of DbTFs and non DbTFs is made and patterns are defined that are next to be analysed. The choice of data determines the scope of epistemic insight because it depends on underlying assumptions and provides the basis for prediction model creation.
Protein function prediction can be based on different kinds of data. For example, amino acid sequences provide the building blocks of proteins. Amino acid sequence similarity measures can be computed to infer protein function [21]. Jensen et al. [21] predicted protein function with features computed from amino acid sequences such as post-translational modifications and localization properties. Later, Jensen, Gupta, Staerfeldt and Brunak [22] used protein sequences to predict gene ontology annotations. However, efforts for high scale integration of distributed resources have been undertaken.

At the beginning of the project there was the vision to exploit a vast feature space obtained from an extensive data Knitting Tool that should combine multiple sources including biological semantic databases, inferred knowledge as well as biophysical and biochemical data in a Semantic Knowledge Base (SKB). This comprehensive SKB was planned by NTNU's Semantic Systems Biology research group, and aimed at integration of protein motive data, physical characteristics, protein-protein binding knowledge, network proximity information, multi-level interaction between proteins in pathways, ect... However, the SKB did not develop to an accessible data source. This unexpected development made it necessary to find a new, adequate source of data.
The Gene Expression Knowledge Base (GexKB) was considered to be an alternative. GexKB integrates knowledge about proteins like Gene Ontology (GO) data, Molecular Interaction (MI) Ontology, as well as the weighted gene regulatory network Biorel and forms a seed ontology out of these [43]. Subsequently, more data sources are exploited including protein modification data or the association of proteins with specific diseases, ect. In a screening phase of this project prior data sets were tried

with varying success. Most of the different data types did not show promising results due to limited data or low correlation to the key feature. Combinations of different data types were not tested, because the variable space would explode. The gene ontology annotation data turned out to be the most useful information, because preliminary classification results were better than for any other information type. The combination of the sub-ontologies GO Molecular Function, GO Cellular Component and a selected part of GO Biological Process was able to distinguish many DbTFs from non DbTFs. GexKB includes these ontologies but was created with transcription factors and other proteins and complexes that these bind to and no negatives. In order to get unbiased, comprehensive protein data for both DbTFs and non DbTFs, the Gene Ontology Consortium [2] was selected as data source.

Preprocessing

The descriptive data contains about 11000 dimensions. Preprocessing provides necessary dimensional reduction and enables variable relevance estimation. The reduction includes a variable selection procedure and the construction of new features that summarise relevant variables in a linear combination.

Machine learning

The idea is to create a prediction model that maps the reduced data on the dependable key feature, i.e. being a DbTF. Artificial neural networks (ANNs) represent versatile tools for prediction and are intensively used by companies like Google or Facebook for this purpose. A very good introduction to neural networks is given in the books [11, 35].
Neural nets are composed of modular data processors, called neurons. The networks are inspired by the design of the cerebral nervous system. An ANN has an input and output interface that is linked by a network of neurons modelling a sub-symbolic function. The number of inputs and outputs correspond to the dimensionality of the input data and the key features, thereby enabling multivariate information processing. Neural networks are known for their ability to learn complex input / output relationships and to identify characteristic pattern signatures in the data using their generalization abilities. This makes them particularly suited for pattern recognition.

Because of the complexity of the approach no other predictors like decision trees or support vector machines are applied in the scope of this project. Possibly the transparent and well interpretable preprocessing and postprocessing combined with the somewhat opaque but very adaptive machine learning approach will lead to both, understanding and good prediction performance.

Postprocessing

Postprocessing is crucial for validation and the detection of over-training. An over-trained model has lost its ability to accurately predict unknown instances because it has learned noise instead of the underlying relationships. The accuracy of the classification is analysed with validation instances that are not used for training. The techniques cross-validation and receiver operating characteristic (ROC) are applied.

As artificial neural networks lack a symbolic model it is difficult to learn about the relevant underlying patterns in the data. In addition, a sensitivity analysis is conducted and the influence of each variable on the classification result is measured. Ideally a subset of variables can be found that incorporates less noise and leads to better training and classification results. The resulting variable ranking should give further indication for the variable importance estimation from preprocessing.

Finally the candidate proteins are classified and a ranked list of candidates is created. The results should provide additional evidence to the research community, for instance a priority measure to desing further experiments.

Earlier realisations

Machine learning and artificial neural networks are increasingly applied in natural science. The project goal seems to be feasible as similar approaches have been carried out for other kinds of data - with great success.

De La Calleja and Fuentes [6] automated galaxy morphology classification based on pattern recognition of galaxy image data. They employed principal component analysis for dimensional reduction to overcome the dimensionality problem. Their prediction model was able to classify new galaxies fast and efficiently. Another example is evaluation of the large amounts of data produced in high energy physics. Massive amounts of sensor data from particle collisions is collected, but there might be irrelevant events. The work of Shimon and Daniel Whiteson [44] deals with application of machine learning for event selection.
More examples can be found in bioinformatics. Round blue cell tumors were classified based on gene expression signatures by Khan et al. [23] with the help or artificial neural networks. Their inspiring work influenced the structure of the approach of this project. Hapudeniya [16] provides an extensive overview on ANN applications in bioinformatics. Xu et al. [46] applied a neural network based system called Gene Recognition and Analysis Internet Link (GRAIL) for finding the location and significance of genes in a genome based on uncharacterised DNA sequences. Another field of ANN application is quantitative structure-activity relationship (QSAR) modelling. The idea is to capture biological activity of chemicals based on their chemical and physical properties. QSAR modelling with ANNs becomes more and more popular in recent research and spans from enzyme reactivity assessment [40] to prediction of function of structurally diverse ligands [30]. Seguritan et al. [38] utilised ANNs to classify viral and phage structural proteins according to amino acid composition. Cell cycle related genes were found and assessed by Lichtenberg, Jensen, Jensen and Brunak [7]. They employed an ANN ensemble that was trained with quite generic protein features, for example, phosphorylation, glycosylation, sub-cellular location and instability / degradation.

Report structure

In chapter 2 a literature review is performed in order to make a well informed proposal for the methodology. The set of methods is developed with a focus on the particular use case problem, but the procedure should remain a basis for a broad

range of applications. This chapter includes also an introduction to artificial neural networks.

In chapter 3 the training data is selected. Relevant steps are discussed and argued including instance and feature space selection. Four subsets with different instances are constructed in order to enable a stability check.

The novel approach is implemented, optimised and discussed in chapter 4. Different dimensional reduction techniques are tested for suitability. Based on a discussion of preliminary classification results one particular methodology is chosen for final application. Subsequently candidate DbTFs are classified and the results are carefully discussed in chapter 5.

The gained experience is used to additionally formulate a proposal of how to predict drug synergy by machine learning. This future prospect is related to the AstraZeneca-Sanger DREAM Challenge and is covered in chapter 7.

The report completes with an outlook that gives recommendations for similar implementations and suggestions for future work.

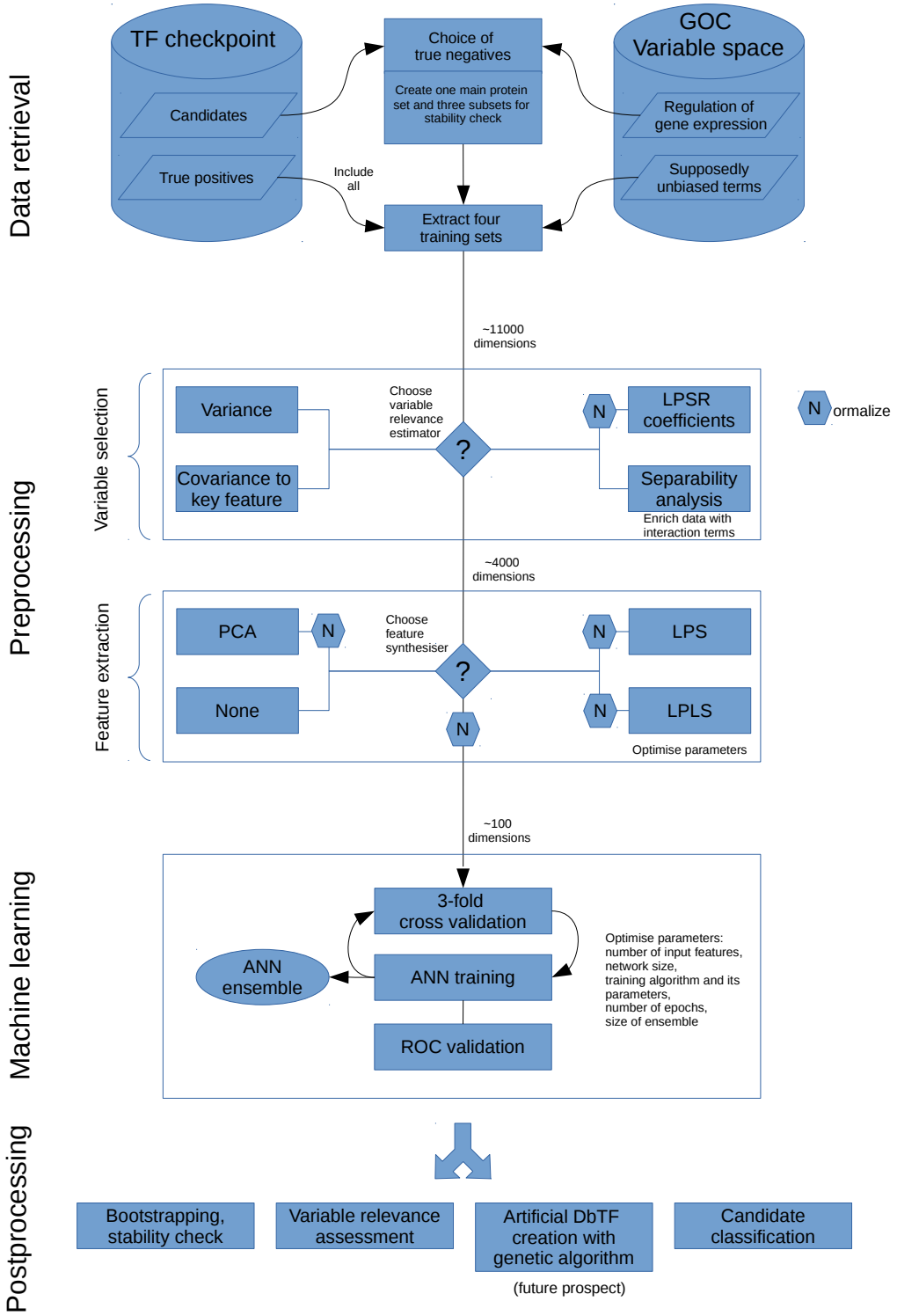


Figure 1.1: The process of data analysis is illustrated from data retrieval to candidate classification. The process includes decision points visualised by blue diamonds. The respective four options are implemented and tested for the gene ontology annotation data. A summary of the process is given in the sections 1 and 2.4.

Chapter 2

Methodology

In this chapter the methodology for protein classification is developed. The approach focuses on the particular context of binary gene ontology annotation data that is expected to be characterised by sparsity, redundancy and about 10000 dimensions. A literature review is performed, state-of-the-art techniques are introduced and critically discussed in terms of their suitability. Subsequently, an intermediate methodology is proposed in section 2.4. An overview of the resulting work flow is illustrated in figure 1.1. The proposal is subsequently tested and assessed with the protein data in chapter 4.

Terminology In this project report the term 'prediction' is the umbrella term for 'classification' and 'regression'. The term 'classification' is used if the key feature (or features) is discrete. 'Regression' refers to one or several continuous key features. The specific use case of this project is a binary classification problem. The key feature is being a DbTF or a non DbTF.
The terms 'instances' and 'samples' are used synonymously in this report. In the use case the instances are always proteins or their data representation.
In the following discussion, the term 'variable' will be used for a descriptor in the original, unreduced data. 'Features' on the other hand correspond to the selected variables and computed components that will be used as machine learning input. In this report a component is a linear combination of variables.
There are two other terms that are explained to avoid confusion. The term 'variable selection' is used for finding a subset of variables and pruning irrelevant ones. 'Feature extraction' means to construct components from the variables and include these in the training data.

2.1 Preprocessing

The proteins are represented as high-dimensional binary vectors that encode the existence of gene ontology term annotations. The combination of these vertices form a matrix X that is about to be analysed in a pattern recognition process. Separately, a binary matrix Y is obtained incorporating the dependable information. As there is only one key feature, Y reduces to a one dimensional array encoding whether the protein is a specific DNA binding transcription factor (DbTF) or not.

Preprocessing is related to dimensional reduction in order to remove redundancy and noise from the descriptive data matrix X. Many approaches belong to one

of the following two concepts: One approach is to select a subset containing the most promising variables and prune the irrelevant ones. These variable selection techniques differ by the definition of variable relevance. The other strategy is to find combinations of variables to create advanced features that summarise the original variables - a process that is sometimes referred to as feature extraction.

In the following section a subset of techniques is introduced and discussed in terms of suitability for the particular use case problem.

2.1.1 Variable Selection

Different mechanisms for variable selection are proposed, applied and compared in the literature [14, 20, 40]. Usually a variable ranking is performed that can subsequently be used for subset creation in a forward or backward selection approach. Forward selection processes start with an empty set that is iteratively enriched with the most promising variables. Backward selection starts with the full domain of variables and prunes the least promising ones. As backward elimination assesses variable relevance in the interplay of all remaining variables, it is supposed to better take interdependencies between variables into account [14] at the cost of computational effort.

Supervised and unsupervised

Unsupervised filtering aims for ranking variables focusing on the inherent structure of the descriptive data. The number of possible criteria is tremendous, however usually a subset of maximal entropy is desired. Unsupervised methods only focus on the intrinsic and inherent structure of the descriptive data and do not necessarily capture the relationship to the key feature. By nature this class provides flexibility because it is not influenced by any kind of prediction model.

Supervised methods consider the dependency between variables and key feature. These approaches take the dependent key feature into account and hence judge variables or data patterns from a goal orientated perspective. Sometimes, machine learning is applied as a black box mechanism. A measure of performance is computed and iteratively maximised by selecting variables in an optimisation step. If the machine learning models are independent from the final prediction approach, the selection technique is called a supervised filter. If the variable selection process is performed with the help of the final machine learning approach, it is often called wrapper [14].

Unsupervised and supervised approaches are controversially discussed in the literature. Varshavsky, Gottlieb, Linial and Horn [41] recommended to stick to unsupervised methods in order to keep the data unbiased and have a lower risk of over-training. Feature selection should base on the descriptive data only and should not be influenced by a modelling function like supervised methods would do.
Boulesteix [3] argued that supervised methods are usually better suited for prediction, because they are goal driven and work with a large variety of data sets. Though supervised filters and wrappers are relatively easy to apply they might have some limitations: blackbox predictors are prone to over-training, an issue that is in detail described in chapter 2.2. In order to detect and avoid over-training, the data is split into training and validation sets and cross-validation is performed. One could argue,

that the variable selection process is only based on smaller bootstraps and not on the entire data at once [14]. Also, the approach is only feasible if dimensional reduction is not necessary for the predictor to work.

Collinearity

Some standard filters are easy to apply but some lack some essential common sense as they ignore the interactions between variables [40]. If the interplay of two variables can be modelled with a linear relationship with reasonably good accuracy, there is redundancy.
There are controversial discussions in the literature about the criteria for removing correlated variables. According to the review of Guyon and Elisseeff [14] one might consider to remove variables with very high correlation. Szaleniec [40] recommends to remove all variables that have a pairwise correlation coefficient larger than 0.9. Simultaneously, one should acknowledge that synergistic effects could make correlated variables particularly valuable, so the threshold could be problem dependent. A lack of verifiable criteria complicate threshold selection. Guyon and Elisseeff [14] recognise that even on the first glance irrelevant variables can surprise in the presence of others. Probably redundancy is not a big problem, if the collinearity follows the same structure in all data sets. This situation is sometimes referred to as pattern consistency.

Some state-of-the-art and complex variable filters exist, that take collinearity into account. SVD-entropy is proposed in the work of Varshavsky, Gottlieb, Linial and Horn [41] and has been proved to be successful in biological contexts. Entropy is computed with and without a variable in a leave-one-out scheme. The change of entropy is computed and interpreted as a relevance score. However, the unsupervised nature of SVD-entropy could lead to a loss of relevant information. Furthermore, multivariate space with about 10000 dimensions excludes backward elimination techniques because of computational reasons.

Model restrictions

Supervised applications often make use of a prediction model that is bound to model characteristic limitations. Variable selection may be less transparent due to model complexity. This poses a risk when no justifiable criteria for threshold selection can be found. There is the risk of over-training. Also, a bias might be introduced to the data by overlooking or overemphasising specific structures. For example, a linear model might neglect variables that have nonlinear characteristics.

Surprisingly, a flexible supervised method that minimises model restrictions could not be found in the literature. This is why a very simple, rather model free ranking procedure is proposed. Variable relevance can be assessed for single variables in a virtual one-variable-classifier. In addition, interaction terms can be added similarly to a Taylor expansion. The so-called separability score quantises the ability to resolve the key feature by density histogram computation. It can be applied on binary classification problems and it might be possible to modify it for regression tasks, however this lies beyond the scope of this project.
Separability analysis is explained in figure 2.1 with the help of artificial data that simulates different variable distributions. Two groups of instances, A and B, have each a density distribution within a variable x_0. The variable x_1 is only shown to

improve readability. For each group a histogram with n bins is computed separately. As depicted in the diagram in figure 2.1, both histograms are subtracted from each other and the absolute is taken to obtain a third histogram, which has a green color. The sum of the bins in this difference-histogram is called separability score and is interpreted as a relevance measure. The separability score s is computed by

$$s = \frac{1}{cN_A + N_B} \sum_{i=1}^{n} |c \cdot A_i - B_i| \tag{2.1}$$

where n is the number of bins of the histograms, N_A is the total number of instances in group A and N_B is the total number of instances in group B. A_i and B_i are the counts of instances in the bins of the histograms for groups A and B. In the formula the histogram of group A is scaled by a factor c. If the fraction of total instances $c = N_A/N_B \neq 1$, one group has an excess of instances. One histogram can be scaled accordingly to give equal contribution of both groups to the separability score.

In the next iteration pairs of variables are assessed in order to consider first order interaction terms. The procedure is similar to the variable subset selection process from de Lichtenberg, Jensen, Jensen and Brunak [7]. In an iterative process combinations of variables are ranked according to their combined prediction performance as a two-variable-classifier. This involves two dimensional histogram computation, subtraction and integration like for the one-dimensional case. The best pairs are used to build new combinations. One logical operation is chosen to create a new term that is added to the data. Five different operations are considered: x_0 AND x_1, x_0 OR x_1, x_0 AND $\neg x_1$, $\neg x_0$ AND x_1, x_0 XOR x_1. One-dimensional separability analysis is applied on the new terms and the best logical operation is chosen. For continuous variables, other operations like multiplication might be considered.
The approach gives an opportunity to detect difficult checker board arrangements like the XOR challenge - a setting that is shown in figure 2.2. Continuous data is used to improve readability. No one of the two variables shown can resolve the key feature. However, two-dimensional separability analysis is able to detect a nonlinear relationship by histogram computation.

The model's only free parameter is the number of bins for density computation. The standard deviation of the separability scores with n, n-1 and n+1 bins offers an accuracy estimate and the average represents a more robust result. The number of bins should not be chosen too high as this would correspond to over-training. There are two reasons for this: First, the assumption should hold that the bins still contain enough instances to enable statistical analysis. Second, the resolution that the bins imply should be able to resolve the relevant structures in the data. There should be a very low risk of over-training if the number of bins is significantly (like one percent) lower than the number of instances. For binary data however there is no need to choose the number of bins.

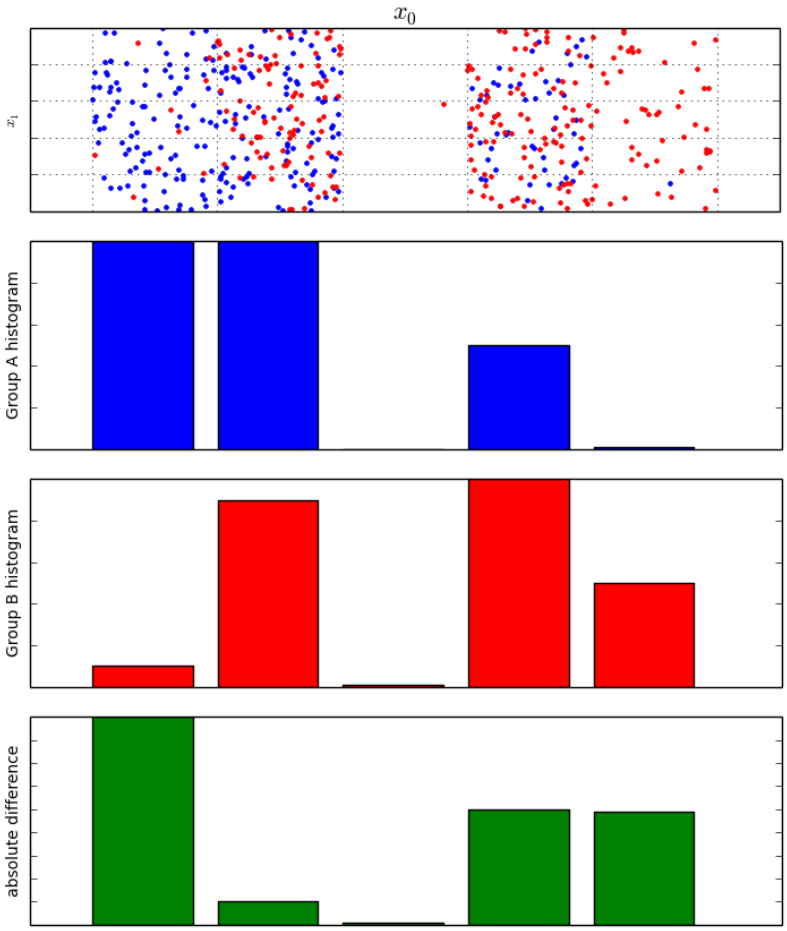
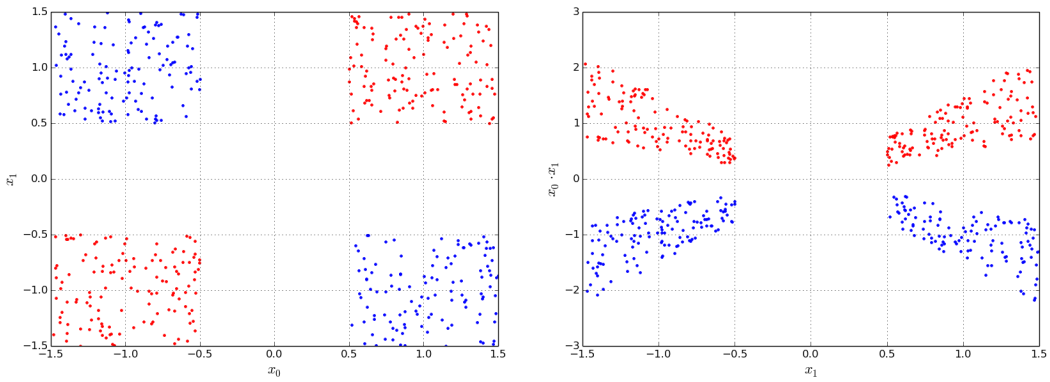


Figure 2.1: Separability score computation is illustrated with artificial data in a binary classification problem. Different density distributions in variable x_0 are presented for demonstration. The axis x_1 is not relevant, but is useful for visualisation. The blue and red histograms show instance counts. The green histogram is the absolute difference of the blue and red ones. The normalised sum over the green bins is interpreted as relevance score for the variable x_0.



(a) The two groups are defined by an XOR function that separability analysis can detect.

(b) The multiplication of the two variables creates a new feature that can be further treated by linear techniques.

Figure 2.2: The challenge to detect checker board problems is addressed by separability analysis. This example shows an XOR relationship.

Proposal for variable selection

A set of four approaches is chosen to be tested for suitability. Supervised, unsupervised, linear and nonlinear methods are contained in order to enable flexible hybrid applications. Because of the size of the descriptors space, forward selection processes are chosen at the cost of overlooking variable interactions. This means that variable selection has to be conducted very carefully to retain nonlinear information.
The proposal includes a) variance analysis (unsupervised ranking), b) separability analysis (nonlinear supervised ranking), c) correlation to the key feature (linear supervised ranking) and d) absolute PLSR regression coefficients (linear supervised filter). The options are illustrated in figure 1.1.

a) A basic unsupervised filtering approach is to rank variables according to standard deviation. A problem specific threshold is chosen in order to exclude rather constant variables. This seems to be fast, simple and reasonable strategy for the gene ontology data as the majority of terms has less than 1% non zero entries.

b) Separability analysis is a rather model free supervised method, that is not mentioned in the literature that was reviewed. It is applied to estimate nonlinear relevance of both single variables and variable pairs. In addition it is applied to enrich the descriptive data with first order interaction terms.

c) A very greedy approach is to rank variables based on their correlation to the key feature. For each variable the 2×2 correlation matrix is computed and one of the non-diagonal entries are chosen as signed relevance score. It might be interesting to compare the results to the other methods because the method offers intuitive interpretation.

d) In supervised filters computationally efficient models can be employed that cope with many dimensions. A time and memory efficient linear model is chosen. Later, a nonlinear predictor can be used for final classification. Partial Least Squares Regression PLSR can be used for both data reduction and variable importance estimation [27, 45]. It is introduced in the next section about feature extraction. One option is to interpret the linear regression coefficients as relevance scores.

As dimensional reduction is necessary to enable machine learning, a wrapper technique is not applied in preprocessing. Because of limited time and for computational reasons, some prominent techniques like SVD-entropy or advanced embedded systems that combine training and variable selection are not considered.

2.1.2 Feature extraction

Feature extraction aims at finding combinations of variables that form a compact, less redundant and less sparse representation of the data. The entire data contributes to the new representation so that synergistic effects can be preserved. In the following, the feature extraction techniques PCA, PLS and LPLS are presented.

As variables might have different units and ranges of values, they might not be directly comparable to each other. The following approaches depend on the scaling of the data and antecedent normalisation is necessary. Sometimes the standard scores are computed (e.g. [1, 20, 36, 37]):

$$z = \frac{x - \bar{x}}{\sigma}$$

with the standard deviation σ. In any case variables are mean centred.

PCA Principle component analysis (PCA) is an unsupervised approach to capture axes of highest variance and to reveal the intrinsic structure of data. Originally it was formulated by Karl Pearson in 1901 [33], but it is argued if Hotellings [17] was the actual inventor in 1933. Once proposed for multivariate data analysis in social science, PCA diffused into a wide range of science and industry. A very good introduction can be found in the papers [20, 47].
The basic idea of PCA is to find an orthogonal set of bases in variable space where the new axes, called principal components, are orthogonal and not correlated to each other. Hence, the procedure involves singular value decomposition of the covariance (or correlation) matrix. The linear independence of the principal components solves the multicollinearity problem. Ideally, a small choice of principal components can be extracted, explaining the major variance of the whole data. The technique is illustrated with some artificial data in figure 2.3.

The covariance between two variables a and b that have N samples is defined by:

$$cov(a,b) = \frac{1}{n-1}\sum_{i=1}^{N}(a_i - \bar{a})(b_i - \bar{b}) \tag{2.2}$$

The bar over $\bar{a}$ denotes the mean of a variable. Let X be the centred $(N \times P)$ data matrix that is about to be analysed with N instances and P variables. The matrix product $X^T X$ is a scaled covariance matrix as one can see in 2.2. After decomposition $X^T X = S\,V\,D^T$, D contains the right handed eigenvectors called loadings. $S\dot{V}$ contains the scores which represent the new coordinates of the instances. The new bases are called principal components and are already sorted with respect to the variance they explain [20]. Components can be selected to reduce the dimensionality of the data. Instead of singular value decomposition an iterative algorithm called NIPALS [18] can be applied in order to quickly extract only some of the first principal components.

PCA is successfully applied in the work of Khan et al. [23] on biological data and in the papers [18, 39] on spectral data before classification. It is recommended as effective tool for multivariate data analysis because the principal components can usually be associated to an interpretable effect or cause [40]. In addition, PCA has clustering potential. In the work of Yeung and Ruzzo [47] the principal components are used to capture cluster structure with great success.
Effectiveness of PCA when dealing with discrete data is critically analysed in the paper of Kolenikov, Angeles et al. [24]. It is pointed out that results worsen when dealing with ordinal variables that incorporate many categories. In this case correspondence analysis should be preferred. This is not a problem for the use case because binary data has only two categories.
The effect of PCA on prediction is controversially disputed in the literature. Janecek and Gansterer [20] point out that data with significantly more attributes than instances might be challenging. Another disadvantage of PCA lies in its unsupervised nature: It may condense the intrinsic structure of the data and give well interpretable principal components, but it ignores the relation to the dependable key features and therefore might neglect important information. The first principal components, which include the majority of variance, are probably not the components that are relevant for prediction [1]. The percentage of variance that the choice of principal components contains does not relate to clustering quality [20, 47]. Chang [4] discusses

an example with artificial data where the most relevant information is found in the few first and last principal components. According to Yeung and Ruzzo [47] the first few PCA components are as good for classification as a random choice. This is shown by treatment of biologial gene expression data and artificial data. The method may produce many components that preserve too many unnecessary details within the data including systematic noise [41].

There is the idea to use a search algorithm in order to find the most useful PC's and prune the others [23, 47]. Another option is to apply the methodology of variable selection on the PCA components. The last strategy is implemented and tested in chapter 4.2.

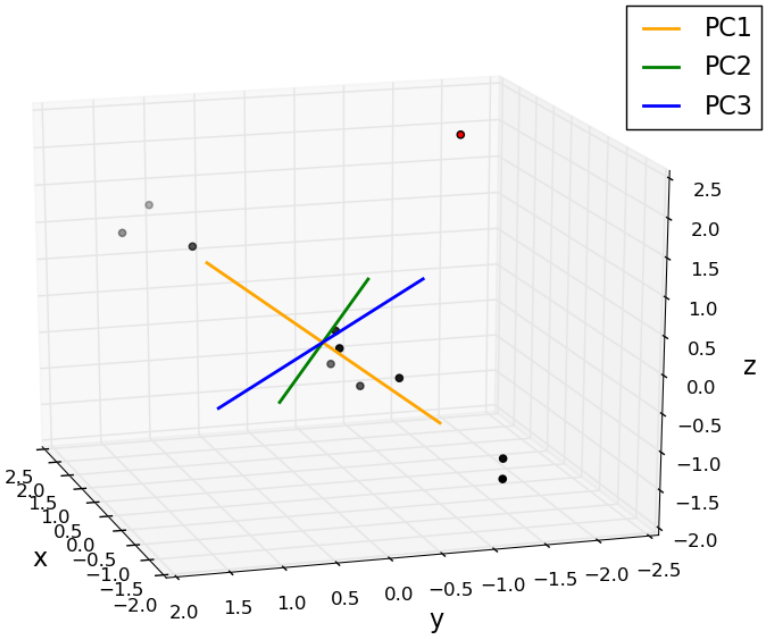


Figure 2.3: The functionality of PCA is visualised. The principal components are illustrated as lines. One observes that the components are sorted according to their variance contribution.

PLS Unlike PCA, which focusses on the variance within the variable space, supervised partial least squares regression (PLSR) takes the dependency between the descriptors and key features into account. The approach is well described in the papers [1, 3] and is applied in this project because of its growing popularity. PLS finds a set of linear combinations that perform a simultaneous decomposition of the descriptive X data with N instances and P variables and the key feature matrix Y with N instances and K key features. The decomposition is obtained with the restriction that the linear components - also called latent vectors - maximise the covariance between X and Y. A variant of the NIPALS algorithm can be applied for latent vector extraction in reasonable time. In this project the NIPALS algorithm described in [1] is implemented. The PLS scores are used as a reduced, compact representation of the descriptive data.
The computation of the scores can be optionally followed by a linear regression step. The regression coefficients can be used to estimate variable relevance. In this project the idea is to use a nonlinear classifier for the final prediction and apply PLS only for dimensional reduction, neglecting its regression results.

According to Boulesteix [3] PLS is the only known feature extraction technique that can cope with very high-dimensional data that has more variables than instances. In the paper of Nguyen and Rocke [32] PCA and PLS are compared to each other with micro array gene expression data in the context of tumour detection. In most cases PLS clearly outperformed PCA.

The number of extracted components determines the amount of useful information and irrelevant noise in the reduced data. Apparently there is no widely preferred procedure to choose the number of latent vectors. In some cases a supervised filter is applied and the PLSR regression results are optimised in respect to the number of components to include [1]. Validation techniques like cross-validation are applied to detect over-training. However one could argue that the linear regression model overlooks nonlinear relationships. As a nonlinear predictor is applied in this project, the loss of nonlinear patterns should be minimised. Instead of PLS regression, the final machine learning technique is used in a wrapper approach to choose the optimal number of latent vectors.

LPLS Sometimes in difficult problems there it the necessity to exploit additional information as effectively as possible. A supervised approach, that is conceptually similar to PLS, includes an additional data block in the component construction process. Let X be the descriptive data matrix with N instances and P variables, Y the key feature matrix with N instances and K key features and let Z be the additional matrix of the shape $P \times L$. The data blocks are arranged in an L-shape with X in the center (hence the name LPLS). Exo-LPLS aims to maximize covariance between the data blocks X, Y and Z with Y and Z containing the dependable data. The components are automatically sorted in their level of covariance explanation [37]. As the shape of Z already implies ,the additional information matrix Z should depict the variable space. One option is a distance matrix that explains relationships between variables, but in principle every data describing the variables can be chosen.

A variation of the NIPALS algorithm that is applied to extract a number of the first components [36]. Before its application the data blocks are centred [37]:

$$Y^0 = Y - 1_N \bar{Y}^T \tag{2.3}$$

$$Z^0 = Z - \bar{Z} 1_N^T \tag{2.4}$$

$$X^{00} = X - \bar{X} 1_N^T - 1_N \bar{X}^T + 1_N \bar{\bar{X}}^T 1_K \tag{2.5}$$

LPLS NIPALS contains a free parameter α. The parameter weights the influence of the Y and Z matrices on component construction. $\alpha = 1$ would assign maximal contribution to the supplementary information Z while $\alpha = 0$ would minimize the contribution [36].

Nonlinear techniques The two supervised feature extraction methods, that were introduced, are restricted to linear models. A number of nonlinear dimensional reduction techniques exist. A self organizing network is a prominent example and a good introduction can be found in the book [9]. Variants are proposed and tested in the paper of Roweis and Saul [34] and in the work of Demartines and Hérault [8], where a self organizing network is applied to map a sub-manifold on an output space for dimensional reduction. However, some methods are not likely to work in very high dimensional space and there is an additional challenge when the type of nonlinearity

is not known beforehand. Hence, nonlinear techniques are not focused on in feature extraction but might be promising for future applications.

Proposal Four options are considered for feature extraction. These include a) PCA, b) PLS, c) LPLS and d) no feature extraction. The different proposals are visualised in the flowchart 1.1 and tested in chapter 4 for suitability. The procedures are summarised below.

a) Though PCA does not seem to suit some prediction problems, this soft-modelling approach can give valuable insight into the data and help to understand its inherent structure. Including PCA in the proposal will enable subsequent comparison between an unsupervised method and the supervised methods. The procedure is divided into a two step process. First, PCA is applied and all principal components are obtained. This does not lead to dimensional reduction yet, as the total variance is conserved. In the second step relevant principal components are identified by one of the variable relevance estimators that are covered in variable selection in chapter 2.1.1.

b) PLS summarizes covariance and hence could suit the classification task at hand. All in all PLS seems to be discussed far less controversially than PCA in the literature. The idea is to use the PLS scores as a compact representation of the descriptive data and use them to train a nonlinear classifier.

c) AS not much neutral or critical comparisons between LPLS and the other methods were found in the literature, it seems to be interesting to include it in the proposal. In section 4.2.3 additional data is selected to guide the dimensional reduction process.

d) Feature extraction is skipped in order to observe the influence of feature extraction on prediction. The effect of collinearity on prediction can be studied.

Nonlinearity is not considered in feature extraction. The idea is to enrich the data with interaction terms found by separability analysis. The enriched data can be subsequently judged in a linear fashion by feature extraction methods.

2.2 Machine learning

Introduction to artificial neural networks (ANNs)

Artificial neural network predictors are still state-of-the-art and very popular in the literature. They combine adaptivity and good performance with an aesthetic concept. An Artificial Neural Network (ANN) is a computer algorithm with a modular structure, which is inspired by the human nervous system. A neural network consists of building blocks, the neurons, which function like two-state threshold elements [13]. Their ability to implement highly nonlinear functions was discovered by the pioneers McCulloch and Pitts [26].
The structure of a neural network is defined by the number of neurons and their linkage. The neurons are interconnected by weighted links that correspond to synapses in the human brain. A signal propagates through the network either repressing or stimulating other neurons depending on the link weights. A neuron functionally consists of dendrites, cell body and axon - or in computational science - input, activation function and output. The so-called activation function maps the

summed input potential on the output value. In the implementation used in this project the artificial neurons do not have any kind of internal state. The input value of a neuron n is hence computed by

$$U_n = \sum_{i=1}^{N} w_{ni} x_i + \Theta \tag{2.6}$$

and the output value is $Z(U_n)$ for the n-th neuron with N predecessors, weights w_{ni}, a bias term Θ and the activation function Z. The quantity U_n corresponds to the potential that is accumulated in the axon until the neuron fires. The number of neurons, the types of activation functions and the link weights fully determine the behaviour of the system.
The activation function can be chosen freely, as long as it does not diverge. Logistic or sigmoid functions are frequently chosen because they offer a lower and upper threshold. This property is useful for discrimination of an active and inactive state. For this project sigmoid functions are implemented.

Training of neural networks

The ANN is trained to map the reduced data on the key features in a supervised, iterative optimization approach. First the ANN is randomly initialized. The weights of the neural links are modified in order to minimize the summed square error of the prediction. This task is delicate because the error hyper surface is characterised by local minima and sharp structures. Feed-forward ANNs are rather difficult to train but in the last years powerful backpropagation methods have been developed. In backward propagation the output error is propagated backwards through the network and the weighted links are corrected to minimise the prediction error. The choice of the training algorithm affects training time and the quality of the classification.
A modified training algorithm, called Rprop [19], is applied in this project. The *fann* package is employed (`http://fann.sourceforge.net/fann.html`) because it offers a flexible and fast neural network implementation in C.

Input encoding

Discrete data should be encoded in binary format because other ordinal variables would imply a numeric relationship between categories that does not exist. In general, the input should be normalised to be in the output range of the activation functions used, like $[-1, 1]$ for the sigmoid functions implemented in this project.
Another issue is an unequal amount of true positive and true negative instances in the training data. Usually the training process assumes, that the numbers of true positives and true negatives are the same. If one class is overbalanced, the network might generally prefer this class. The learning rate could be adjusted to remove the bias, but in the literature this solution was not fount to be thoroughly tested. A different solution is considered. Equally sized bootstraps are taken from the domains of true positive and true negative instances. Multiple predictors are trained and embedded in a neural network ensemble. Thus, the entire information is used.

Network structure

There are many options to link neurons together and construct network structures that show different characteristics, abilities and limitations. The structure of an ANN is determined before machine learning if no advanced nested method is used. ANN structure has an important impact on prediction performance, learning speed and how prone the network is to over-learning.
In a feed-forward network the neurons are arranged in layers. One layer of neurons is fully linked to the previous and next layer, but inside one layer there is no interconnection as this would introduce some kind of unwanted memory effect. One can distinguish between the input layer, an arbitrary number and size of hidden layers and an output layer.
In figure 2.4 an example of a neural network is given and activation patterns are illustrated with colors. The example is useful to understand how information is propagated through an ANN. The information flow is always in the direction from the input layer (left) to the output layer (right). The hidden nodes - if any - further process the information internally and forward it to the output nodes. One can see that the activated input nodes (blue big circles on the left) activate one of the hidden neurons due to promoting red links. This particular hidden neuron activates the lower output neuron. A bias node, which is always firing independently from the input, activates the upper output neuron.

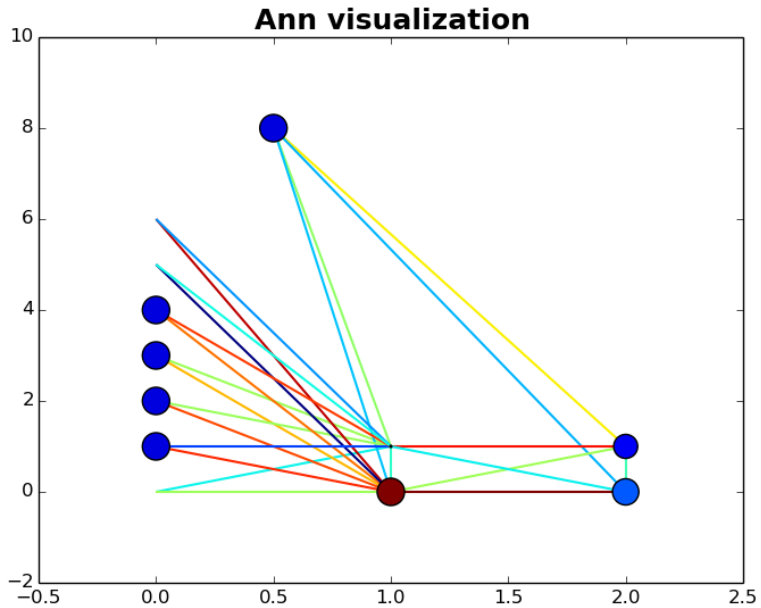


Figure 2.4: Activation pattern in a fully interlinked neural network with seven input nodes (right), one bias neuron (upper node), two interlinked hidden neurons (middle, brown) and two output nodes (right). The radii of the circles are proportional to the neuron output. Blue links are repressing links, red and brown links are activating links.

Over-training

The peculiarity when dealing with machine learning is to downgrade ones excitement about 100% accuracy predictions. A predictor of very low training error might be completely wrong for new data sets because it might have lost its essential generalisation ability. Loosing this ability is often referred to as "over-training". Irrelevant signatures like noise in the data have been learned that are not relevant for prediction. During the iterative training process, the ANN learns the rough data structure first. With each iteration the error surface is explored in more detail. First the error is reduced by taking into account characteristics of the two groups - later the training instances are incorporated as individuals. In this stage the fine structure including all the noise is learned.
An impressive example for over-training is shown in figure 2.5b where the training errors goes down significantly but the classification errors of separate validation instances do not at all. The example illustrates the need to stop training before both error types separate. Figure 2.5a shows the error development of a successful training process. If the errors of the training and validation sets are roughly in the same range, over-fitting has not occurred yet and the network still preserves its generalization ability. When the error of the training sets are significantly lower it is already too late.

A precondition for over-training is a complex network structure that has the ability to incorporate complex fine structure. A high number of free model parameters promotes over-fitting. This means that a simple ANN structure that is capable to map the input to the output should be preferred to a more complex one. This gives rise to the question, how the optimal network complexity, i.e. the number of hidden nodes should be determined.
Though some formulas exist that recommend the total number of hidden nodes, the choice is highly data specific and no rule of thumb can make the decision for the researcher. The dimensionality of the input data determines the input data complexity. Likewise the number of hidden neurons directly affects the networks potential to have more complex input output mappings. Hence, both parameters have to be chosen dependently. Thus it seems to be reasonable to conduct a two dimensional parameter scan and measure prediction performance for different combinations of input and network complexity. This performance surface will hopefully enable better understanding about hidden noise and nonlinearity in the data.

Cross-validation

Cross-validation enables to indicate over-training. Cross validation is a powerful tool because it is compatible with all predictors and can compare them as long they use the same data [15]. The set of available instances is partitioned into a training and a validation subset. The validation set serves as a control group for the learning process and is not used for training. During the training process the mean square errors are monitored for both subsets separately.
Cross validation is justified if the subsets are large enough so that they can represent the whole data [40]. However the precondition is not always guaranteed. This is why the training process is repeated with different partitions of the original data to enable a statistical error analysis and to be independent from randomly biased subsets.
The ratio that is used to split the instance space into training and validation subspace

can be chosen freely. The ratio used here was also successfully tried by Khan et al. [23]. The training set contains 2/3 of all data sets and the validation set 1/3 because this is expected to give reliable validation results while still providing a representative training set.

Output decoding and accuracy estimation

The individually trained neural network predictors have different input-output mappings. There are various ways to reach a consensus scheme. One option is to make a pre-selection so that only the best neural networks ANNs contribute to the final vote [7, 38]. However this rather greedy approach might promote over-training because not all data contributes to an equal amount [15]. Generalization abilities of a broad ensemble tend to be higher than that of few, very good performing ANNs.
In a consensus scheme the individual votes are combined to a democratic vote. This strategy was proved to be successful e.g. in Khan et al.'s paper [23] where 3750 ANNs were trained and used as an ensemble predictor. Hansen and Salamon [15] studied the behaviour of an ANN voting committee. The residual error decreases if each ANN has a correct prediction in more of half the cases.

The continuous output $O \in [1, -1]$ needs to be converted to a binary value in order to become a classification result. However, the continuous output indicates the certainty of the prediction. Accuracy can be described by the level of agreement of the single predictors. The standard deviation of the single ANN outputs seems to be a reasonable indicator of accuracy. Nevertheless, one should keep in mind that prediction errors cannot be lower than the validation errors measured during cross validation.
Although the accuracy proposed seems to be a reasonable quantity, this strategy is interestingly not considered in the literature above. In the literature usually confidence intervals are chosen for a single predictor's output that marks a prediction as certain or uncertain [23, 38]. However, the non digitalised mean regression output and the regression standard deviation are expected to offer more detailed, instance specific information.

2.3 Postprocessing

2.3.1 Receiver operating characteristic (ROC)

Since machine learning is prone to over-fitting, reliable monitoring and validation tools are necessary to critically assess prediction accuracy. Receiver operating characteristics (ROC) is introduced as validation technique. It provides a definition of accuracy. Metz [28] proposed to compute the following quantities:

TP = No. of true positives correctly classified

TN = No. of true negatives correctly classified

FP = No. of false positive classifications

FN = No. of false nagative classifications

P = No. of positive instances

N = No. of negative instances

Sensitivity = TP / (P + N)

Specificity = TN / (P + N)

False Positive fraction = FP / N

False Negative fraction = FN / P

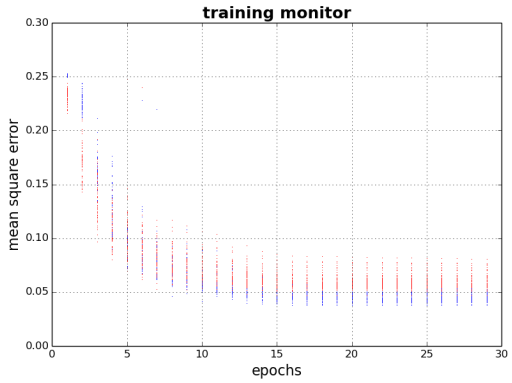
Accuracy = Sensitivity · N / (P + N) + Specificity · P / (P + N)

Another helpful measure of accuracy is given by Matthews correlation coefficient.

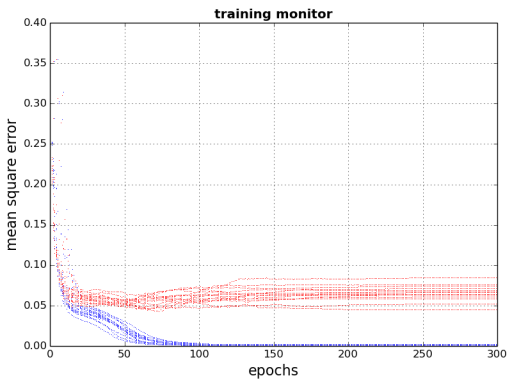
$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{2.7}$$

The MC coefficient is a value in the range [-1, 1] and encodes the transition of perfect disagreement (-1) via random classification (0) to perfect match (1).

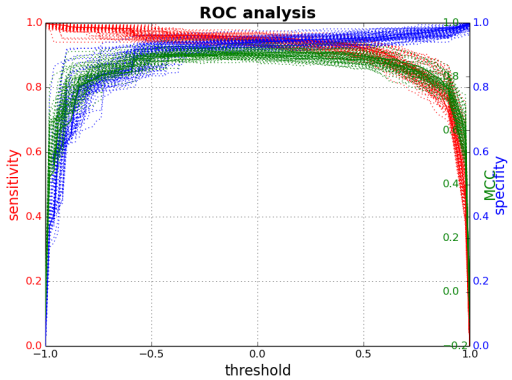
The quantities depend on the threshold that is used for discretion of the originally continuous regression result. The robustness of the classifier is studied by scanning the threshold and plotting sensitivity against specificity. The performances of the individual ANNs and of the ensemble can be compared in these "ROC curves" [7]. The individual predictors are validated with their validation instances only to avoid bias. During validation the prediction result of the ensemble is defined by the mean output of all single predictors that do not have the particular instance in their training set.
Furthermore, the integral over the MC coefficients is interpreted as a performance score. The MCC integral is normalised so that a perfect predictor achieves an integral of 1. This is achieved by dividing the integral by the range of the thresholds.
An example for a good and a bad training result is shown in figure 2.5a. A bad predictor is characterised by a linear relationship between specificity and sensitivity. There exists no discrimination threshold that has both, good specificity and sensitivity. A good predictor has maximal curvature and ideally reaches a sensitivity and specificity of one for the same threshold.



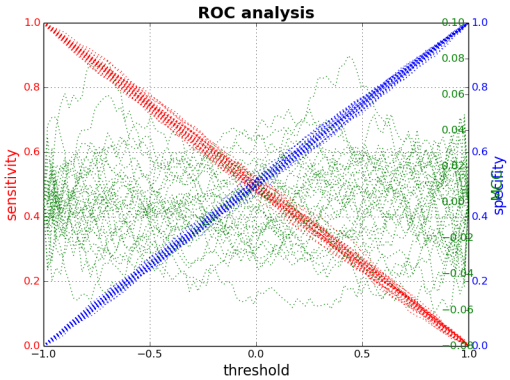
(a) Successful training process for a number of neural networks. The blue points show the errors of the training data and the red ones relate to the validation sets.



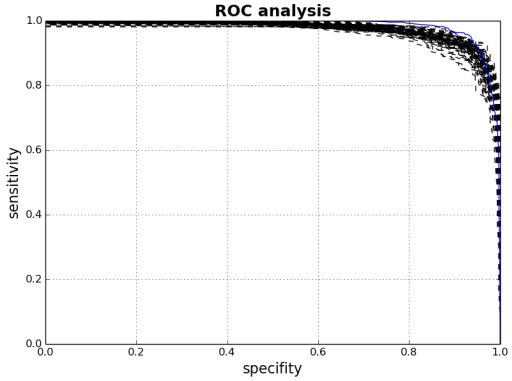
(b) Example of over-training. The errors of the training set go down while the errors of the validation set go up or stay constant.



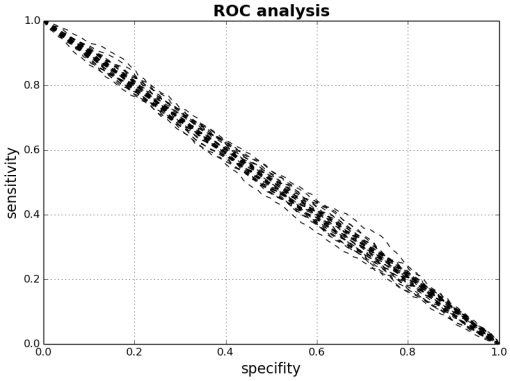
(c) ROC analysis for a reasonably good predictor. Discrimination thresholds are shown on the abscissa. Sensitivity (red) and specificity (blue) are shown on the ordinates. The MC coefficients (green) are high for many thresholds.



(d) ROC analysis with shuffled input data. Straight lines of sensitivity (red) and specificity(blue) indicate a random classification behaviour. The MCC curve (green) has weak fluctuation around zero.



(e) ROC analysis for a reasonably good predictor in another representation. Sensitivity and specificity are plotted against each other. There are thresholds with both, high sensitivity and specificity.



(f) ROC analysis with shuffled input data. Straight lines indicate that there is no discrimination threshold with good sensitivity and specificity at the same time.

Figure 2.5: Examples for good (left) and bad (right) training results.

2.3.2 Sensitivity analysis

In order to provide more evidence for variable relevance estimation, a sensitivity analysis is performed. By this input feature relevance assessment, the descriptive data can be further reduced and possibly prediction performance improves. Both strategies belong to the groups of backward selection processes because they interpret feature relevance in the context of all other features. Two different approaches are selected from the literature: The first one tests the response of prediction performance when an input feature is removed from the descriptive data in a leave-one-out scheme. The other option consists of a finite difference approach. The change in prediction output is computed for the variation of one input feature.
The resulting sensitivity score should not be confused with the sensitivity that is computed in ROC.

Performance sensitivity Szaleniec [40] suggests to remove one of the features from the data and redo the training process. This corresponds to a leave-one-out formula and is actually a wrapper technique for variable selection. The relevance of a variable is then defined by the decrease of classification accuracy when it is left out.

Finite difference approach In the paper of Khan et al. [23] the sensitivity of input variable k is assessed by the finite gradient:

$$S_k = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{\partial o_i}{\partial x_k}$$

with the number of instances N_s, output variation o_i and variable variation x_k. This simple sensitivity measurement reveals to what extend the network relies on each input. The sign of the quantity tells about a suppressive or promoting dependence between the variable and the key feature.

2.4 Summary, Proposal

The dimensionality is reduced during preprocessing in a two step procedure including variable selection and feature extraction. For each step four options have been proposed (see figure 1.1 on page 7). The eight methods are tested and compared to each other in chapter 4. The final preprocessing procedure is subsequently decided on the basis of this analysis. A collection of criteria structures the discussion, including: a) explanation and interpretation power, b) suitability for dimensional reduction and preliminary training results if available, c) preservation of nonlinear information, d) arguable choice of parameters and e) computational feasibility.
In addition, the data is enriched with nonlinear interaction terms by separability analysis.

An artificial neural network approach is a promising, robust and adaptive tool for machine learning. Training parameters are optimised including the number of input features and the number of hidden neurons used in the network. This is achieved by scanning the two parameters and monitoring prediction accuracy. The number of training epochs is chosen by monitoring training and validation errors. Cross-validation and receiver operating characteristic (ROC) are applied to detect and

minimise over-training. The integral of the Matthews correlation coefficients (MCCs) serves as a measure of predictor performance.
All individual predictors that are obtained from cross-validation vote in an ensemble. There is no oligarchy as no neural networks are pruned. The individual votes are averaged. Both the continuous regression vote and the standard deviation convey the certainty of the prediction and can be used as instance specific accuracy estimate.

Two variations of sensitivity analysis are implemented in order to provide more evidence for variable relevance estimation. Possibly the results can be used to further reduce the dimensionality of the problem and further improve prediction accuracy.

Finally, the biological results are discussed and candidate instances are classified in chapter 5.

Chapter 3

Data Retrieval

Data retrieval is a pattern creation process that determines the epistemic limitations of the subsequent analysis. In this chapter protein and descriptor space selection is thoroughly argued and documented. The resulting data will be subsequently reduced in dimensionality and used for machine learning to predict specific DNA-binding RNA polymerase II transcription factors (DbTFs).

Gene Ontology (GO)

In information technology various knowledge representations have been invented to store and organise complex forms of data. The word ontology has its origin in the Greek word 'onto' which translates to 'being', bravely implying that the ontology aims at a description of the world. In an ontology information is compartmentalised into classes. These classes, or terms, are ordered in a hierarchical tree structure and connected by semantic relationships. These relationships encode for example 'is a' or 'is part of' dependencies. In the particular use case the terms are gene ontology terms (GO terms) that represent diverse protein properties, for instance, the involvement in biological processes, having a molecular functionality or being located in a cellular compartment or a complex. Proteins are either annotated to terms or not and can thus be represented as binary vectors.
One advantage of the ontology approach is the variety of complex information it can handle. Inference and logical reasoning algorithms can be applied because the computer 'understands' the data thanks to the hierarchical tree structure and the semantic connections. For example, members of subclasses inherit the memberships of all upper classes. However, inheritance is expected to increase collinearity and redundancy in the data.

The gene ontology consortium (GOC) [2] is well curated and has public availability. It encodes specific DNA-binding RNA polymerase II transcription factors with the GO term 'GO:0000981' in the branch 'regulation of gene expression'. On 09.02.2016 the consortium provided 109104 inferred or experimentally validated DbTFs in total, about 38 million gene products and about 40 thousand terms.

Domain of classification

The domain of classification includes 2325 human, 2119 mouse and 1601 rat gene candidates from the TFcheckpoint database [5].These candidates are assumed to have transcription factor activity or at least interaction with them, however experimental evidence is still missing. Based on the assumption that transcription factors of the human and the two other species mouse and rat function in a biologically comparable way, the orthologs can be combined in order to enlarge the training set. 2655 unique candidate proteins are found to have annotations in GOC. Only 144 of these are inferred DbTFs according to GOC. The project focuses on classifying the domain of 2655 candidate proteins.

Choice of true positives

The list of true positives contains proteins that are known to be DbTFs a priori. The TFcheckpoint database provides a literature-based collection of 818 human, 796 mouse and 679 rat transcription factor genes. These DbTFs have been manually checked for experimental evidence [5]. The intersection of the experimentally verified DbTFs from human, mouse and rat from TFcheckpoint and the DbTFs from GOC constitutes a set of 952 DbTFs.

Choice of true negatives

An important aspect of true negative selection is to balance dissimilarity to the true positives and similarity to the candidates. The negatives should be no DbTFs to a high degree of certainty and they also should be involved in biological processes that are related to the candidates. The challenge can be described with an example: Proteins, that are not located inside the nucleus and that are not annotated in the gene expression regulation branch, have a high probability to be no DbTFs. However, most of the candidate proteins are located inside the nucleus and they are usually annotated in the gene regulation expression branch. Most of the candidates will be classified as DbTFs. This criterion for negative instance selection is not optimal, because the data is biased. One will not learn new aspects about the candidate proteins, because they are already quite similar to DbTFs.

The choice of true negatives is conducted in two steps. In a discrimination step a criterion is designed to identify non DbTFs. In the second step the domain of negatives is further reduced. A criterion needs to be found that defines similarity to the candidate proteins.

In the first step, the 952 true positives are analysed to find a criterion to identify the negatives. Some GO terms are particularly more often annotated to the true positives than to 952 random human proteins extracted from GOC. The GO term list was ranked by correlation to the key feature, i.e. being a DbTF. The list was studied manually and a selection of DbTF related terms was made. The list includes the complete branches that start with the GO terms "GO:0000981" describing DbTFs, "transcription factor activity", "RNA polymerase II transcription factor binding", "sequence-specific DNA binding" and "sequence-specific double-stranded DNA binding". Proteins, that are not annotated to any term in these branches, are

treated as negatives. As the true positive and candidate instances are from human, mouse and rat, the negatives are also restricted to this domain. Negatives, that are also candidates, are removed.

About 20000 proteins remain. The second selection step includes a similarity assessment between the negatives and the domain of classification. The similarity analysis is based on information independent from the actual training data, in order to keep the training data unbiased. Hence, a variable subspace V is defined that will later be excluded from the training data.
V combines the GO term branches beginning with the nodes "regulation of gene expression", "nucleic acid binding", "core DNA-dependent RNA polymerase binding promoter specificity activity", "transcription factor activity", "core RNA polymerase I binding" and "transcription, DNA-templated". These branches were chosen manually because they are assumed to be indirectly related to DbTFs and should not be used as a basis for candidate prediction. V contains about 900 GO terms.
The true negatives are ranked according to their similarity to the candidates in the subspace V. A similarity measure is proposed. Let D_i be a vector containing the euclidean distances between the true negative i and all candidate proteins. Let d be the effective distance. $d = \bar{D_i}^{10}$ denotes the mean euclidean distance to the ten closest candidate instances. This definition considers the local environment of the candidate proteins in V. The effective distances of the about 20000 negatives are computed and shown in figure 3.1. About 10000 of the negative instances are able to represent at least ten candidates. The combination of these negatives and the positives constitutes the instance selection for the training data.

Although the final set of true negatives is chosen with great care, it cannot be proven that it does not contain any DbTFs. In order to give potential DbTFs that are overlooked a minimal impact, a large number of about 10000 potential negatives is chosen. This strategy works well if the assumption holds that the number of unknown DbTFs is not much higher than the number of known DbTFs. Based on this assumption the percentage of false non DbTFs will be low. The excess of negative instances is expected to support the validation process. Machine learning needs an equal number of true positive and true negative instances, so the individual classifiers are trained with random bootstraps from the large pool of negatives and the smaller pool of true positives. In addition, there are more validation instances than training instances and over-training is easier to detect. As all data is used, the ensemble is expected to have better generalisation abilities.

Obtaining the training data

The training data was obtained from the GOC database on 09.02.2016. For each protein a binary vector is extracted that encodes the existence or non existence of an annotation. Both, experimentally verified and computationally inferred annotations are included. The variable subspace V is excluded from the training data. The dimensionality of the resulting data spans 11023 GO terms.

The data is checked for a potential bias between true positives and true negatives. In figure 3.4 the standard deviation of each variable is shown. The standard deviations are computed separately for the DbTFs and for the non DbTFs. The true positive data tends to have a higher variation. The difference is consistent, but very small. It could not be clarified, if this is a justified difference between DbTFs and non

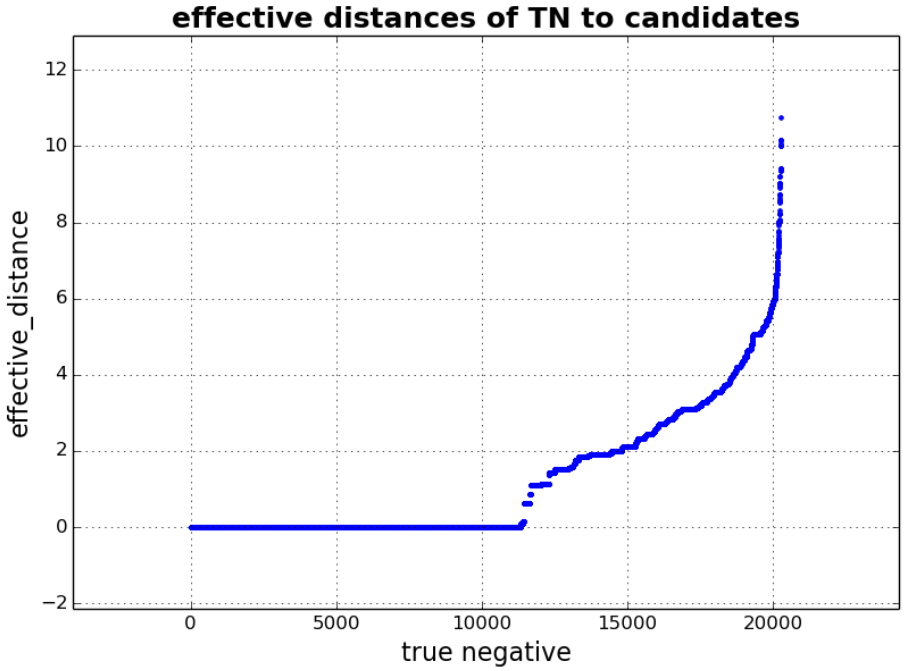


Figure 3.1: The effective distances between the true negative instances (abscissa) and the candidates are computed in the variable subspace V. About half of the negatives represent the candidates well in V.

DbTFs or if sparsity is lower for DbTFs, only because there has been more research conducted about them.

Later in the project, there was some concern over the 'biological process' sub-ontology. Due to the way this sub-ontology is constructed, it might be indirectly influenced by knowledge about transcription factors, though the subspace V is excluded from the training data. In addition, the terms do not describe 'objective' physical or chemical properties. Instead, they describe involvement in abstract biological processes and incorporate a high level of interpretation. There was the idea to focus on terms related to molecular function and cellular component information. However, removing the biological process sub-ontology lowered dimensionality from about 11000 to about 2500 and lead to considerably worse training results. The classification performance is assessed on page 51. The decision was in favour of the biological process sub-ontology, less sparsity, higher prediction accuracy and potential synergistic effects between the different sub-ontologies.

Create three subsets for cross checking

Having an excessive supply of true negative instances enables selection of multiple subsets for stability analysis. The subsets are presented in table 3.1 and are motivated in the following. The original data set is referred to as X_{orig}.

The subset X_{rand} contains all 952 true positives and a random selection of 952 true negatives. It is chosen as a control group that probably represents the characteristics of the original data set X_{orig}. In figure 3.3 the number of annotations are shown for each protein. X_{rand} is by far the most sparse subset. In figure 3.2 the number of annotations with experimental evidence code is shown for each instance. One observes, that the data set usually contains computationally inferred annotations.

Data set	No. of true positives	No. of true negatives	Criterion for true negative selection
X_{orig}	952	9123	representativeness of candidate proteins in the extended gene regulation branch
X_{rand}	952	952	random selection
X_{evid}	952	952	prefer proteins with many experimentally validated annotations
X_{anno}	952	952	prefer proteins with many experimental or computationally inferred annotations

Table 3.1: The three smaller data sets are subsets of X_{orig} that are chosen by certain criteria.

Another data set X_{anno} is chosen to have minimal sparsity. All true positives are included, as well as 952 true negatives with the highest number of experimental or inferred annotations. This choice bases on the assumption that a less sparse data set could lead to better prediction performance. The low sparsity is visible in figure 3.3.

In order to have a subset with reliable annotations, X_{evid} is constructed. It contains all true positives and 952 true negatives with predominantly experimentally validated annotations. This can be confirmed in figure 3.2, where the numbers of annotations with experimental evidence code are illustrated. Figure 3.3 implies that X_{evid} contains more information than X_{rand}.

Details on X_{evid} An important concept in the assessment of GO term annotations is the 'evidence code'. A detailed description of the evidence codes can be found on the gene ontology website `www.geneontology.org`. All protein's annotations have an evidence qualifier attached. The qualifiers can be divided into experimental evidence and computationally inferred evidence. In general, the experimental evidence codes represent the strongest form of evidence. Based on this assumption an evidence review is performed. The experimental annotations include:

Inferred from Experiment (EXP);

Inferred from Direct Assay (IDA);

Inferred from Physical Interaction (IPI);

Inferred from Mutant Phenotype (IMP);

Inferred from Genetic Interaction (IGI) and

Inferred from Expression Pattern (IEP).

In figure 3.2 the occurrences of experimental evidence are shown. The majority of proteins has no experimental evidence at all, so only an extract from the whole diagram is shown.

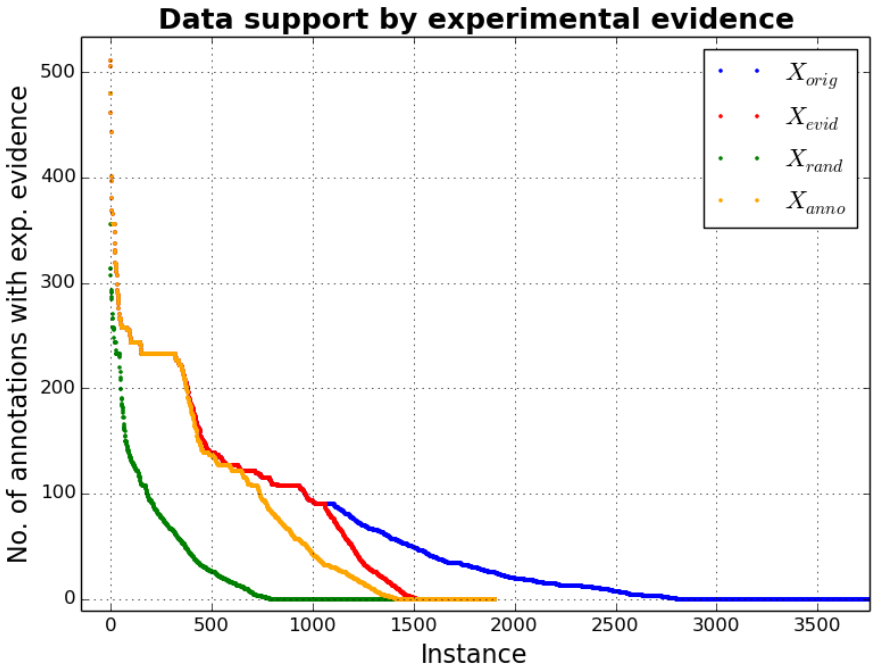


Figure 3.2: The diagram shows the number of associations with experimental evidence for true positive and true negative samples. The instances are sorted for each subset respectively. Most instances have no experimental validated annotations at all, but computationally inferred ones.

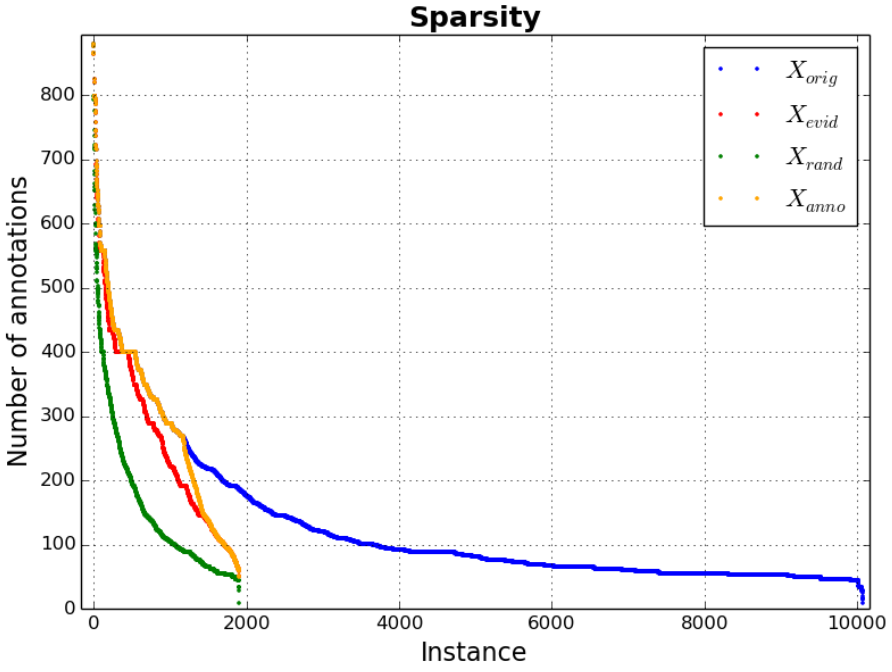


Figure 3.3: The number of annotations are counted for the true positives and the true negatives. The instances are sorted for each subset respectively. X_{rand} is the least sparse subset.

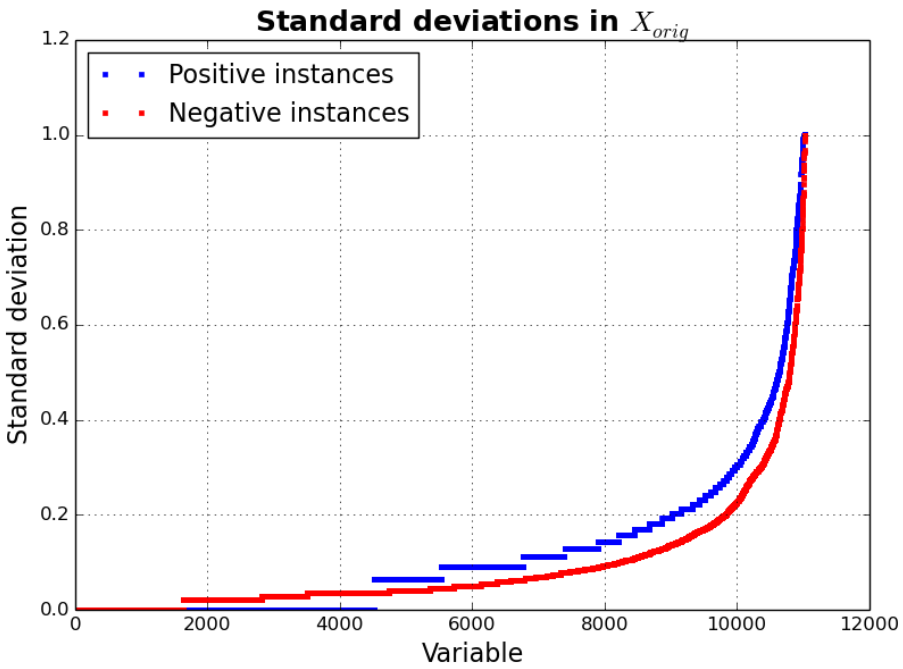


Figure 3.4: The standard deviation is computed for each variable for the set of true positives and negatives separately. The variables are sorted for each subset respectively. The true positive data is slightly less sparse than the true negative data.

Chapter 4

Results and Discussion, Methodology Development

In the chapter 2 a methodology is proposed for the classification of specific DNA binding transcription factors (DbTFs). Figure 1.1 on page 7 illustrates the proposal. Corresponding multivariate data is extracted in chapter 3 and four subsets are built in order to enable stability checks. These subsets are summarised in table 3.1 on page 31.
The methods proposed in chapter 2 are applied on the data sets. Parameters are optimised and the results are discussed. Based on the discussion a problem specific choice of methods is obtained and summarised on page 56. The results of variable selection, feature extraction and postprocessing are discussed separately to enhance readability.

4.1 Development of variable selection

The literature review in the methodology chapter 2 resulted in a choice of four different variable selection methods including variance, separability analysis, correlation coefficients and PLSR coefficients. A list of criteria structures the discussion. The criteria for the assessment are a) explanation and interpretation power, b) suitability for dimensional reduction, c) preservation of nonlinear information, d) arguable choice of parameters and e) computational feasibility.

4.1.1 Standard deviation

For each variable and for each subset the standard deviation is computed and shown in figure 4.1. Variables are sorted according to the standard deviation of the original data set X_{orig}. In the following the criteria a) - e) are discussed based on this diagram.

a) The subsets are compared to each other and their information content is estimated. All distributions have a long tail consisting of rather constant variables that add only few information. The true negative instance selection has a strong influence on the variance in the data, because the subsets have different variation. The two data sets X_{evid} and X_{anno} tend to have higher variation. The standard deviations are able to show that the data sets X_{evid} and X_{anno} are biased.
The subset X_{rand} represents the original set X_{orig} better than the other subsets,

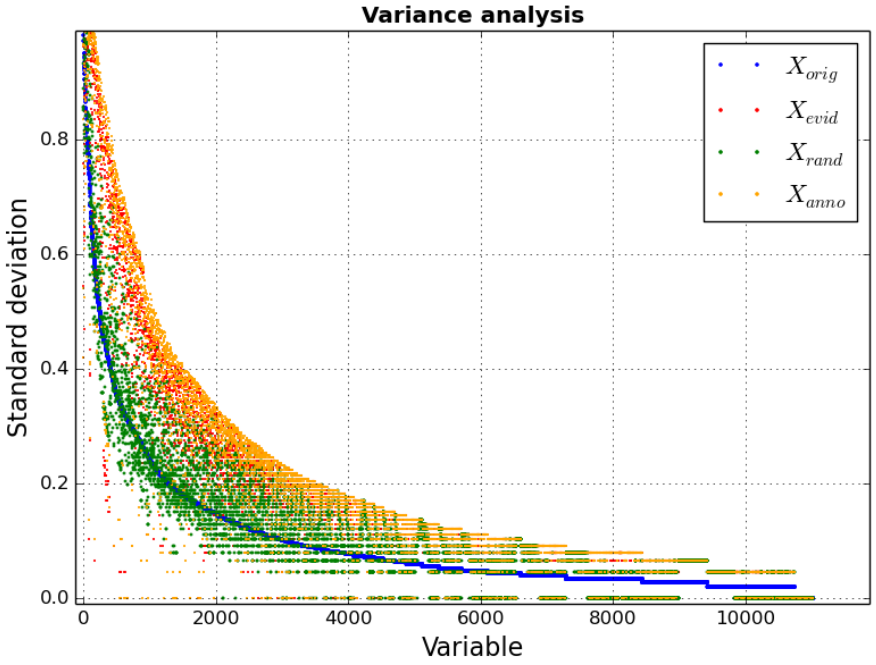


Figure 4.1: The standard deviation of each variable is shown. The variables are sorted according to the standard deviation of the set X_{orig}. Subset selection has a strong influence on the variance in the data.

because it does not have a significant offset. This finding is in line with the original idea, since the true negatives are drawn randomly from X_{orig}. However there is strong fluctuation in X_{rand}. There are two possible conclusions to draw: On one hand the fluctuation could mean, that the true negatives are very diverse and that there is a high probability, that some true negatives are able to represent the candidates. However, there is also the negative way to interpret the fluctuations: the sparsity of the data might prevent reasonable statistics.

b) Standard deviation neglects variable interaction and is a completely unsupervised approach. Hence it does not offer a reasonable variable relevance estimate. Nevertheless, standard deviation can be used for detecting the least informative variables and removing them.

c) If variance analysis is used to remove rather constant variables only, the risk of loosing nonlinear content will be low.

d) As no potentially relevant information should be removed from the data, a low threshold of 0.1 is chosen. Variables with standard deviation 0.1 have only approximately 0.25 percent non-zero (or non-one) elements. Nevertheless dimensionality is about halved, which is a big improvement.

e) There is no particular computational challenge to mention.

4.1.2 Separability analysis and data enrichment with interaction terms

Separability analysis is in detail explained in chapter 2.1.1 on page 11. The separability scores are computed for each variable separately and shown in figure 4.2.

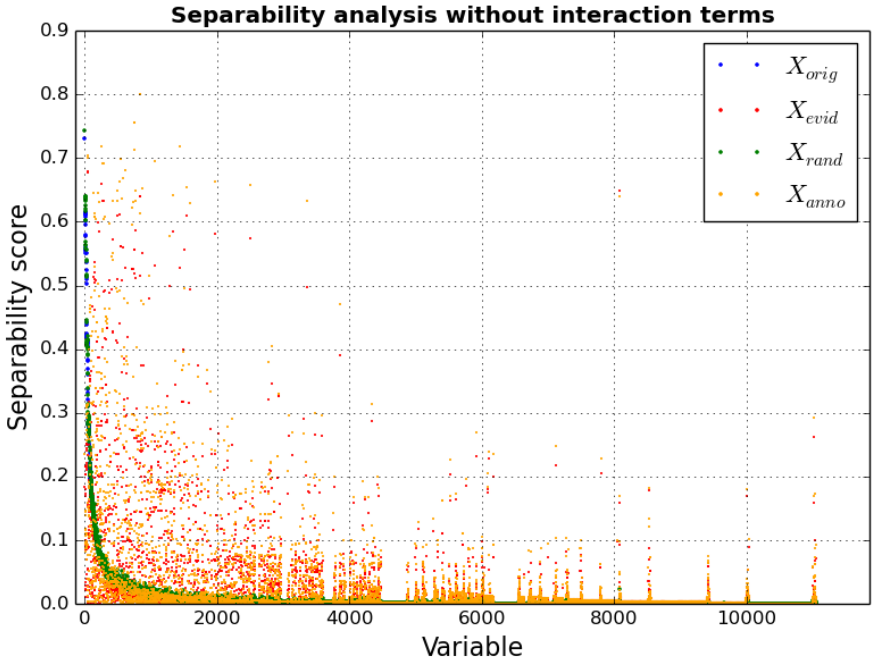


Figure 4.2: Separability analysis is performed and interaction terms are neglected. Variables are sorted according to the separability scores using the data set X_{orig}. The separability scores are invariant under random instance selection and balance the excess of true negatives in X_{orig}

a) The algorithm balances the different numbers of true negatives and true positives in the data set X_{orig}. Indeed, X_{orig} and X_{rand} overlap. The similarity of the curves is a hint that random instance selection does not bias the data. This is an essential observation because it means that reasonable statistics can be done in spite of the sparsity.
For a binary variable separability score is found to be proportional to the absolute covariance between the variable and the key feature. This is not necessarily the case for continuous variables. We will see, that there is no overlap in the case of correlation coefficients and for covariance, which are directly affected by an excess of true negatives.
The other two data sets show a different behaviour. Some variables with low separability score in X_{orig} and X_{rand} have much higher values in X_{evid} or X_{anno}. This means that these two subsets could achieve lower training errors during machine learning.

b) The ability to detect nonlinear interaction terms is a clear advantage for prediction. Separability analysis seems to be a helpful, model free instrument for variable relevance assessment. However, collinearity is not considered. A variable subset created with separability analysis might contain redundancy. Probably, the tool is better suited for PCA or PLS component selection, because these components are statistically independent from each other.

c) Separability analysis detects nonlinear point distributions in one variable. For binary data this is obviously irrelevant. However, the method can be expanded and relevant nonlinear variable-variable interaction terms can be identified. The procedure is in detail described in chapter 2.1.1 on page 11.

Not all pairwise combinations can be checked for interaction because of computational limitations. A pre-selection of promising variable pairs is made. The

search is conducted with the 500 variables of highest standard deviation in order to remove the long tail of rather constant variables.
Subsequently, the set of variable pairs is further reduced. Variable pairs with a high correlation coefficient are expected to have a linear relationship and might not show interesting interaction. Based on this assumption, the variables are clustered with the help of a correlation distance matrix. The idea is to select one variable in each cluster and obtain a set of non redundant variables.
The distance matrix is defined by one minus the correlation matrix. A hierarchical clustering algorithm is applied. The complete metric is used that defines the distance between two groups u and v as $d(u, v) = \max(d(u[i], v[j]))$ with the euclidean metric d. The metric is supposed to make the groups as uncorrelated as possible. A threshold determines the maximal distance between two groups u and v, that can be merged together. The number of clusters and their sizes depend on the choice of this parameter. The threshold is scanned and the clustering results are shown in figure 4.3. The number of groups and the variance of their sizes are shown. One would like to have a balance between the number of groups and the variation of group size. A too low threshold results in many groups that are too correlated to each other. A too high threshold leads to few clusters, which incorporate the majority of variables and much information is lost. A threshold of 0.6 seems to be a good choice, because it is the highest threshold before group size variance increases and super-clusters form. For each group a leader is selected that maximises standard deviation in the group. Only these leaders are considered in the interaction assessment.

After the computation of the separability scores, variable pairs with a separability score higher than 0.5 are kept and eight interaction terms remain. For these variable pairs the optimal logical operations are identified. The logical modes are: x_0 AND x_1, x_0 OR x_1, x_0 AND $\neg x_1$, $\neg x_0$ AND x_1, x_0 XOR x_1. Each logical operation is applied on each pair and the one-dimensional separability scores are computed. The logical operation with highest one-dimensional separability score is chosen and the eight interaction terms are determined.
The new terms are presented in figure 4.4. In the diagram the separability scores of the interaction terms are shown in red and the original variables have a blue color. The eight interaction terms have a higher ranking than many other original variables. These interaction terms will be used for the final training data.

d) There is no free parameter to choose for binary data. Otherwise, the grid size for histogram computation has to be determined. In the methodology section it is argued to choose the grid size according to the scale of the structures that one would like to capture in the data. One should avoid high ratios of bins and instances (like more than one percent) in order to assure, that there are enough instances in each bin to enable reasonable statistics. If six bins are chosen for about 1900 instances, the ratio is about 0.003. Thus, separability analysis can be applied, for instance, to select statistically independent components from PCA or PLS.

e) The computation is efficient if variable interaction is neglected. If variable combinations are considered, the combinatorial space explodes. In step c) variables with high standard deviation and pairs with low covariance are checked for interaction. These seem to be reasonable criteria because only informative variables are tried and no time is wasted on linear relationships.

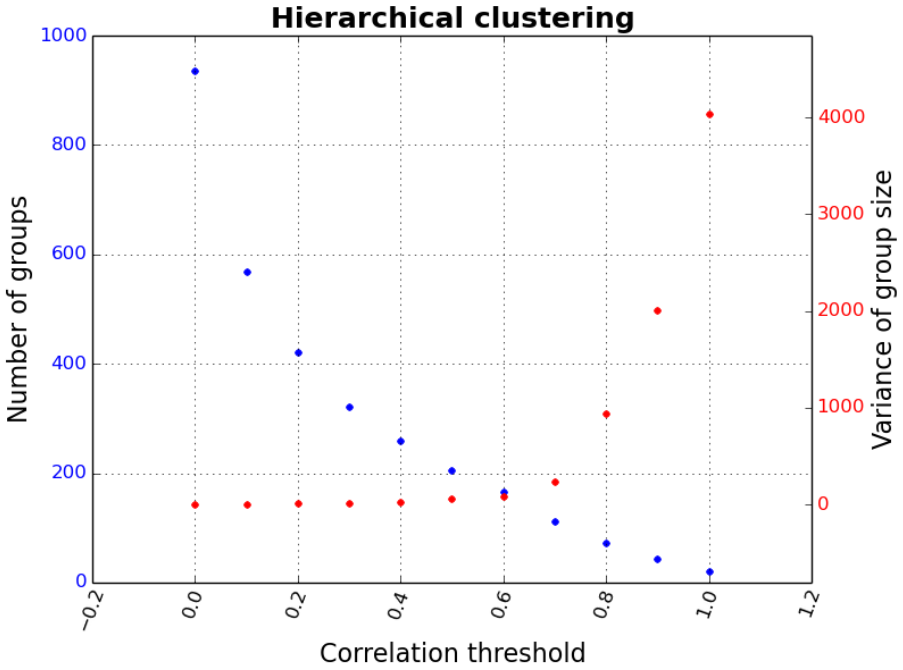


Figure 4.3: Hierarchical clustering is applied on the correlation distance matrix of X_{orig} in order to select a subset of non-redundant variables. The threshold for cluster creation is scanned. The number of groups and the variance of their sizes are shown.

4.1.3 Correlation to the key feature

The correlation coefficients between the variables and the key feature are presented in figure 4.5. Correlation is covariance divided by the standard deviation.

$$cor(a,b) = \frac{1}{n-1}\frac{1}{\sigma_a \sigma_b}\sum_{i=1}^{N}(a_i - \bar{a})(b_i - \bar{b}) \tag{4.1}$$

a) The correlation diagram shows much fluctuation that seems to be stronger for the subsets X_{evid} and X_{anno}. The observation that the fluctuations dominate for negative correlation is a sign for the biased selection of true negatives in these sets. If the correlation coefficient is positive, a true negative must have a predominantly negative value for this variable. As X_{evid} is less sparse for the true negatives, the positive correlation coefficients have a lower scale than the negative coefficients.

b) There are some arguments against using correlation coefficients for variable selection.
First, correlation between the descriptive variables is ignored.
Moreover, it is found that the correlation coefficients between the variables and the key feature are affected by random instance selection. The correlation coefficients of the data sets X_{orig} and X_{rand} do not overlap, although X_{rand} contains the same true positives and a random selection of true negatives. The correlation coefficients seem to be noisy and the coefficients of X_{rand} have a larger scale. This phenomenon also occurs for covariance. The reason for this behaviour is the excess of true negative instances in X_{orig}. For this data set the terms relating to the true negatives contribute less in formula (4.1). The reason is that the key feature value -1 is closer to the mean of the key feature. The higher impact of the true positives cannot counterbalance this effect. Hence,

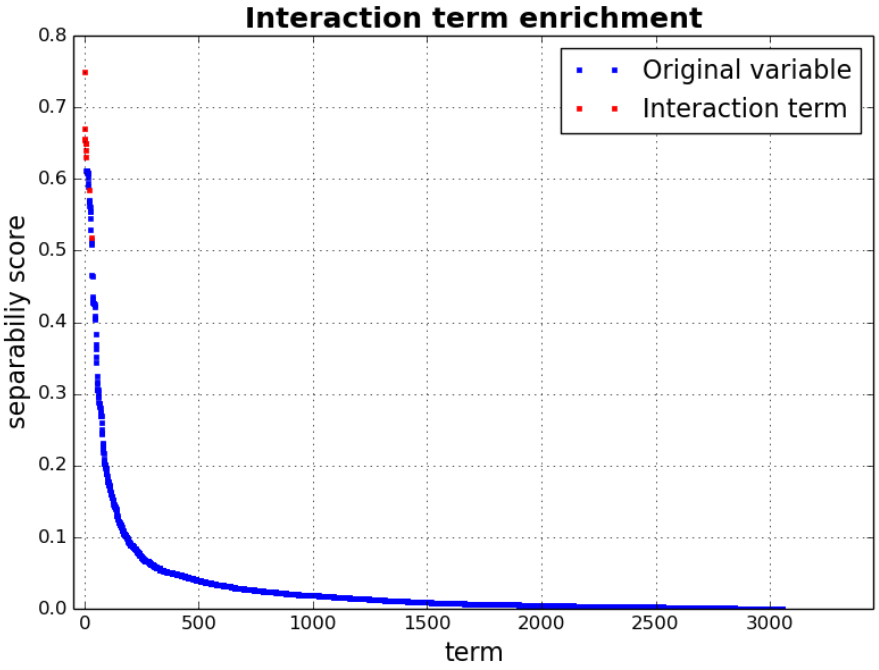


Figure 4.4: Enrichment of X_{orig} with first order interaction terms. The separability scores are computed for both, the original variables and the most important interaction terms, that were found.

the total correlation coefficient decreases. Thus, even random instance selection affects the method.
Another argument is found during comparison of the correlation coefficients with the separability scores. In figure 4.7 the separability score, which is in this case proportional to the covariance, is compared to the correlation coefficients. The about fifty most relevant variables are ranked very similarly by both methods. However for the other variables there is disagreement. Ranking with correlation coefficients promotes sparse variables, because of the division by their low standard deviation in formula (4.1). This type of relevance assessment does not take into account the amount of information that a variable conveys. The separability scores and covariance include this aspect and are therefore better suited for dimensional reduction.

c) Correlation coefficients do not consider nonlinear interactions.

d) There is no parameter in this model to choose.

e) Computation of the correlation coefficients is very feasible.

4.1.4 PLSR regression coefficients

PLSR is applied on the mean-centred normal scores of the data. The PLS regression coefficients are interpreted as relevance estimate. Later, the PLS components will be used for dimensional reduction. In this particular section the PLS regression model is used for variable ranking only and the model is not used for classification. The detection of over-training effects, cross-validation and ROC is performed in chapter 4.4 for the final training data.

a), b) Some interesting observations are made. In the case that only one feature vector is extracted, the linear regression coefficients are proportional to the

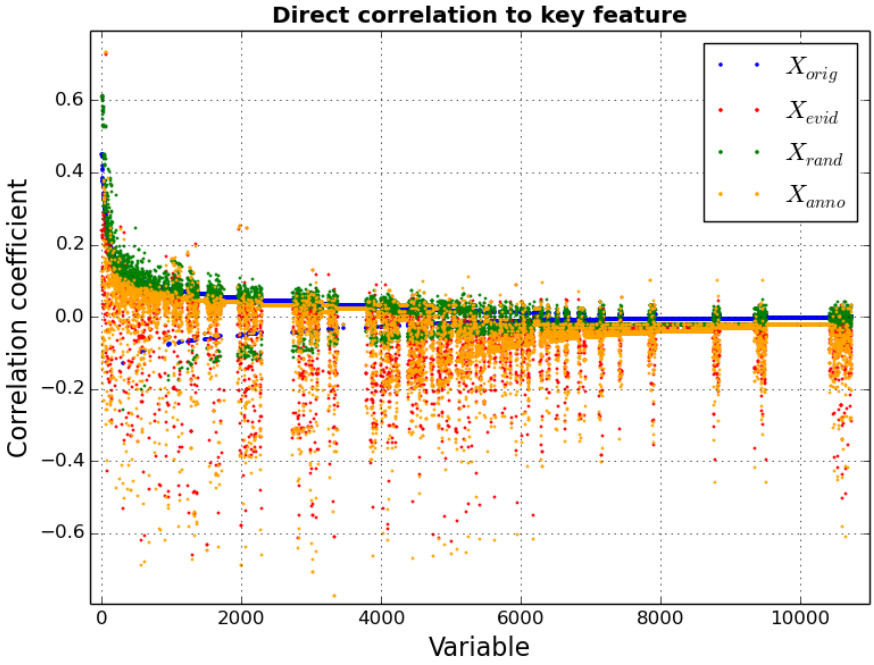


Figure 4.5: Correlation coefficients between the variables and the key feature are computed. The variables are sorted according to the data set X_{orig}. The correlation coefficients are heavily affected by instance selection.

correlation coefficients between the variables and the key feature. If no normal scores are used for normalisation, the linear regression coefficients are proportional to the covariances. Hence, the criteria a) and b) for the correlation coefficients are also valid for PLS.

c) PLS components are linearly independent from each other, but if variables are selected based on regression coefficients, the data will still contain redundancy. In this particular approach the collinearity problem is not solved.

d) The PLS model depends on the number of feature vectors that is extracted. This number has a strong impact on the regression coefficients. For one feature vector the coefficients are proportional to the correlation coefficients. The correlation coefficients are shown in figure 4.5. This figure can be compared to the case that two latent vectors are extracted. The results are shown in figure 4.6. The subsets cannot be distinguished because of strong fluctuations. Apparently, the second feature vector introduces much subset specific noise. The behaviour does not improve with more components, so only the first latent vector is extracted.

e) PLSR is computationally feasible.

4.1.5 Conclusion

Based on the discussion it seems to be reasonable to prune variables with a standard deviation of less than 0.1. Separability analysis, correlation coefficients and PLS regression coefficients are not recommended for variable selection. Variable ranking based on separability analysis suits PCA and PLS components, which have no correlation.
The standard deviation filter is applied on the data sets:

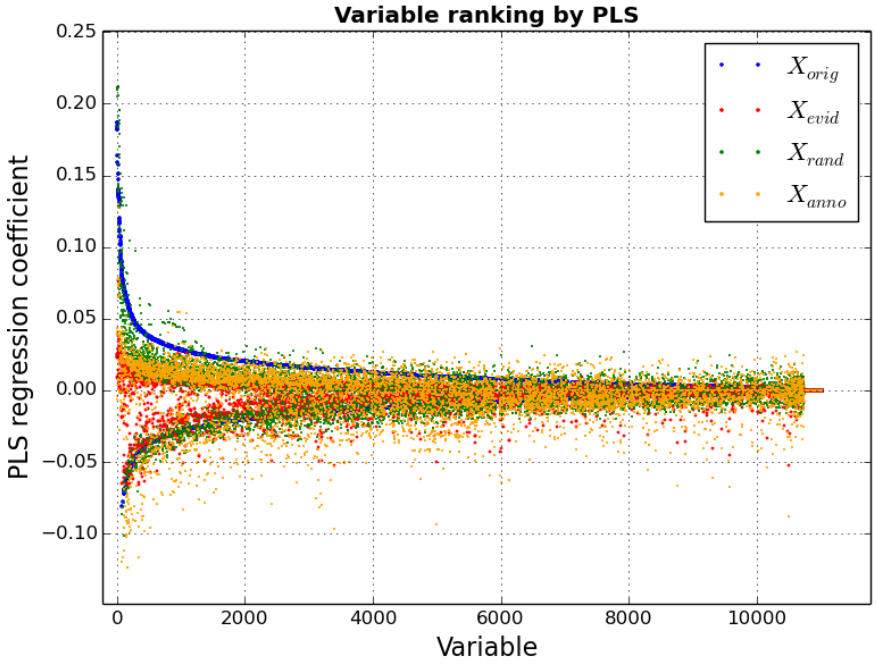


Figure 4.6: PLS is performed and two latent vectors are extracted. The PLS regression coefficients are shown and interpreted as relevance estimate. They are sorted according to the data set X_{orig}. The second component introduces so much noise that the data sets cannot be resolved.

X_{orig} with 10075 proteins and 3047 GO terms, interaction terms not included,

X_{evid} with 1904 proteins and 4972 GO terms,

X_{rand} with 1904 proteins and 4080 GO terms and

X_{anno} with 1904 proteins and 5694 GO terms.

Summary stability check

The observations from variable selection can be used for a stability assessment.
As the two data sets X_{evid} and X_{asso} are both created with criteria related to the number of annotations, their sparsity is lower than for X_{orig} and X_{rand}. The true positives and the true negatives have a biased representation in X_{evid} and X_{asso}. The training errors could be lower for X_{evid} and X_{asso}, but there is the risk that the candidate proteins could be classified on a non biological basis.
The sparsity of X_{rand} raises concern over its potential to resolve the true negatives and the true positives. Its low information content could promote over-training effects.
It seems to be a good idea to train many classifiers with bootstraps of the full data set X_{orig}. The classifiers vote in an ensemble. In this approach all information contributes and over-training can be easier detected due to a larger number of validation instances.

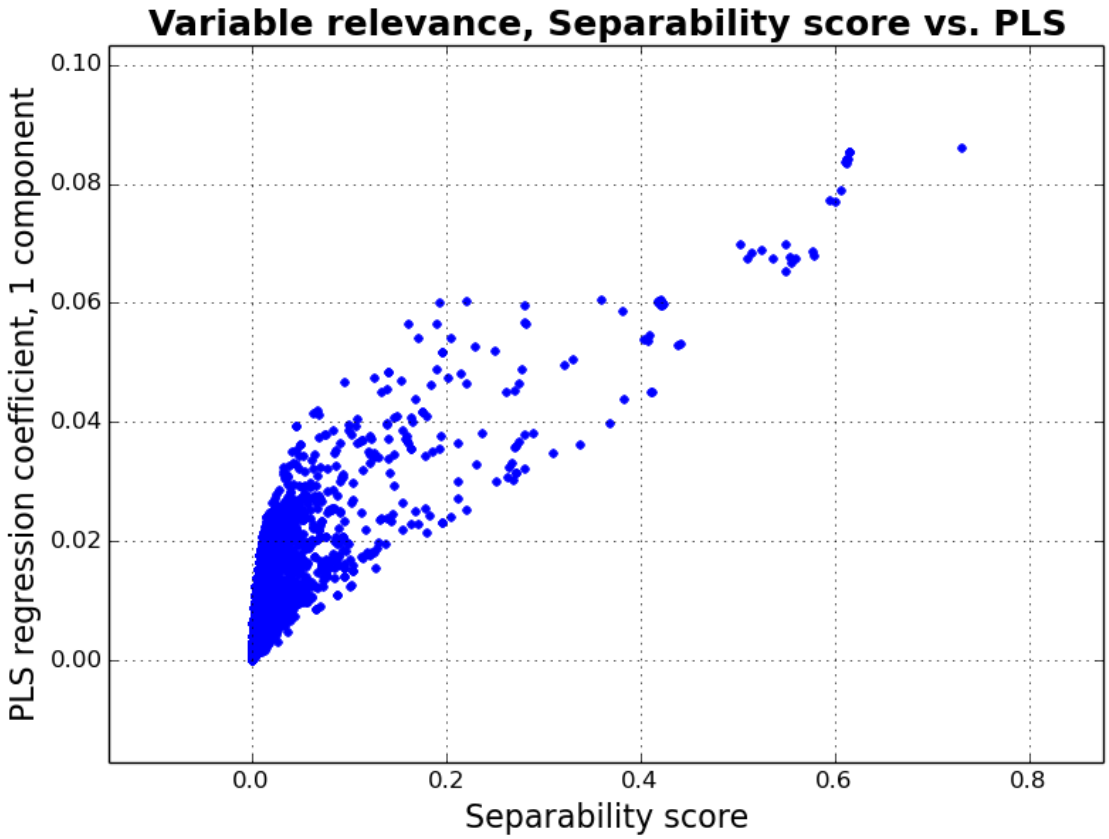


Figure 4.7: One latent vector is extracted and the PLSR model is computed. Separability scores (here proportional to covariance) and absolute PLS regression coefficients are compared to each other for each variable. The regression coefficients promote variables with very low standard deviation.

4.2 Development of feature extraction

The discussion about the four options PCA, PLS, LPLS and only variable selection is structured by the following criteria: a) explanatory and interpretation power, b) preliminary training results, c) preservation of nonlinear information, robustness in regard to collinearity, d) arguable choice of parameters, e) computational efficiency. Preliminary classification results are computed and the usefulness of the different approaches is assessed.
As there are many combinations of feature extraction techniques and data subsets, not all training results are discussed in detail. The feature extraction methods are compared to each other using the subset X_{rand}, because it is the most unbiased subset and its small size makes the application of LPLS more feasible. The nonlinear interaction terms found by separability analysis are not added in order to keep the known characteristics of the subset.

Subsequently, a hybrid approach is conducted that combines positive aspects of different techniques and the nonlinear interaction terms. The hybrid approach is summarised on page 56 and results in a compact representation of the data which is used for candidate classification.

Normalisation Prior to feature extraction the data is mean-centred. No Z-transformation is applied as a preprocessing step. Three reasons are considered:

First, the binary variables are already well comparable to each other. Second, it was observed during variable selection, that dividing by standard deviation promotes variables of very low variation. Third, the premise of a normal distribution of points among a variable might be questionable. Nevertheless, experimenting with normal scores resulted in no significant changes in classification performance.
Prior to training each score vector is normalised to be in the range of [-1,1].

Training and validation procedure Criterion b) involves a training and a validation step. The procedure is summarised here, in order to avoid text duplication. The components with highest separability scores constitute a training set X_N. N is the number of components included. Cross-validation is performed and 15 neural networks are trained with random bootstraps. The ensemble is validated by ROC and the integral of the MC coefficients is interpreted as a measure of performance. The number of hidden neurons and the number of components N is scanned. For different combinations of parameters performance scores are obtained. These are plotted in figure 4.11 on page 53.

4.2.1 PCA

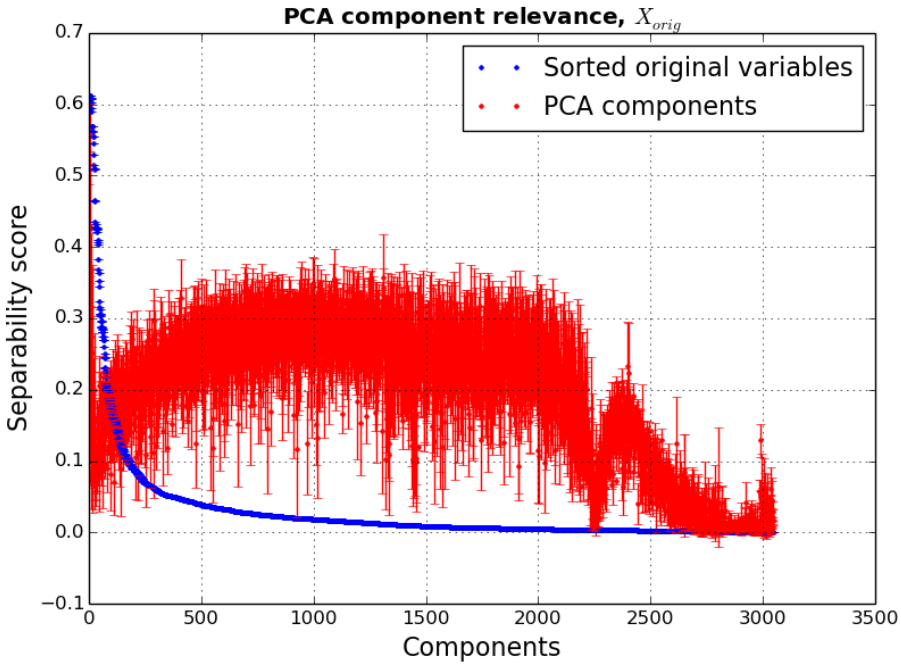


Figure 4.8: The separability score of each principal component is shown. The sorted separability scores of the original variables are also included in the diagram to enable comparison. PCA distributes relevant information over many components.

a) PCA offers statistically independent components which might convey an interpretation. The meanings of the first principal components are analysed and their ability to resolve the key feature is studied.
The variables are sorted according to their absolute loadings in order to find a subset that contributes much to the first component. However, in the first component almost all variables contribute with loadings of similar magnitude. 2800 variables from X_{rand} are selected because they have approx. equally high

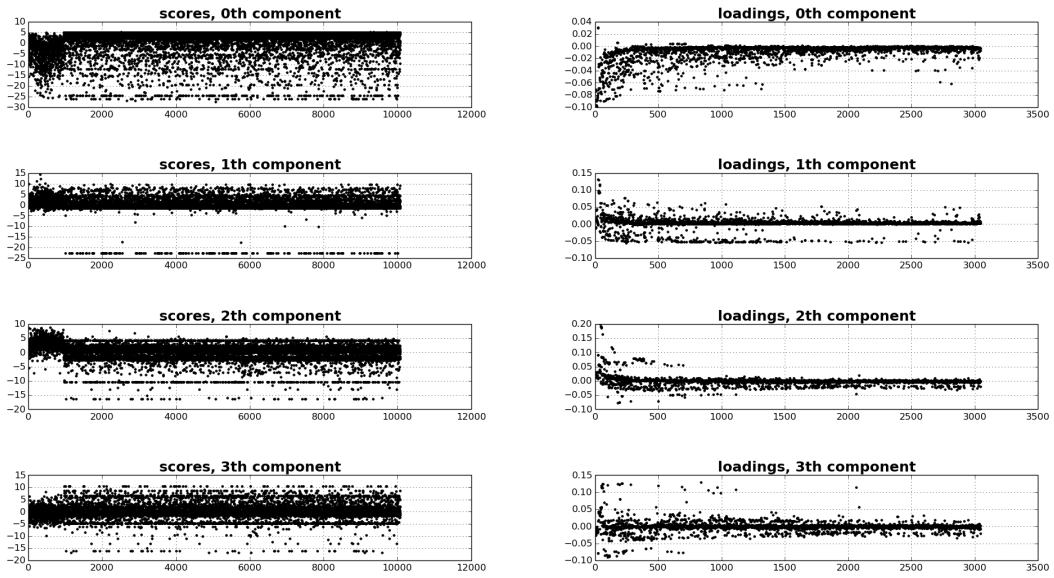


Figure 4.9: Overview of the PCA model showing the first four principal components for the subset X_{orig}. Left column: The score of each protein is plotted on the y-axis for the particular principal component. Right column: The loading of each variable is shown on the y-axis for the particular principal component.

contribution to the first principal component. A particular meaning could no be found. The many highly expressed variables seem to be rather closely related to each other in the biological process branch. The distance matrix that is also used as additional information in the LPLS approach, describes the distances between single GO terms in the ontology tree structure. The mean of the distances of these 2800 variables is taken. The average distance is 7.6. The maximal distance between two variables is much higher (17). The other principal components have only about 100 strong contributors and the mean distance between these GO terms is always higher than 16. Apparently the first component focuses on many neighbouring variables in the biological process branch.

In the second principal component the first 100 dominating variables could not be assigned to any particular ontology branch or meaning.

In the third component the 14 topmost variables are related to negative regulation of various biological processes. These terms are part of the biological process ontology.

The fourth component clearly focuses on terms from the Cellular Component ontology because the 20 terms with highest loadings are all related to cell compartments.

The fifth component contains many variables of different subontologies and no precise meaning could be found.

The fact that some of the components mix sub-ontologies is a hint that there are linear dependencies between them. Thus, there are expected to be synergistic effects between the sub-ontologies, which could improve classification.
The "biological process" sub-ontology seems to contribute much to the inherent structure of the data, because it dominates the first few components. In the chapter about data retrieval 3 there was the idea to remove the biological

process branch. According to PCA this branch seems to be very important for the data and much variance would be lost if it was excluded from the analysis.

In figure 4.9 the scores and loadings of the first few principal components are shown for the full data set X_{orig}. On the left hand side the scores are shown for each instance. On the right hand side one can see the loadings for each variable. The first, third and fourth principal components seem to capture some characteristics of the key feature because there is a transition in the score plots between the true positives (first 952 instances) and the true negatives (the following proteins). The second component does not seem to have the ability to resolve the key feature. These observations are compared to separability analysis in figure 4.10. The separability scores of the first components are encoded as green circles for X_{orig}. The scores of the first, third and fourth components are about double the separability score of the second component. The component relevance estimation from looking at the PCA scores and from separability analysis support each other.

b) Separability analysis is applied on the PCA scores in order to get an impression of the number of relevant components. A grid with six bins is chosen in order to have a low risk of over-training. In figure 4.10 the results are presented. The diagram encodes the PCA components as green symbols. The shapes of the symbols correspond to the different data subsets.
Many principal components can resolve the true positives and true negatives a bit, but no component is able to separate more than 70% of true positive and true negative instances. The remaining components are only able to separate about 0 - 30%.
In a global scale the separability scores of the components are rather equally distributed. In figure 4.8 the separability scores of the PCA components (red) and the separability scores of the original variables (blue) are compared to each other. Since the PCA components are continuous, the separability algorithm is able to compute an error estimate, which is also shown in the diagram. This is not possible for the binary original variables because of the way how separability analysis works. For the data set X_{rand} there is no PCA component that has a higher separability score than the top original variables. PCA does not condense relevant information into few components at all, which makes PCA not be suited for dimensional reduction in this case.

Another observation is there is much fluctuation among the different subsets in figure 4.10. This means that instance selection influenced the major patterns in the data. PCA, which is a completely unsupervised method, is also affected by instance selection and therefore not very robust.

Nevertheless, machine learning is tested. The scores of the principal components are used as training data. The components with highest separability scores are chosen. The training and validation procedure is described on page 44. For different numbers of components and numbers of hidden neurons performance scores are obtained. These performances are presented in figure 4.11a as a three-dimensional plot.
The MC integrals are usually about 0.6 which is about 0.3 lower than for other feature extraction techniques. Adding more components leads to a small improvement of accuracy which supports the idea that the relevant information is distributed over many components. The increase in the number of hidden neurons is bound to a drop in performance due to over-learning. Confronted

with many noisy components, the neural network tends to incorporate irrelevant statistical fluctuations.
The training is redone for X_{anno} and X_{evid}. For some combinations of parameters the integral of the MC coefficients is higher than 0.8. Apparently PCA copes better with less spare data sets. This observation is supported by the results of separability analysis in figure 4.10. The data sets X_{anno} and X_{evid} (green crosses and triangles) are usually able to separate 0 - 20% more true positives from the true negatives than in X_{evid} (green squares). However, the two better performing subsets are biased because of instance selection. Thus, PCA is not recommended in this context for dimensional reduction.

c) The hidden layer in the neural network classifier can learn nonlinear relationships between the input features. However it is found in b) that employing hidden neurons in the neural network classifier does not improve the prediction results. Instead, the networks show signs for over-training. Apparently the neural networks are not able to make use of nonlinear interactions between components. There are at least two possible explanations for this observation. Either the components that show nonlinear interaction are not contained in the selection of components, or the creation of linear combinations destroyed any nonlinear patterns.

d) The choice of principal components for training is similarly difficult as in variable selection. The findings in b) that the relevant information is distributed over many components further complicates the selection process. As the components are linearly not dependent, it is reasonable to perform component selection with separability analysis.

e) Principal component analysis works even in high-dimensional space with up to 10000 variables and samples.

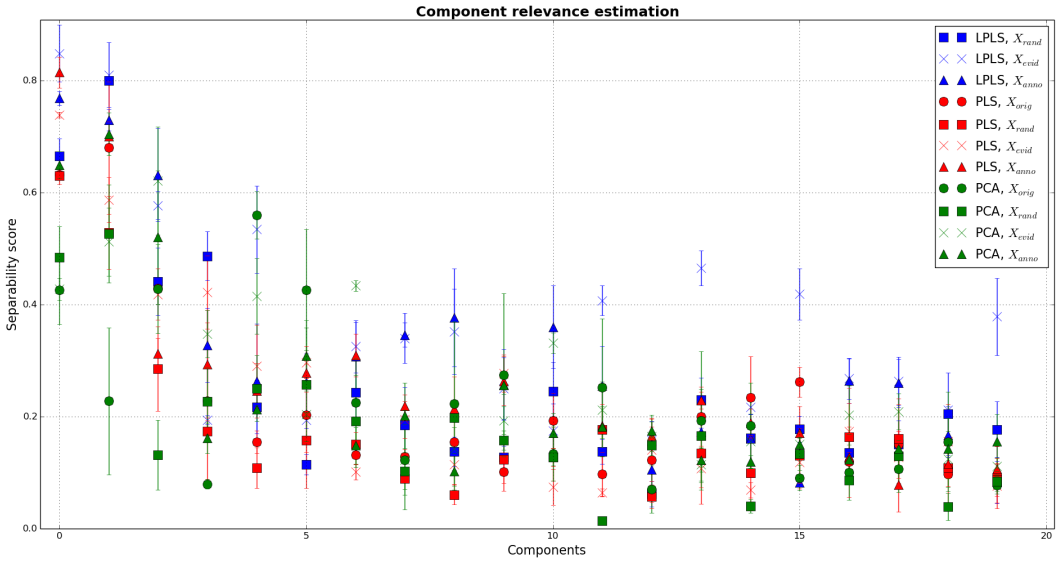


Figure 4.10: Comparison of the dimensional reduction techniques PCA, PLS and LPLS. The separability scores are shown for the first 20 components for different subsets.

4.2.2 PLS

a) The meaning for some principal components of PCA could be unscrambled in the previous section. This is not found to be easy for PLS. Similarly to PCA there are about 2800 GO terms in the first latent vector of X_{rand} that have equally high contribution. For the other components there are usually between 50 and 200 variables that are dominant in the loadings. No particular meaning or sub-ontology could be associated with the first few latent vectors.

b) Separability analysis is applied in order to measure the ability of the individual latent vectors to separate the non DbTFs from the DbTFs. As the latent vectors are sorted from the beginning according to their covariance contribution, it is helpful to identify the first component that has a higher separability score than the precursor. The separability scores are shown in figure 4.10. For the sixth latent vector the separability rises for all data sets. The first two components separate about 60 - 80% of all true positive and true negative instances. After the fifth component the separability scores seem to fluctuate randomly in the range between 0 and 30% just like for PCA.
PLS shows a rather greedy behaviour, because the relevant information content is condensed into about five components. The remaining components have low ability to resolve the key feature.

Training is conducted with the PLS scores as training data. The nonlinear neural network classifiers are used and the PLS regression step is not performed. The learning and validation procedure described on page 44 is applied. The number of hidden neurons and the number of PLS components are scanned. The performances are shown in figure 4.11b. The prediction performance reaches MCC integrals of about 0.92. This is a very good result in comparison to the other feature extraction techniques. The inclusion of more latent vectors leads to an increase of performance if not more than about 25 hidden neurons are employed. In criterion c) the increase of performance is further discussed.

PLS is also tried on X_{anno} and X_{evid}. The classification performances are quite similar. This means that PLS performance is less sensitive to instance selection than PCA, which does not take the presence of the key feature into account.

Outlier proteins that cannot be well classified are not removed, because there are only few proteins - especially DbTFs - available and all proteins are kept in the data.

c) First, PLS was not expected to conserve much nonlinear information because it is a linear approach. There was concern that variables with a nonlinear relationship could end up in a single latent vector, or the dynamics of these variables could disappear in the concert of other variables that contribute to the components. In both cases it is impossible to resolve the interaction between the variables.
However, the training results in b) indicate that nonlinearity is processed by the neural networks: Using some hidden neurons in the networks increase the prediction performance a bit. In addition to that prediction improves if more latent vectors are added. In criterion a) it was found that the additional components do not separate DbTFs and non DbTFs well on their own. Together they surprise during prediction in a positive way. This is another hint for the exploitation of nonlinear information.
In order to be sure that the components really contribute information the

parameter scan was redone, this time with five of the most relevant components and 95 randomly shuffled additional components. In this case the classification did not improve when the random variables were included, but reached a plateau after the first few components. The higher order latent vectors add indeed some useful nonlinear information.

d) The number of components to include in the training process can be optimised by the parameter scan conducted in step b). The MCC integral is about 0.92 for 100 components and ten hidden neurons. In contrast to PCA, where the relevant components are scattered across the whole domain of components, the latent vectors of PLS are already sorted according to their covariance to the key feature. This simplifies component selection.

e) PLS computation is very fast thanks to the NIPALS algorithm. In contrast to PCA, only the first few latent vectors are computed because they contain most of the covariance in regard of the dependable.

4.2.3 LPLS

LPLS enables a sophisticated dimensional reduction approach by considering additional information about the variables. One promising idea is to create a distance matrix that describes the interrelationships between the GO terms. Usually this information would already be contained in descriptive data. However, due to the distinct sparsity adding prior knowledge about variable relationships could assist the feature extraction process.
The distance between two GO terms is defined by the shortest path length in the ontology tree. Terms that are not linked to each other receive a default distance. This default distance is the maximal distance 16 plus one. The distance matrix is normalised by Z-transformation because the variation has to be downscaled to match the other matrices. All data blocks are centred like described in the methodology chapter 2.1.2.
The inversion of some matrices in the LPLS NIPALS algorithm poses limitations. A procedure to simplify the data is described in step e).

a) Similarly to the component analysis of PLS no distinct meaning or sub-ontology could be identified for the first few LPLS components. There is the assumption that the components relate to neighbouring GO terms or GO term branches. In the future the LPLS model creation could be computationally optimised in order to include the entire data set. Then one could check if the variables that are dominant in a particular LPLS component have a short mean distance within the gene ontology tree structure.

b) The relevance of the components is assessed by separability analysis. Separability analysis is applied on the component scores in a similar way to PCA and PLS. The separability scores in figure 4.10 imply that the LPLS components have a strong ability to separate the key feature. However, it is not clear whether the error bars would enlarge if all instances and variables would be used for the LPLS model. The subset X_{evid} shows very good results and its first two components are able to resolve about 80% of the true positive and true negative instances. In contrast to the other data sets and feature extraction techniques, even higher components still contribute valuable information. The reason for this behaviour remains unclear, however there is an assumption:

The true negative instances in X_{evid} were chosen in order to maximise the number of annotations with experimental evidence. Some properties of proteins might be easier or more popular to be studied experimentally. Probably the choice and arrangement of terms in the ontology tree structure was historically influenced by experimental research. The annotation patterns of the true negatives correlate with the GO terms in the gene ontology tree that are usually studied in experimental research. The true positive instances have less experimentally validated annotations and follow different patterns. Hence, the LPLS components, which incorporate aspects from the ontology structure, might distinguish the true positives and true negatives.

Machine learning is performed and classification performance is computed. Neural networks are trained with the LPLS X-scores. The training and validation procedure described on page 44 is applied. Classification results are illustrated in figure 4.11c.
Higher LPLS components seem to add useful information because the prediction results improve if they are included. Due to the limited number of total components computed, it is not possible to say whether a performance plateau is reached or whether higher components will further improve prediction. All in all prediction results are comparable to PLS and MCC integrals of about 0.9 are obtained.
An increase of the inner complexity of the neural network does not lead to a performance drop. One possible explanation is that the LPLS components are not linearly independent from each other. This adds a small amount of redundancy to the reduced data. This small amount of redundancy seems to stabilises the neural network. This effect is also observed with the hybrid data that is constructed at the end of this chapter.

c) It is not entirely clear whether LPLS preserves nonlinear information. Using some hidden neurons during training improves the MCC integral by only 0.01. Thus, the neural networks do not seem to process any useful nonlinear information.

d) The LPLS algorithm requires a parameter $\alpha \in [0, 1]$ that determines the influence of the additional data block Z on the model. The optimal value for α is highly problem specific and is found by optimisation using X_{rand}.
Originally the idea was to find the free parameter by the following approach: Machine learning is conducted with the linear LPLS internal model. Cross-validation is applied. The linear regression results are analysed by ROC and a performance score (MCC integral) is obtained. α is optimised to maximise the prediction performance.
However, it was found that α did not affect prediction results significantly. In addition separability analysis was performed on the LPLS scores for different values of $\alpha \in [0, 1]$. The results can be seen in figure 4.12. In the diagram the separability scores for the first four components are illustrated for different values of α. The change of α has an ambiguous effect on component separability and there does not seem to be an optimal α. $\alpha = 0.8$ is chosen.

e) Due to matrix inversions of the sizes NxN and VxV with the number of instances N and of variables V LPLS is bound to computational limitations. A subset is created. 500 true positives and 500 true negatives are chosen randomly from the data set X_{rand}. The 1000 variables with highest separability scores are selected. A stability analysis with bootstrapping was not done because of the same computational limitations, but should definitely be considered if LPLS

scores are used for candidate prediction.

4.2.4 Variable selection only

In order to test the benefit of feature extraction, machine learning is redone with a subset of variables. One might design a complex procedure to select variables that are not linearly dependent on each other. However, in order to study the effect of collinearity and redundancy, the variables with highest separability score are selected.

a) The criterion explanatory and interpretation power does not apply here.

b) The original variables are used for machine learning. The training and validation procedure is described on page 44. The number of variables included and the number of hidden neurons are scanned. In figure 4.11d the prediction performances are shown for different numbers of hidden neurons and numbers of input features.
Overall prediction results are good but not as good as for PLS and LPLS. After adding about 50 variables a performance plateau is reached at a MCC integral of about 0.75. This performance score is about 0.15 lower than for PLS or LPLS. Adding more variables than 100 did not result in an improvement.

The original data has an advantage in respect to PCA and PLS. Neural network complexity does not worsen over-training. The correlation matrix of the 100 chosen variables reveals collinearity and redundancy in the data. It seems that adding redundant information to the training data does not decrease classifiaction accuracy. Instead, decent amounts of redundancy seem to stabilise the neural networks.

c) Adding more hidden neurons to the neural network does not improve the prediction. If only 100 variables are selected for the training data there is low probability that the variables that show interesting nonlinear interactions are located together in the training data. Using 1000 variables does not lead to higher classification accuracy either. Probably the learning algorithm does not detect nonlinear information because the error surface of such big networks is too complex.

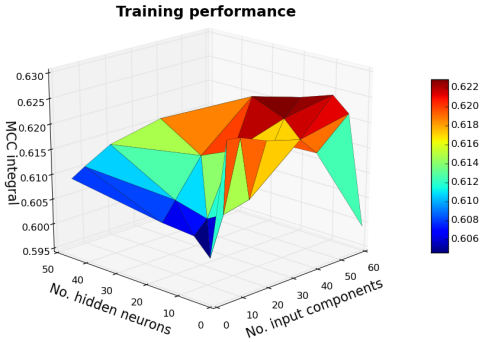
d) There are no parameters to choose.

e) The training of neural networks with few thousand input neurons is computationally possible.

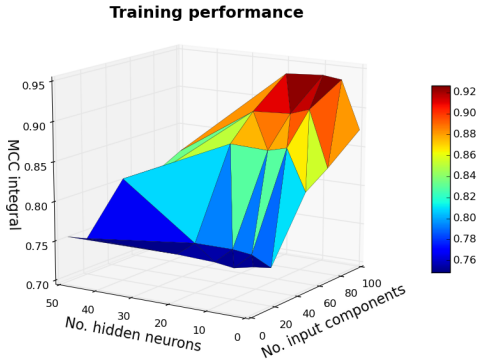
4.2.5 Machine learning results if the "biological process" sub-ontology is removed

In chapter 3 on page 27 it was argued to do the training without the "biological process" sub-ontology. In order to test the feasibility of this suggestion, the data set X_{rand} is modified and the biological process terms are removed. The machine learning process is described on page 44. The variables with highest separability scores are selected for the training data. The number of inputs and the number of hidden neurons are scanned. The MCC integrals are presented in figure 4.11f. The highest performance scores are only about 0.58. This result is compared with the respective classification performances from the data with the process branch included in figure 4.11d. The best performance with the biological process terms included

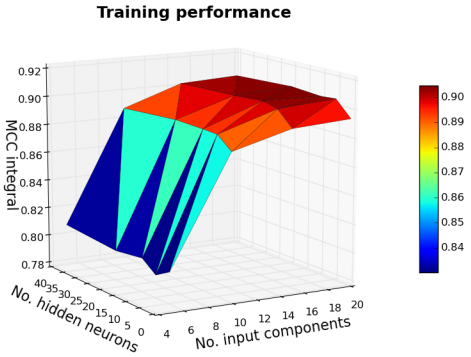
is about 0.75. The high difference in prediction accuracy of about 0.17 is a strong argument to keep the biological process sub-ontology in the final training data.



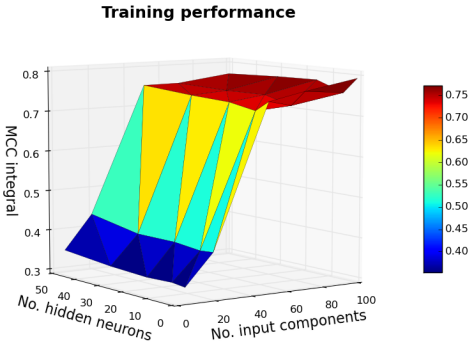
(a) PCA, X_{rand}



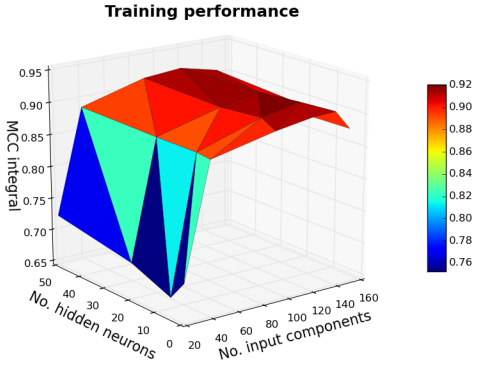
(b) PLS, X_{rand}



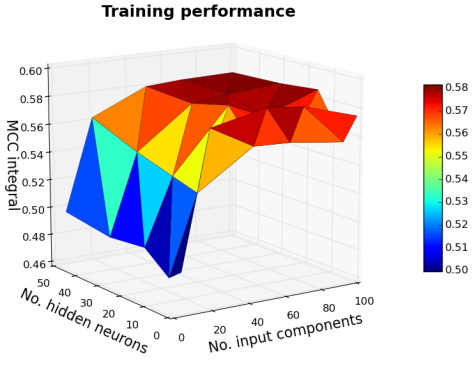
(c) LPLS, X_{rand}



(d) No feature extraction, X_{rand}



(e) Hybrid training that bases on X_{orig}.



(f) No feature extraction, the biological process sub-ontology is removed, X_{rand}. The prediction results are much worse than in the respective diagram 4.11d.

Figure 4.11: The diagrams show the ensemble performances for different parameters. The number of input features and the number of hidden neurons in the ANNs is scanned. The resulting three-dimensional performance surfaces can be used to find reasonable parameters. The detailed procedure of how the diagrams are generated is described on page 44

.

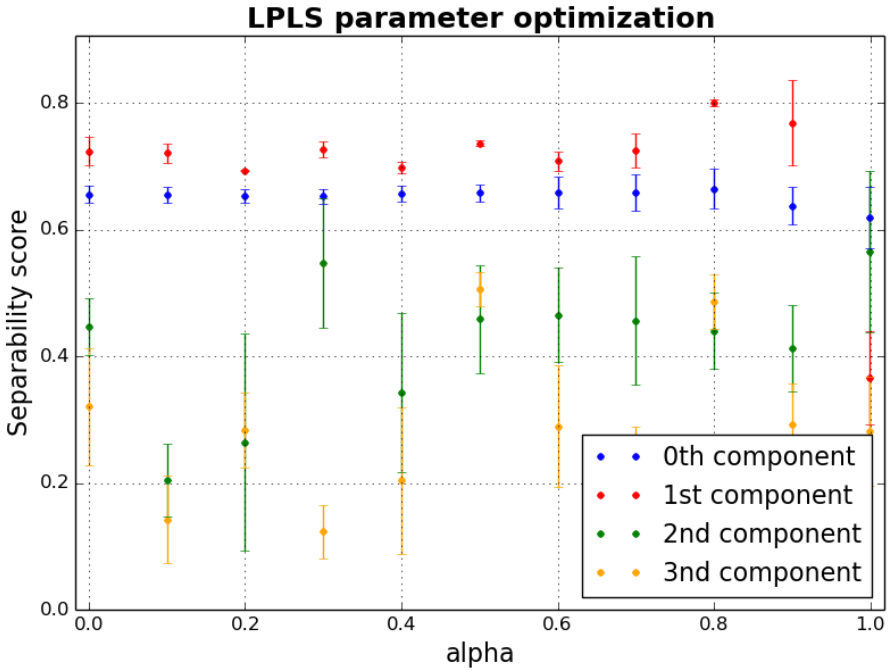


Figure 4.12: Separability analysis is performed on the first LPLS components for different values of the free parameter $\alpha \in [0, 1]$. α has no distinct effect on the ability of a component to separate DbTFs from non DbTFs. $\alpha = 0.8$ is chosen.

4.2.6 Conclusion, Final approach of preprocessing

Data set selection
During data retrieval in chapter 3 four data sets were obtained. The previous discussions show that the data sets X_{evid} and X_{anno} are biased because of their selection of true negative instances. No specific bias is found for the data set X_{orig} and the subset X_{rand}. Dimensional reduction with a standard deviation filter and PLS or LPLS yielded the best classification performances. results for classification. X_{orig} is chosen as basis for the final training data, because it contains 952 true positive instances, as well as all the 9123 true negative instances. During cross-validation random bootstraps are taken so that the number of true positive and true negative training instances are equalised before machine learning. The additional true negative instances are supposed to add more information for training and increase the size of the validation sets for ROC validation.

Hybrid approach
The experience gained during variable selection and feature extraction leads to a hybrid approach that is proposed and implemented. It combines positive aspects of different dimensional reduction techniques. The idea is to train the neural networks with PLS scores and a collection of some non redundant original variables. This strategy is chosen to add more granularity to PLS components and to add a small amount of redundancy that possibly stabilises the training process.

Selection of original variables
The selection of original variables is determined in two steps. First, variables are pruned in order to minimise redundancy. Then, the remaining variables are selected according to separability scores.
A subset of rather linearly independent variables is obtained by hierarchically clustering using the correlation distance matrix (1 - correlation matrix). The procedure is similar to the variable pre-selection from finding the nonlinear interaction terms on page 4.1.2. During clustering the complete metric is used that defines the distance between two groups u and v as $d(u, v) = \max(d(u[i], v[j]))$ with the euclidean metric d. In figure 4.13 the number of clusters and the variation of their sizes are shown for different thresholds. One can see that a low threshold leads to a high number of clusters that are still quite linear dependent on each other. For a correlation threshold higher than 0.6 large clusters condense. The variation of the group sizes increases, because there are few super-groups and some very small groups. This is not desired due to a loss of information. The threshold 0.6 is applied because is it the highest threshold with similarly sized clusters. For each group one representative is chosen with the criteria of highest separability score.
Second, separability analysis is performed and the remaining variables are ranked according to their ability to resolve the key feature. A threshold of 0.2 is applied in order to select the most relevant ones. This means that the variables can separate 20% of DbTFs from non DbTFs. 15 representatives are obtained.

Selection of PLS components and network structure
The number of PLS components to include into the training data and the number of hidden neurons are determined by scanning both parameters and computing the MCC integrals as a measure of performance. The PLS components with highest separability scores are included. 3-fold cross validation is performed and 15 neural networks are obtained for each parameter setting. The MCC integrals are averaged for each setting. The results are presented in figure 4.11e. The classification performance has

a plateau for about 115 input features. The MCC integral reaches about 0.92 which is comparable to the results of pure PLS. In order to make the diagram comparable to the other data sets and dimensional reduction techniques the number of true positive and true negative indies are equalised. Otherwise there would be an excess of true negative instances. For more details, see chapter 4.4.1 on page 58.
Increasing network complexity does not lead to a decrease in performance. The neural networks that were trained with the hybrid data are less prone to over-training than the neural networks that were trained with the pure PLS scores. This is another hint that a small amount of redundancy has a stabilising effect on ANN training. In addition there might be positive synergistic effects between the variables and the PLS components.

Number of training epochs
The optimal number of training epochs is chosen based on the training errors shown in figure 4.14. After about 13 training epochs the training and validation errors do not overlap any more, which is the first sign of over-training. 10 training epochs seem to be sufficient for the task and over-training is kept to a minimum.

4.3 Summary, stepwise instructions for the final prediction approach

1. Variable selection: Gene ontology terms with a standard deviation smaller than 0.1 are removed.
2. Enrichment with interaction terms: The interaction terms are obtained in a two step approach. In the first step a pre-selection of variable pairs is assessed for nonlinear interaction. In the second step for each promising variable pair a nonlinear operation is chosen and applied that results in a new variable. First, the 500 variables with highest standard deviation are assessed. The correlation distance matrix (1 - correlation matrix) is computed and hierarchical clustering is performed. The complete metric is used that defines the distance between two groups u and v as $d(u, v) = \max(d(u[i], v[j]))$ with the euclidean metric d. The clustering threshold 0.6 is chosen to construct the groups. For each group the representative with highest standard deviation is chosen. Variable pairs that exceed the separability score 0.5 are kept. In the second step the interaction terms are created. Different logical operations are tried: x_0 AND x_1, x_0 OR x_1, x_0 AND $\neg x_1$, $\neg x_0$ AND x_1, x_0 XOR x_1. The one-dimensional separability scores are recomputed for each interaction term separately. For each variable pair the mode with highest separability score is chosen. Eight interaction terms are identified and included in the data.
3. Normalisation: Mean centring is applied on the data.
4. Feature extraction: The first 100 PLS latent vectors are computed. The PLS scores represent the reduced data.
5. Hybrid data construction: A selection of original variables is merged with the PLS scores. This selection is obtained by the same clustering technique as in the second step. The threshold 0.6 is chosen for cluster construction. 15 group leaders are chosen based on highest separability score and included into the hybrid data.

6. Normalisation: The input data is normalised to fit the range [-1,1] which is the output range of the neurons in the neural network.
7. Training: 3-fold cross-validation is performed and 1500 neural networks are trained with random bootstraps. During the process the number of true positive instances and true negative instances are equalised. 115 input neurons, ten hidden neurons in one hidden layer and one output neuron constitutes the feed-forward ANN structure. The training consists of ten epochs with the Rprop algorithm.

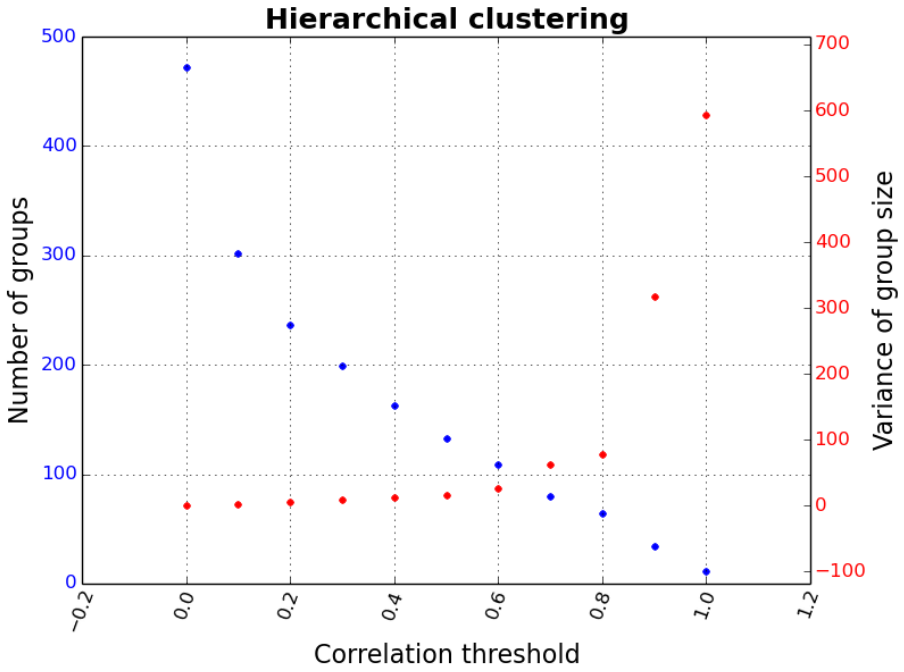


Figure 4.13: For the hybrid approach a variable subset is chosen that has low collinearity and redundancy. Variables are hierarchically clustered using the correlation distance matrix. The correlation distance threshold for group construction is scanned. The number of groups and the variance of the group sizes are shown in the diagram.

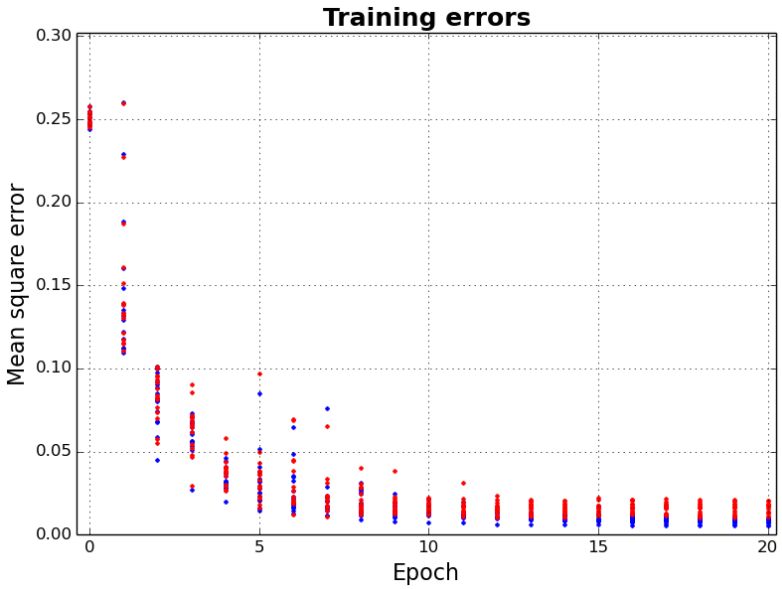


Figure 4.14: The training errors are shown for the final training data and for a random subset of ANNs. Red markers show the validation errors and the blue points represent the training errors. Ten epochs seem to be optimal for training.

4.4 Postprocessing

4.4.1 ROC validation

An ensemble consisting of 1500 single neural network predictors is trained with bootstraps from the final training set via 3-fold cross-validation. The ensemble is validated by ROC. Only the validation instances are used for validation respectively for each single network. Validation of the ensemble performance is done with great care so that no training data is involved in validation: The prediction output of the ensemble is defined by the mean output of all single predictors that do not have the particular instance in their training set.
In the figure 4.15 sensitivity, specificity and MC coefficients are presented for different discrimination thresholds. Figure 4.16 plots sensitivity against specificity. The dashed lines represent the performances of a random subset of single predictors. The striped, solid lines relate to the ensemble. There exists a broad region where both sensitivity and specificity is high. At the threshold 0 both sensitivity and specificity of the ensemble are about 0.98 and the MCC coefficient is about 0.93. About 98% of all validation samples are accurately classified. The integral over the MCC coefficients is 0.870.
In the ROC diagrams one can see that specificity tends to be worse than sensitivity. In figure 4.16 the specificity is about 0.03 lower than sensitivity. One explanation is that the validation sets have about 30 times more true negative instances than true positive instances. If the pool of true negatives is reduced to equalise the number of true positive and negative instances before cross-validation (similarly to the data set X_{rand}) the number of validation instances is equalised. In this case the asymmetry vanishes. The MCC integral of the ensemble increases from 0.870 to 0.921. Hence, the excess of non DbTFs in the set X_{orig} explains the asymmetry. The over-training effect reveals itself only for large validation sets. As the number of DbTFs is very limited it is not possible to exclude a similar over-training effect for sensitivity.

In addition to that the MCC integral is computed for each neural network separately. The mean and standard deviation is taken for the integrals and the performance 0.766 ± 0.055 is obtained. The higher prediction performance of the ensemble (0.870) is in accordance with the findings from the papers [23] and [15]. The ensemble has much better generalisation abilities than the single predictors.

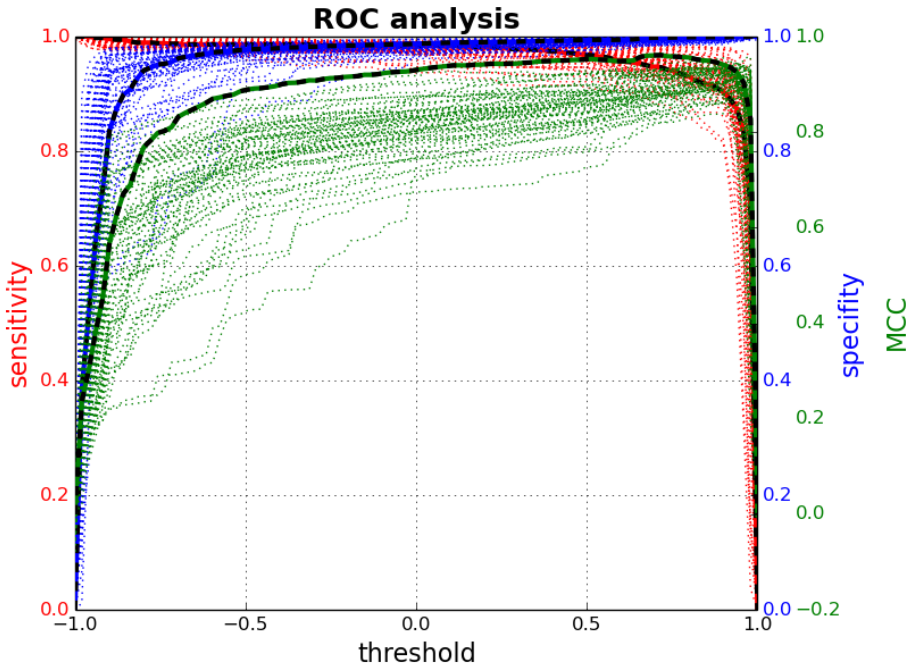


Figure 4.15: ROC curves showing the prediction quality for different discrimination thresholds. The final training data is used. The solid lines represents the ensemble performance. The MCC integral is computed to 0.868 for the ensemble.

4.4.2 Results and discussion, Sensitivity analysis

Relevance assessment of gene ontology terms is an important aspect of dimensional reduction. During variable selection in chapter 4.1 various ways to rank variables are discussed including standard deviation, separability analysis, correlation coefficients and PLS regression coefficients. The trained neural network ensemble can be studied in order to get additional evidence for variable relevance estimation. Variable selection can be optimised by pruning noisy variables and components. Subsequently training is redone and ideally better prediction results are obtained.

In the methodology chapter 2.3 two approaches are proposed. These include performance sensitivity and a finite difference approach.

Performance sensitivity

While experimenting with this method a problem was encountered. Retraining of neural networks usually changes them slightly due to the non deterministic training algorithm and different bootstraps during cross-validation. These fluctuations were in the same range as the change in prediction performance that is induced by leaving out a variable. One computationally expensive solution is to train hundreds of predictors to enable better statistics, however this turned out to be not feasible.

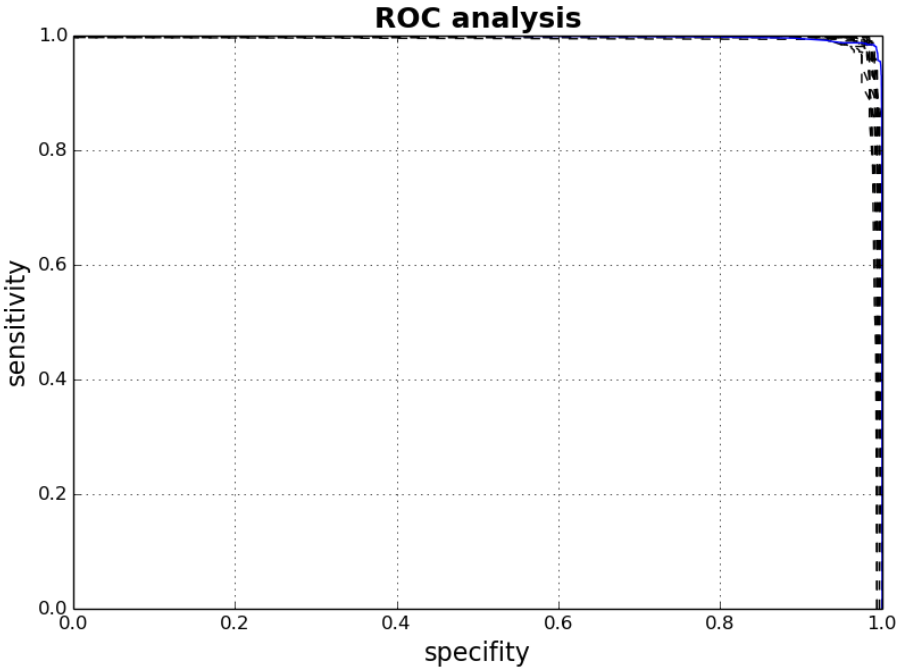


Figure 4.16: ROC curves showing the relation of sensitivity and specificity. The final training data is used. The solid blue line represents the ensemble performance. There exist thresholds with both, high specificity and specificity. For some ANNs specificity is about 0.03 lower than sensitivity.

In order to remove the random fluctuations and significantly improve computational performance the approach is modified. Each input feature is assessed separately. First, all samples in the data set are classified by the trained ensemble and ROC is performed. The MCC integral p_0 of the ensemble is computed. Subsequently the i-th input feature is randomly shuffled and the patterns for this feature are destroyed. Prediction is redone and a different MCC integral p_i is obtained. For the i-th feature this procedure is repeated ten times in order to decrease a potential bias that is created by shuffling. The mean of the differences

$$s = \frac{1}{N} \sum_{i=1}^{N} |p_0 - p_i| \tag{4.2}$$

is interpreted as the influence of variable i with sensitivity score s. Standard deviation of the differences is chosen as an accuracy estimate.

The sensitivity results are computed and shown in figure 4.17. In the diagram the separability scores are also shown for comparison. One observes a strong level of agreement for the four highest ranked input features. However, sensitivity and separability place different emphasis on the other input features. The overall correlation coefficient 0.53 shows a medium level of dependency.
Although the single predictors were trained with different bootstraps, the sensitivities are very similar because the error bars are quite small. This implies a low level of over-training. One might argue why the error bars do not explain these deviations. However, sensitivity and separability measure variable relevance via two fundamentally different ways and it is not expected that both quantities match numerically.

Pruning input features that have a low relevance estimate and repeating the training could improve the prediction accuracy. However, there was no improvement no

matter what combination was tried. If the four highest ranked features are removed the MCC integral drops from 0.870 to 0.822. This is a hint that these features are indeed relevant for prediction. No subset of features could be found that actually improves prediction.

Finite difference approach

In the finite difference approach the input features are assessed for their influence on the predictor. The response Δy of the ensemble is computed for the variation $\Delta x_i = 2$ of input feature i. The input i is set to +1 and -1 respectively. The sensitivity $\Delta y/2$ is averaged along all instances in the training set in order to have a data consistent statistical analysis. Let y_i^+ and y_i^- be the outputs of the ensemble for the high and low input. The sensitivity s is computed by

$$s_i = \frac{1}{2}\overline{y_i^+ - y_i^-}$$
$$\delta s_i = \frac{1}{2}\overline{\sqrt{\delta y_i^{+2}} + \sqrt{\delta y_i^{-2}}}$$

where δy_i are the regression accuracies for each instance and the overbar denotes to the average.

The sensitivity scores are compared to the separability scores. ,The feature relevance estimates show almost no sign of correlation. The correlation coefficient between the separability scores and the absolute sensitivity scores is only 0.03.
The finite difference approach is compared to the scores from performance sensitivity. In figure 4.18 both sensitivity measures are plotted against each other. The correlation is stronger with a correlation coefficient of 0.53. The highest ranked features agree, but there is much difference in the relevance assessment of the other features.

No subset was found that improves classification results. Pruning input features that have a low score for both approaches did not result in better prediction results either. Apparently the information flows in the artificial neural networks are too complex to be studied by simple input-output measurements. The idea to use sensitivity analysis as an intelligent way to further prune input features and obtain a more powerful training set does not seem to be feasible. In addition to that it seems that the training process of neural networks are very robust in regard to small amounts of noise or redundancy.

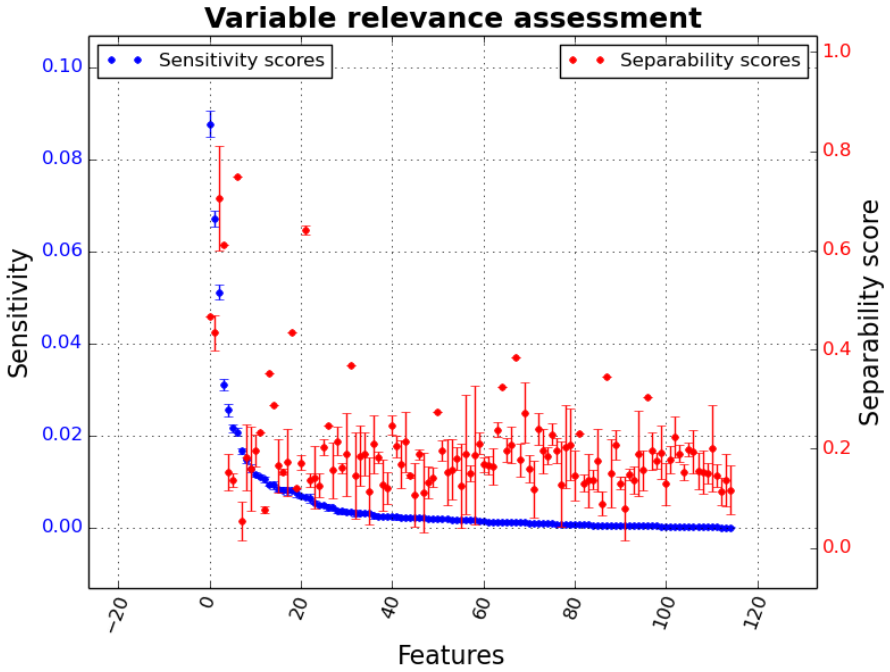


Figure 4.17: Sensitivity analysis is performed with performance sensitivity (blue). The resulting relevance assessment is compared to separability analysis (red).

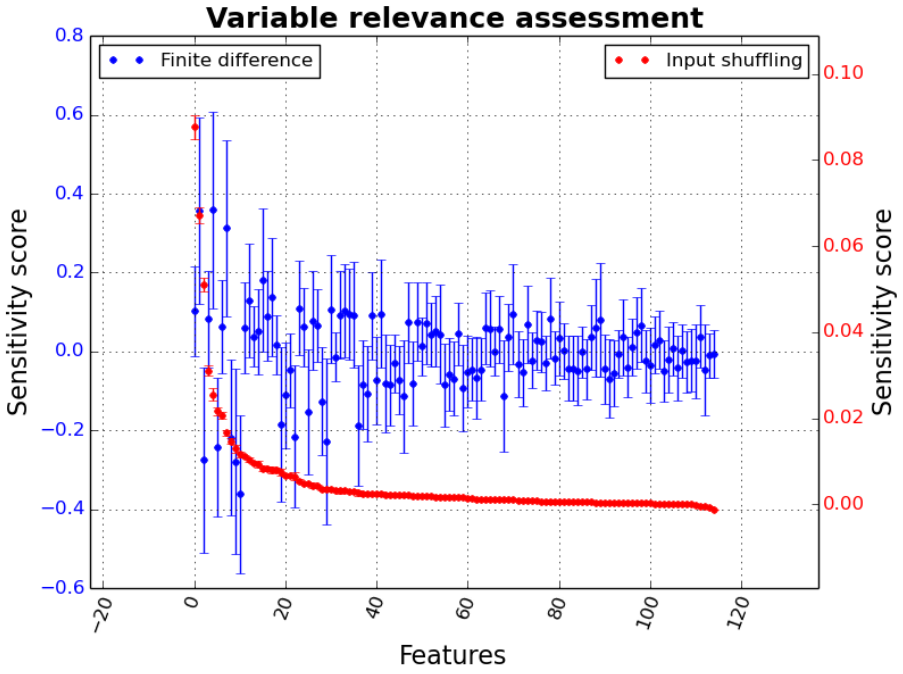


Figure 4.18: Sensitivity analysis is performed with the finite difference approach (blue). The resulting relevance assessment is compared to the sensitivity results from performance sensitivity (red).

Chapter 5

Results and Discussion, Candidate Classification

In chapter 4.3 on page 56 the composition of the training data and the machine learning process is summarised. The stepwise procedure is implemented and ROC validation is performed in chapter 4.4.1 on page 58. Subsequently the candidates are classified whether they are specific DNA binding RNA polymerase II transcription factors (DbTFs) based on gene ontology annotation data.
The candidate proteins that are extracted from the database TFcheckpoint in chapter 3 are classified with an ensemble of 1500 single predictors.
The gene ontology data is brought into the same shape as the training data. The same mappings and transformations are applied, the same interaction terms are computed and the same PLS model is used. The same normalisation is performed in order to map the feature vectors on the [-1,1] interval, which is a necessary step when working with neural networks. Some values are outside the interval. This is a natural effect from using new data and no issue. These outliers are set to the interval boundaries in order to make the input compatible to the neurons in the neural networks.

Prediction results

The regression results are computed for the single neural network predictors. The regression scores are averaged for each candidate and the standard deviation is interpreted as accuracy estimate. From 2655 candidates 398 are classified as DbTFs and for 228 of these the error is lower than the regression result. 2257 proteins are classified as non DbTFs and for 1990 of these the error is lower than the regression result.

The relative error is computed in order to make a subset selection of the candidates. Let y_i be the regression result of candidate i and δy_i the accuracy computed. $y_i = 1$ means positive classification, $y_i = 0$ means no agreement of the neural networks and $y_i = -1$ means negative classification. The relative error is $\delta_{\mathrm{rel,i}} = \delta y_i / y_i$.
Finally a ranked list of predicted DbTFs is created. A subset is chosen that constitutes the most certain DbTFs. In a selection of 54 potential DbTFs each protein has a relative error of less than 10% to be a DbTF. The ranked list of the 54 candidates, their regression results and accuracies is presented in the appendix 9. From these selected candidates each is classified by 1500 neural networks with a regression result

of higher than 0.95. The standard deviation of the single ANN votes is always lower than 0.095 for each selected candidate, which is a high level of agreement.

Discussion

The accuracy estimate must not be interpreted as an absolute measure of certainty because of the epistemic and methodological limitations that arise from the data and the prediction techniques. Inside the context of the retrieved data at least 2% of all instances are classified in a wrong way according to ROC. The methodology for dimensional reduction and prediction seems to be suited for the problem at hand. The chance of a biased prediction result is minimal for the 54 candidates, because the single predictors were trained with different bootstraps from the data and large validation set were used (about 630 true positive and 600 true negative training instances per ANN, as well as about 320 true positive and about 8418 true negative validation instances).
In a more general context it is difficult to assess accuracy because there exists only a very limited number of positive training instances. During training it was assumed that there might be undetected over-training effects. The reason for this assumption is the observation that the inclusion of the excess of true negatives in the cross validation process leads to slightly asymmetric ROC curves. Sensitivity decreases by about 3%. The MCC integral of the ensemble decreases from 0.921 to 0.870. Hence, the additional true negative instances enable the detection over-training. Possibly this is also the case for the true positives.

In addition to this it is not clear to what extend the data from biological databases are influenced by the knowledge that a protein is a DbTF. Possibly there is specific research conducted on known DbTFs, thereby adding to the patterns in the GO annotation data. Obviously these characteristic patterns do not exist for unknown DbTFs. Consequently, the preferences of researchers influenced data structure in the past. This could imply a low sensitivity for the application of the trained ensemble on unknown DbTFs.
One criterion to test this assumption is to redo the training and prediction with training data that has a low probability to be influenced by research on DbTFs. One option is to exclude the biological process sub-ontology, like mentioned in chapter 3. This option was tested, the training results turned out to be inferior and no satisfactory prediction results were obtained. The MC coefficient integrals were about 0.15 lower than with the sub-ontology. This means that no reasonable prediction can be performed without the process ontology. In order to check to what extend earlier preferences of researchers biased the gene ontology data, one could select a different data source and compare the classification results. This suggestion is specified in the outlook chapter on page 67.

Another aspect of accuracy debating is the selection of true negative instances. In chapter 3 it is argued that the true negatives should have a balanced level of dissimilarity to the true positives and similarity to the candidates. A major difficulty is to assure that chosen negative instances are not unknown DbTFs. This problem is addressed in the current approach by a GO term based selection of potential non DbTFs. In addition an excess of negatives is included in the validation process and the individual classifiers are trained with random bootstraps from the large pool of negatives. If, for example, about 90 unknown DbTFs were among the 9312 non

DbTFs there would only be a percentage of 1% wrong instances. This strategy works well if the assumption holds that the number of unknown DbTFs is not much higher than the number of known DbTFs.

Chapter 6

Outlook

As a consequence of the results and discussions from methodology development in chapter 4 and the discussion of the candidate classification, improvements and validation concepts are suggested for further work.

In order to collect additional evidence for the candidate prediction it might be necessary to change the data source into one, that is less influenced by research preferences and former interpretation. Physical, structural or chemical properties offer a different perspective on the candidates. There is a broad range of promising information that was not covered in this project including amino acid sequence similarity, motif and structural protein data.

Another idea to surpass the limitations of the data is to extend it with other orthologs. Including more species will increase the amount of available data and enable more careful validation. Possibly the current gene ontology data is already influenced by other orthologs because of the internal structure of the knowledge base and computationally inferred annotations.

Although the chosen methodology obtains good prediction results and the validation implies an accuracy of 98% correct classifications, it might be interesting to study more aspects of machine learning with sparse data. In the project different kinds of dimensional reduction techniques such as separability analysis, PCA, PLS and LPLS were implemented and compared to each other and the focus was on a neural network approach. As a next step it would be interesting to computationally optimise LPLS and further test its abilities to include additional information in the dimensional reduction process.
Possibly the not optimal results of PCA can be further improved. A different normalisation of the data can be performed to obtain the same results as correspondence analysis.

An additional question relates to the characteristic input patterns that lead to distinct classification results. Gallego, Gago and Landín [12] propose a procedure that employs a genetic algorithm to find input combinations that maximise the classification result. Thus artificial representations of fictive proteins are obtained by a meta-modelling approach. The exemplary proteins can be statistically analysed, e.g. by PCA or correspondence analysis in order to find characteristic patterns for DbTFs and learn about the proteins. The first step was implemented during the work on the project. A genetic algorithm was prepared and applied to create some artificial proteins that had very high regression results of about 0.99. The subsequent analysis is expected to take some time in the future.

Chapter 7

Future Prospects: Drug Synergy Prediction

7.1 Introduction

Drug synergy is the "unexpected" effect of two drugs used simultaneously, beyond the expected effect based on separate treatment with the single drugs. This synergistic effect is essentially the deviation from the addition of the individual effects. Based on the experience gained in the main part of the thesis, a proposal is prepared for drug synergy prediction based on machine learning. The implementation of the proposal is expected to take some time in the future. Some ideas in this chapter might be inspiring for future machine learning approaches related to drug synergy prediction. Recently the AstraZeneca-Sanger company exclaimed the Drug Combination Prediction DREAM Challenge (`www.synapse.org/#!Synapse:syn4231880/wiki/`). It was hoped that the research community would find strategies to integrate various types of provided experimental data and pre-knowledge in a crowd competition situation in order to reach one step closer to personalised cancer treatment.

Synergy can be caused by many biochemical mechanisms. Here one starts from the premise that it is reasonable to assume that synergy may occur prevalently when protein targets of different drugs act in different or parallel pathways. From this assumption follows that prediction of drug synergy needs a study of the functional interdependencies of proteins in life-sustaining cellular processes. Knowledge about these processes, about the proteins that are members of these pathways and about the putative drug targets among them exists, however it is still difficult to integrate these different pieces to obtain a similarity assessment for two drugs in the context of the actual cell line and individual patent. In the following it is described how this assessment could be performed, how the results are used to construct a training set for machine learning and how the training could be conducted.

Drug targets may play a role in relevant pathways that rely on complex protein-protein interactions and biochemical signal transduction events. The complexity originates from the fact that almost all proteins perform their function together with other proteins in protein complexes. These complexes are formed based on a protein's ability to specifically interact with their molecular environment due to their unique shape and electrostatic landscape. A protein can be required for proper functioning of another protein, or needs partners in a protein complex in order to properly function. There are several ways to assess how proteins are associated

with each other, meaning that data can be generated that supports their functional relatedness. These include:

a) Proteins that are members of the same complex may have high interdependency

b) Proteins working closely together in the same pathway(s) have interdependent functionalities

c) Closeness of proteins within the Reactome global signal transduction network may be converted into "relatedness"

d) Genes that are coordinately expressed in mRNA data in the same cell line in different environments might indicate dependencies

e) Proximity of proteins measured by an adequate metric (e.g. number of terms and relationships along the path connecting the terms that annotate the proteins) inside the biological process branch of the Gene Ontology might detect relationships

f) Joint presence in the same regulatory cascade that connects to (anti-) survival node in the logical model of Flobak et al. [10]

g) others...

It is beneficial to take several of these aspects into account which may indicate the relation of protein function, since different perspectives on the problem might increase granularity of the drug data and make a similarity assessment of two drugs more specific. A small survey of the literature has therefore been performed.

In the first approaches a) and b) sets of interdependent proteins are considered that are likely to cooperate in protein complexes or pathways. These sets constitute models or networks that are affected if a single protein member is targeted. The Molecular Signature Database (MSigDB) from Gene Set Enrichment Analysis (GSEA) might help to find relevant protein sets. The sets are checked for the combined presence of two drug targets. The number of hits might be an indicator whether two drug targets have the potential for synergetic effects. However, this procedure alone does not take into account interdependencies of different pathways and lacks quantitative information. The pathways themselves are part of a network of cellular processes and might affect each other.

Another approach c) to obtain the 'relatedness' between drug targets is offered by the Reactome pathway database (`www.reactome.org`). Reactome can be used to find the minimal distance between any two genes in its global signal transduction network. One way to obtain a dataset may be to download the Reactome data and use NetPathMiner [29] to calculate all shortest paths and then select the drug target pairs. Alternatively, the PathwayAnalyzer tool in Cytoscape may be used, but probably the Reactome file sizes will be prohibitive. The list of shortest paths will provide the data for a relatedness matrix.

Whereas the drug target relatedness may provide a general framework for predicting drug synergy, the likelihood that this may result in significant correlation with synergy seems low. To increase the correlation some insight in the black box that the cells represent is needed. Cell lines will pose a specific context for treating general knowledge about target relatedness, and customise its use in that particular cell. Information about the black box might be found in the additional data provided by the DREAM challenge: mutations that alter sequences in the protein's peptide chain leading to new conformations and (mal-)function, methylation affecting DNA condensation and hence transcriptional gene control, deletions or duplications affecting

the number of copies of target genes, or even experimental accuracy. If not taken into account, these cell-specific contexts introduce noise or systematic errors and will likely dominate the data. Hence, the effects of proteins perturbing or influencing each other need to be estimated for each cell line. This could be achieved with the mutation data, methylation data or the copy number variation data provided by DREAM.

7.2 Proposal

The challenge to predict drug synergy is the combination of a data integration problem and a machine learning problem. The former addresses the need to make best use of background knowledge and is solved by identification of quantities that are more meaningful and condensed for machine learning than the vast accumulation of raw data. The latter lies in the choice of a suitable predictor and validation technique.

In the literature attempts have been made to combine different types of omics data. Nagaraj and Reverter [31] integrate a variety of gene properties like kinases, secreted proteins, transcription factors, post-translational modifications of proteins, DNA methylation and tissue specificity. They analysed cancer-associated genes and constructed a probabilistic truth table to find more candidate genes. In the work of Vaske et al. [42] altered activities in cancer related pathways were identified. Factor graphs incorporated different kinds of gene related data (DNA copies, mRNA and protein levels, and activity of the protein) and were applied to model genes and their interplay within pathways.

So far, no method is known that combines data on the cell line level (gene expression, methylation, copy number variation, mutation) and on the drug level (protein targets, generic drug properties) into one framework for machine learning purpose. We design an 'Omics Data Integration Network' (ODIN) that serves as a framework to flexibly incorporate different data. The network structure is derived from prior knowledge found in the Reactome database and the Molecular Signatures Database from GSEA. Reactome curated GSEA sets are considered as pathway related modules. These modules define nodes in the ODIN framework. Network construction is divided into three steps: module creation, module enrichment and module linkage.

First, empty nodes are created and associated to a particular GSEA set like seen in figure 7.1. Second, in the actual data integration process, drug and cell related information is propagated through the pathway models resulting in model specific quantities. This step is visualised in figure 7.2. The quantities are attached to each node respectively, forming vertices that characterize the pathway associated to the module in a drug and cell line specific context. Third, the modules in the nascent network are linked together by machine learning and an artificial neural network (ANN) approach. The synergy measurements provided by DREAM and the vertices constructed in step two represent training sets for this purpose. During an iterative training process, the ANN is taught how to link the modules and their attached information in order to predict synergy. The figures 7.4 and 7.5 illustrate the last step.

Module creation

The Molecular Signatures Database (`software.broadinstitute.org/gsea/index.jsp`) provides curated gene sets in different biochemical contexts. The Reactome curated set contains 674 pathway related gene sets. Each gene set is associated to one node in the data integration network in a one to one relationship.

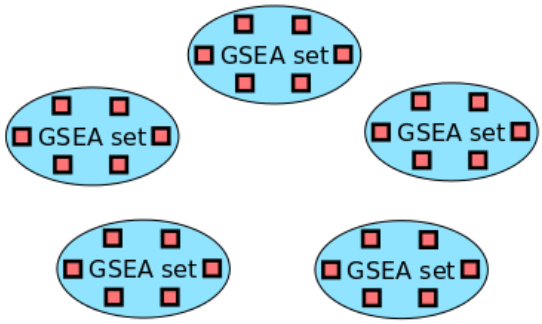


Figure 7.1: Reactome curated gene sets are extracted from the GSEA Molecular Signatures Database.

Module enrichment

The originally empty nodes are annotated with quantities that characterize the GSEA set under consideration of the two drugs or the cell line. How this is achieved in detail is data type dependent.

In respect to the drug related data, knowledge about the target protein(s) of the drugs are considered. One can estimate the significance of a protein target in a GSEA set by a membership check. For each module a binary value is computed that simply states whether the protein target is contained in the GSEA set or not. The value is attached to the corresponding node in the data integration network.

Treatment of the cell line specific data is similar like done by Nagaraj and Reverter [31]. Integration of gene expression data is performed by overexpression analysis. The expression levels of each cell line are compared to a reference. The reference level might not be found in any database due to the presence of mutations in the cell lines. Mutated or cancerous cells might have developed some differentiated expression profile. It might be easier and more accurate to choose the average expression based on the present data as a reference. For each cell line for each gene a discrete number -1, 0, 1 is computed that is interpreted as underexpressed, normal expressed or overexpressed. The overexpression analysis is used to enrich the GSEA set annotation by two quantities: the fraction of overexpressed and underexpressed genes in a set.

The mutation data is transformed into a binary event matrix as proposed in the DREAM challenge description. The entries in the matrix describe whether a gene is mutated in a particular cell line or not. This insight is projected onto the pathway level with the help of the GSEA sets. For each module a binary quantity is computed that encodes if at least one gene in the module is mutated or not. Methylation and

copy number variation data might be integrated similarly to the mutation data in order to receive a binary classification.

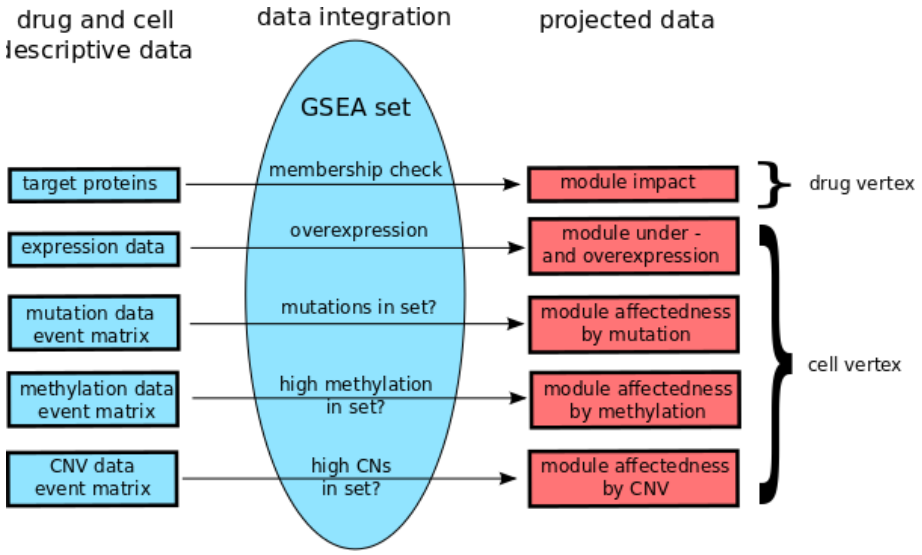


Figure 7.2: The gene set modules are enriched with drug and cell line specific quantities. As a drug pair consists of two drugs, there will be two contributions.

Module linkage

In the former steps we set up the modules of the nascent ODIN network and added drug and cell line specific information to the nodes. In order to predict synergy, the quantities attached to the nodes have to be linked together to simulate module interactions. An artificial neural network is applied and trained to learn the causal links between the modules. The synergy measurements offered by DREAM and the GSEA module vertices obtained from the steps before are used to construct training sets. The design of the network structure can be seen in figure 7.5. The artificial neural network is trained in order to find a highly nonlinear mapping from the vertices that are attached to each module to a synergy output. Hyperbolic tangent functions are chosen as activation functions. The hidden nodes are connected to a bias node. An iterative backpropagation learning method is applied. Cross validation monitors the learning process to detect over-training. ROC analysis can be conducted to measure prediction accuracy.

One possible downside is the complexity of the task and the limited number of training sets. Also it might be an excessive demand to confront the learning algorithm with all module annotations at the same time. An unmodified approach does not appear to be feasible, because the predictor has to consider both cell line and drug specific information - the ANN would have to learn all the biomolecular interactions in many types of cells and the characteristics of all drugs at once. This situation is illustrated in figure 7.3.

The complexity that a predictor has to deal with has to be reduced. It is more feasible to train several artificial neural networks that are referred to as ‘experts’ in a cell line specific context. This means that for each cell line a predictor is trained. These predictors are confronted with drug specific information only, as the cell line’s specific information is constant. If, after training, an unknown drug combination should be examined for synergistic effects, the correspondent cell line expert will

be able to vote. The other way round, predictors with drug specific differentiation are trained. A predictor is created for each drug combination in the training set. As the model annotations related to drugs are constant, only the cell line's specific data has to be considered. Hence, you obtain one predictor for each cell line (83 in total) plus one for each drug pair. The synergy prediction of unknown drug-cell combinations will involve a check, which expert suits the task. However, this more feasible strategy has its limitations: if both, drug combination and cell line are not present in the training set, a synergy prediction is impossible.

The artificial neural network design can be assisted by prior knowledge about pathway interaction. The annotations associated to the GSEA models represent the training data of the present machine learning problem. The vertices are interpreted as input for the artificial neural network. The ANN is composed of the input layer, the first hidden layer that is referred to as the hidden module layer, the second layer that is called the hidden interaction layer and the output layer that contains the sole synergy node. The hidden module layer consists of nodes that are connected to the vertex of one particular GSEA model, so there is a one to one relationship between modules and nodes.

The design of the hidden interaction layer is supported by the use of pathway interaction data. The network design is shown in figure 7.5. The idea is that during the training procedure the links between the GSEA models will converge to a sort of pathway interaction network. In order to guide this process, prior knowledge about pathway interactions can be utilised. Pathway interactions are assumed to be sufficiently independent from the drug pairs and the cell lines that make the training sets. However some mutations and cancer differentiation might even affect dependencies between pathways and there is the risk of introducing some noise. Though, based on this assumption, an interaction analysis can optimize the structure of the artificial neural network predictor and assist the learning algorithm. GSEA sets are clustered into superbins, which do not have to be necessarily disjoint. The clusters contain gene sets that are supposed to work closely together or show high interdependency.

The network structure consists of four layers: the input layer (top layer), the hidden module layer, the hidden interaction layer and the output layer (the sole bottom node). There are two types of hidden nodes regarding the linkage structure: those forming the hidden module layer and those forming the hidden interaction layer. The nodes in the hidden module layer are associated with one particular GSEA module and assesses the information of the modules vertex only. The linkage of the nodes in the hidden interaction layer is guided by prior knowledge about pathway interactions. Reactome curated gene sets that show strong relation to each other are linked together. Hidden interaction nodes are only linked to their corresponding GSEA hidden module nodes. GSEA sets that have their own cluster and show no strong correlation are directly connected to the output node. The advantage of this approach is that linkage of unrelated or even isolated modules can be minimized and the search space of the training process is reduced significantly.

The maximal size of the artificial network is determined by the number of modules and the dimensionality of their vertices. A reasonable network of the proposed structure might have about 3000 input nodes, 600 hidden nodes and one output node if all modules are taken into account. However, it is expected that the majority of these modules do not vary significantly in the training data. Hence, many nodes might be irrelevant and can be neglected. Feasibility has to be checked in practice

and possibly dimensionality reduction techniques have to be applied.

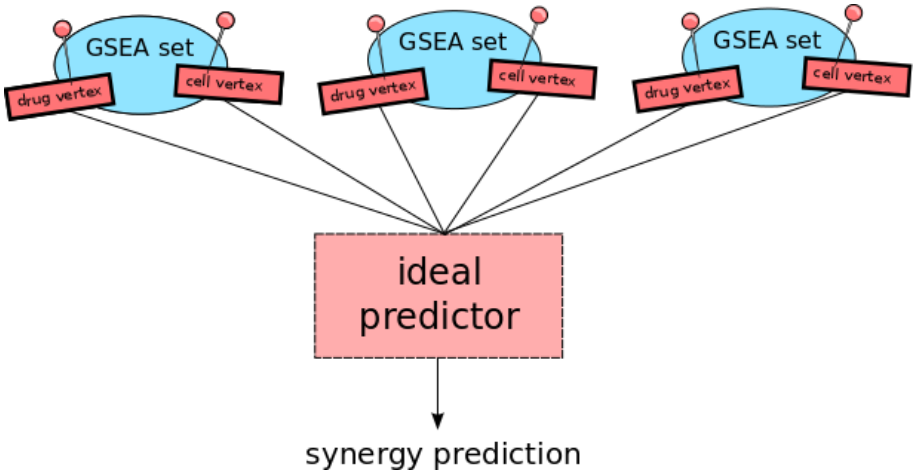


Figure 7.3: An ideal predictor would integrate the different vertices all at once. In order to make the approach feasible, there is a need to modify the approach and make the predictor use only one type of vertex.

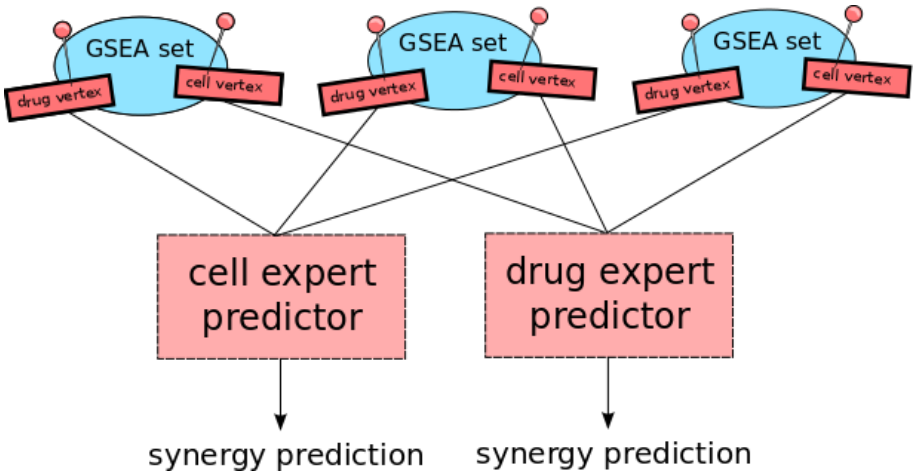


Figure 7.4: In this proposal, several differentiated predictors are trained which are experts in their specific cell or drug pair context. Expert predictors are only confronted with either drug or cell related data.

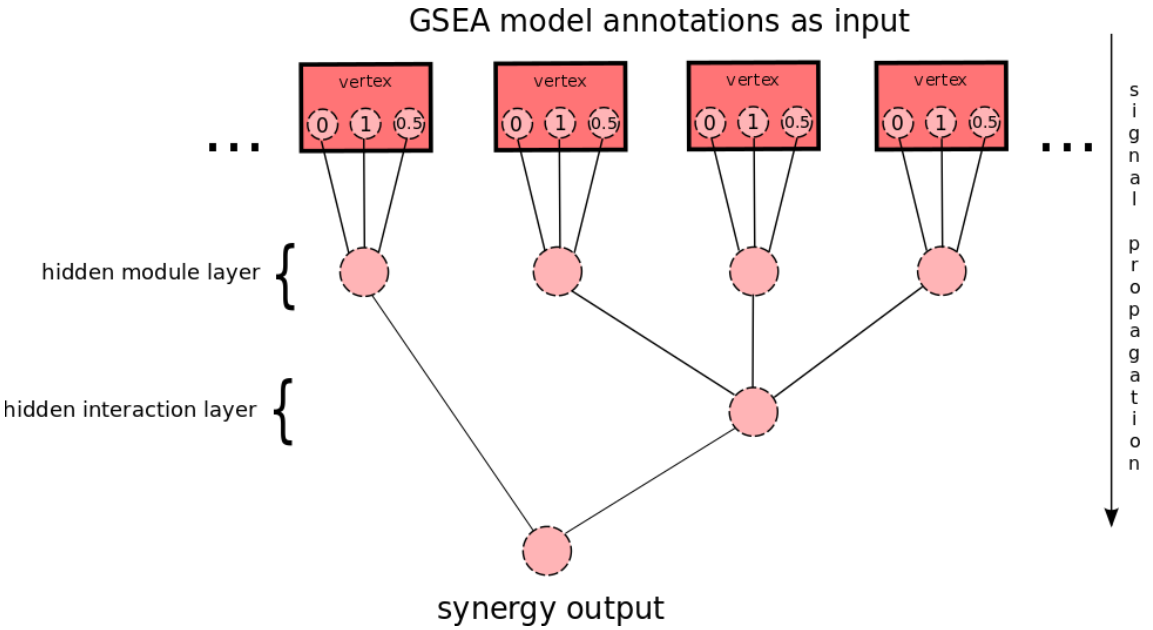


Figure 7.5: The structure of the artificial neural network is presented. The hidden nodes are connected to a bias node which is not shown in the illustration.

Chapter 8

Conclusion

Statistical knowledge inference suffers from limitations when dealing with big data, sparsity, high-dimensionality and lack of foreknown samples posing new requirements on data analysis tools. The aim of this project is to develop a computational method based on machine learning to classify proteins from the Gene Ontology Consortium (GOC) in order to find specific DNA binding RNA polymerase II transcription factors (DbTFs). A number of dimensional reduction techniques like PCA, PLS and LPLS are assessed for suitability. A simple but effective method for variable selection called separability analysis is proposed and tested. 2655 candidate proteins of the species human, mouse and Norway rat are classified with a voting committee of 1500 trained artificial neural networks. Out of these candidate proteins 54 proteins are identified as DbTFs with a relative classification error of less than 10%.

The multivariate data analysis is validated by cross-validation and ROC. During training 98% of all validation instances could be correctly classified though there are seven times more validation instances than training instances. Over-training was tried to be minimised for the particular data. Nevertheless the scope of this validation is limited to the available data and it cannot be excluded that some level of over-training stayed undetected due to the limited amount of known DbTFs. In addition the way how the knowledge bases of the Gene Ontology Consortium have been constructed might introduce a bias to the data that complicates prediction of unknown DbTFs. This makes the gene ontology data less useful for DbTF prediction. The ontology information includes large amounts of inferred knowledge that already contains a high level of interpretation. This weakens conclusions about the physical / chemical mechanisms behind DbTFs. Finally, it is difficult to suggest a distinct validation of the classification results outside the epistemic limitations of the available data.

The domain of 2655 candidates from the TFcheckpoint database is challenging and cumbersome to be studied experimentally. However, the computational inference performed in this project indicates a high probability for 54 proteins to have DbTF activity. For each of the candidates in this selection the relative classification error is less than 10%. These candidates need experimental confirmation by biological specialists for their possible DbTF-status in the next step. The outcome of this project can be used for instance as a priority measure to design further experiments.

Bibliography

[1] Hervé Abdi. Partial least square regression (pls regression). *Encyclopedia for research methods for the social sciences*, pages 792–795, 2003.

[2] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[3] Anne-Laure Boulesteix. Pls dimension reduction for classification with microarray data. *Statistical applications in genetics and molecular biology*, 3(1):1–30, 2004.

[4] Wei-Chien Chang. On using principal components before separating a mixture of two multivariate normal distributions. *Applied Statistics*, pages 267–275, 1983.

[5] Konika Chawla, Sushil Tripathi, Liv Thommesen, Astrid Lægreid, and Martin Kuiper. Tfcheckpoint: a curated compendium of specific dna-binding rna polymerase ii transcription factors. *Bioinformatics*, 29(19):2519–2520, 2013.

[6] Jorge De La Calleja and Olac Fuentes. Machine learning and image analysis for morphological galaxy classification. *Monthly Notices of the Royal Astronomical Society*, 349(1):87–93, 2004.

[7] Ulrik de Lichtenberg, Thomas S Jensen, Lars J Jensen, and Søren Brunak. Protein feature based identification of cell cycle regulated proteins in yeast. *Journal of molecular biology*, 329(4):663–674, 2003.

[8] Pierre Demartines and Jeanny Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *Neural Networks, IEEE Transactions on*, 8(1):148–154, 1997.

[9] Edgar Erwin, Klaus Obermayer, and Klaus Schulten. Self-organizing maps: ordering, convergence properties and energy functions. *Biological cybernetics*, 67(1):47–55, 1992.

[10] Åsmund Flobak, Anaïs Baudot, Elisabeth Remy, Liv Thommesen, Denis Thieffry, Martin Kuiper, and Astrid Lægreid. Discovery of drug synergies in gastric cancer cells predicted by logical modeling. *PLoS Comput Biol*, 11(8):e1004426, 2015.

[11] John A Flores. *Focus on artificial neural networks*. Nova Science Publishers, 2011.

[12] Pedro P Gallego, Jorge Gago, and Mariana Landín. *Artificial neural networks technology to model and predict plant biology process*. INTECH Open Access Publisher, 2011.

[13] Hanoch Gutfreund and Gerard Toulouse. The physics of neural networks. *Spin Glasses and Biology (ed. by DL Stein), World Scientific, Singapore*, pages 7–60, 1992.

[14] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[15] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):993–1001, 1990.

[16] Muditha M Hapudeniya. Artificial neural networks in bioinformatics. *Sri Lanka Journal of Bio-Medical Informatics*, 1(2):104–111, 2010.

[17] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[18] Tom Howley, Michael G Madden, Marie-Louise O'Connell, and Alan G Ryder. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data. *Knowledge-Based Systems*, 19(5):363–370, 2006.

[19] Christian Igel and Michael Hüsken. Improving the rprop learning algorithm. In *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*, volume 2000, pages 115–121. Citeseer, 2000.

[20] Andreas GK Janecek and Wilfried N Gansterer. A comparison of classification accuracy achieved with wrappers, filters and pca'. In *Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, 2008.

[21] L Juhl Jensen, Ramneek Gupta, Nikolaj Blom, D Devos, J Tamames, Can Kesmir, Henrik Nielsen, Hans Henrik Stærfeldt, Krzysztof Rapacki, Christopher Workman, et al. Prediction of human protein function from post-translational modifications and localization features. *Journal of molecular biology*, 319(5):1257–1265, 2002.

[22] Lars Juhl Jensen, Ramneek Gupta, H-H Staerfeldt, and Søren Brunak. Prediction of human protein function according to gene ontology categories. *Bioinformatics*, 19(5):635–642, 2003.

[23] Javed Khan, Jun S Wei, Markus Ringner, Lao H Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R Antonescu, Carsten Peterson, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, 7(6):673–679, 2001.

[24] Stanislav Kolenikov, Gustavo Angeles, et al. The use of discrete data in pca: theory, simulations, and applications to socioeconomic indices. *Chapel Hill: Carolina Population Center, University of North Carolina*, pages 1–59, 2004.

[25] Harvey Lodish, Arnold Berk, S Lawrence Zipursky, Paul Matsudaira, David Baltimore, James Darnell, et al. *Molecular cell biology*, volume 4. WH Freeman New York, 2000.

[26] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[27] Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.

[28] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

[29] Nguyen CH Mohamed A, Hancock T and Mamitsuka H. Netpathminer: R/bioconductor package for network path mining through gene expression. *Bioinformatics*, 30:3139–3141, 2014.

[30] Kyaw-Zeyar Myint, Lirong Wang, Qin Tong, and Xiang-Qun Xie. Molecular fingerprint-based artificial neural networks qsar for ligand biological activity predictions. *Molecular pharmaceutics*, 9(10):2912–2923, 2012.

[31] Shivashankar H Nagaraj and Antonio Reverter. A boolean-based systems biology approach to predict novel genes associated with cancer: Application to colorectal cancer. *BMC systems biology*, 5(1):1, 2011.

[32] Danh V Nguyen and David M Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18(1):39–50, 2002.

[33] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[34] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[35] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 1995.

[36] Solve Sæbø, Trygve Almøy, Arnar Flatberg, Are H Aastveit, and Harald Martens. Lpls-regression: a method for prediction and classification under the influence of background information on predictor variables. *Chemometrics and Intelligent Laboratory Systems*, 91(2):121–132, 2008.

[37] Solve Sæbø, Magni Martens, and Harald Martens. Three-block data modeling by endo-and exo-lpls regression. In *Handbook of Partial Least Squares*, pages 359–379. Springer, 2010.

[38] Victor Seguritan, Nelson Alves Jr, Michael Arnoult, Amy Raymond, Don Lorimer, Alex B Burgin Jr, Peter Salamon, and Anca M Segall. Artificial neural networks trained to detect viral and phage structural proteins. 2012.

[39] Sigurdur Sigurdsson, Peter Alshede Philipsen, Lars Kai Hansen, Jan Larsen, Monika Gniadecka, and Hans Christian Wulf. Detection of skin cancer by classification of raman spectra. *Biomedical Engineering, IEEE Transactions on*, 51(10):1784–1793, 2004.

[40] Maciej Szaleniec. Prediction of enzyme activity with neural network models based on electronic and geometrical features of substrates. *Pharmacological Reports*, 64(4):761–781, 2012.

[41] Roy Varshavsky, Assaf Gottlieb, Michal Linial, and David Horn. Novel unsupervised feature filtering of biological data. *Bioinformatics*, 22(14):e507–e513, 2006.

[42] Charles J Vaske, Stephen C Benz, J Zachary Sanborn, Dent Earl, Christopher Szeto, Jingchun Zhu, David Haussler, and Joshua M Stuart. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm. *Bioinformatics*, 26(12):i237–i245, 2010.

[43] Aravind Venkatesan, Sushil Tripathi, Alejandro Sanz de Galdeano, Ward Blondé, Astrid Lægreid, Vladimir Mironov, and Martin Kuiper. Finding gene regulatory network candidates using the gene expression knowledge base. *BMC bioinformatics*, 15(1):386, 2014.

[44] Shimon Whiteson and Daniel Whiteson. Machine learning for event selection in high energy physics. *Engineering Applications of Artificial Intelligence*, 22(8):1203–1217, 2009.

[45] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130, 2001.

[46] Ying Xu, Richard J Mural, J Ralph Einstein, Manesh B Shah, and Edward C Uberbacher. Grail: a multi-agent neural network system for gene identification. *Proceedings of the IEEE*, 84(10):1544–1552, 1996.

[47] Ka Yee Yeung and Walter L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

Chapter 9

Appendix: Candidate classification results

Table 9.1: The classification results are presented for predicted DbTFs that have a smaller relative error than 10%. A regression value of +1 corresponds to a positive classification, -1 to a negative one.

protein symbol	Dbxref key	protein name	regression result	accuracy estimate	relative error [%]
RAD21	O60216	Double-strand-break repair protein rad21 homolog	0.9994	0.0039	0.39
ZNF498	B3KPY4	Zinc finger protein 498, isoform CRA_a	0.9992	0.0045	0.45
EIF5B	O60841	Eukaryotic translation initiation factor 5B	0.9990	0.0055	0.55
POLR2K	P53803	DNA-directed RNA polymerases I, II, and III subunit RPABC4	0.9990	0.0055	0.55
Arid4b	Q9JKB5	AT-rich interactive domain-containing protein 4B	0.9986	0.0072	0.72
Rgs14	O08773	Regulator of G-protein signaling 14	0.9986	0.0072	0.72
Gatad2b	Q4V8E1	GATA zinc finger domain containing 2B	0.9980	0.0085	0.85
Sin3b	B0BNJ0	Protein Sin3b	0.9979	0.0087	0.87
ZNF479	Q96JC4	Zinc finger protein 479	0.9976	0.0089	0.89
ZSCAN31	Q96LW9	Zinc finger and SCAN domain-containing protein 31	0.9977	0.0089	0.89
ZNF471	Q9BX82	Zinc finger protein 471	0.997	0.010	1.00
ZNF512	Q96ME7	Zinc finger protein 512	0.997	0.011	1.10
ASH2L	Q9UBL3	Set1/Ash2 histone methyltransferase complex subunit ASH2	0.997	0.011	1.10
CBX4	O00257	E3 SUMO-protein ligase CBX4	0.997	0.012	1.20
Smarcd3	Q5U3Y2	Protein Smarcd3	0.996	0.013	1.31
HEL-S-1	V9HWD6	Epididymis secretory protein Li 1	0.996	0.014	1.41
TRIM28	Q13263	Transcription intermediary factor 1-beta	0.996	0.014	1.41
AATF	Q9NY61	Protein AATF	0.997	0.015	1.50
CTDP1	Q9Y5B0	RNA polymerase II subunit A C-terminal domain phosphatase	0.993	0.017	1.71
Ldb1	D3ZT89	LIM domain binding 1 (Predicted)	0.994	0.019	1.91
PA2G4	Q9UQ80	Proliferation-associated protein 2G4	0.997	0.019	1.91
Sebox	Q9ERS8	Homeobox protein SEBOX	0.992	0.020	2.02
GATAD2A	Q86YP4	Transcriptional repressor p66-alpha	0.992	0.020	2.02
Tfpt	Q9JMG6	TCF3 fusion partner homolog	0.994	0.021	2.11
POLR2H	P52434	DNA-directed RNA polymerases I, II, and III subunit RPABC3	0.992	0.023	2.32
Snapc5	D3ZTK9	Protein Snapc5	0.997	0.023	2.31
CRK	P46108	Adapter molecule crk	0.992	0.023	2.32
NOTCH1	P46531	Neurogenic locus notch homolog protein 1	0.991	0.025	2.52
ZNF524	Q96C55	Zinc finger protein 524	0.989	0.026	2.63
Mbtps2	D3ZDS6	Protein Mbtps2	0.987	0.027	2.74
RFX6	Q8HWS3	DNA-binding protein RFX6	0.992	0.027	2.72

Table continues on the next page

Table 9.1: The classification results are presented for predicted DbTFs that have a smaller relative error than 10%. A regression value of +1 corresponds to a positive classification, -1 to a negative one.

protein symbol	**Dbxref key**	**protein name**	**regression result**	**accuracy estimate**	**relative error [%]**
RUVBL2	Q9Y230	RuvB-like 2	0.990	0.027	2.73
DBX1	A6NMT0	Homeobox protein DBX1	0.989	0.030	3.03
ZNF682	O95780	Zinc finger protein 682	0.986	0.032	3.25
ZNF643	D3DPV3	Zinc finger protein 643, isoform CRA_b	0.986	0.038	3.85
ZNF814	B7Z6K7	Putative uncharacterized zinc finger protein 814	0.986	0.038	3.85
Id1	P41135	DNA-binding protein inhibitor ID-1	0.984	0.038	3.86
OSR1	Q8TAX0	Protein odd-skipped-related 1	0.987	0.042	4.26
Etv3l	F7FJQ8	Protein Etv3l	0.985	0.042	4.26
SART3	Q15020	Squamous cell carcinoma antigen recognized by T-cells 3	0.991	0.048	4.84
Tsg101	Q6IRE4	Tumor susceptibility gene 101 protein	0.984	0.049	4.98
CITED2	D9ZGF1	Cbp/p300-interacting transactivator, (. . .)	0.979	0.050	5.11
Zfp157	D3ZFZ1	Protein Zfp157	0.973	0.058	5.96
Smarcal1	B4F769	SWI/SNF-related matrix-associated actin-dependent (. . .)	0.987	0.062	6.28
Hdac8	B1WC68	Histone deacetylase 8	0.975	0.063	6.46
Mllt11	Q5M971	Protein AF1q	0.980	0.065	6.63
CHD9	Q3L8U1	Chromodomain-helicase-DNA-binding protein 9	0.969	0.065	6.71
Cdk5	Q03114	Cyclin-dependent-like kinase 5	0.964	0.072	7.47
TRIP4	Q15650	Activating signal cointegrator 1	0.984	0.074	7.52
Hoxd12	D3ZSN2	Protein Hoxd12	0.965	0.088	9.12
Dmrtb1	D4A494	Protein Dmrtb1	0.962	0.092	9.56
RPS6KA4	O75676	Ribosomal protein S6 kinase alpha-4	0.964	0.093	9.65
SNIP1	Q8TAD8	Smad nuclear-interacting protein 1	0.969	0.094	9.70
GTF2E2	P29084	Transcription initiation factor IIE subunit beta	0.959	0.094	9.80