Inaugural–Dissertation

submitted to the Combined Faculties for the
Natural Sciences and for Mathematics
of the
Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences
(Dr. rer. nat.)

presented by

Dipl.–Bioinf. (FH) Marc Johannes,
born in Bad Kreuznach, Germany
Oral examination: .....

Integration of Prior Biological Knowledge into Support Vector Machines

Referees Prof. Dr. Roland Eils
Prof. Dr. Tim Beißbarth

Abstract

One of the goals of high-throughput gene expression studies in cancer research is to identify prognostic gene signatures which have the potential to predict the clinical outcome of cancer patients. This is commonly investigated using classification methods. However, standard methods show only limited success since they merely rely on gene expression data and assume genes to be independent. Nevertheless, recent studies have shown that the classification can be improved in terms of accuracy as well as interpretability and reproducibility of prognostic gene signatures by including *prior* biological knowledge, such as information about known cellular signalling pathways.

This work gives an overview on databases storing data that is appropriate for use as *prior* knowledge as well as existing algorithms capable of using this data. The utility of these methods in practice is demonstrated on a number of examples for predicting the clinical outcome of patients.

A new classification method capable of using *prior* knowledge about feature connectivity was developed. The Support Vector Machine (SVM) in combination with the Recursive Feature Elimination (RFE) algorithm were selected as basis of the new method. This combination allows to select the features that are most important for the classification. However, RFE selects these features merely based on their influence on the hyperplane found by the SVM. The novel algorithm, called Reweighted Recursive Feature Elimination (RRFE), alters this ranking criterion by combining the RFE weight with a second weight coming from GeneRank. GeneRank is a modified version of Google's PageRank algorithm and calculates a score for each gene based on a graph structure build from a protein-protein interaction (PPI) database.

The assumption of RRFE is that a gene with a low fold change should have an increased influence on the classifier if it is connected to differentially expressed genes. The combination of GeneRank and RFE gives highly connected genes the chance to influence the classifier and in turn help deciphering the underlying biological process. Thus, RRFE accounts for the fact that many functionally relevant genes might not be detectable with current techniques and hence decrease the amount of unexploited information in the data.

Abstract

RRFE was evaluated on four breast cancer data sets, as well as on an integrated one with almost 800 samples. Different clinical endpoints relevant to breast cancer were predicted, including the *ERBB2* status as well as the risk of relapse. RRFE demonstrated its ability to select genes that are correlated with the intrinsic biology of the disease, i.e. the selected genes are significantly associated with cancer-related pathways. This improved interpretability is important since it facilitates the biological understanding. Furthermore, RRFE could improve the stability of gene-signatures and increase the classification performance both compared to standard and pathway-based classification methods.

Besides the theoretical foundations of RRFE, a new R-package containing RRFE as well as two other, recently published, pathway-based classification methods is presented. The package contains all methods needed to perform a benchmark of newly developed algorithms, for assessing differences in classification performance and extracting the genes used by the methods to build the decision rules.

Zusammenfassung

Ein Ziel der klinischen Krebsforschung ist es, neue, prognostische Gensignaturen zu finden, die den klinischen Verlauf der Krankheit vorhersagen können. Um neue Gensignaturen oder Biomarker zu identifizieren, nutzt man in der Bioinformatik oft Klassifikationsmethoden. Allerdings verwenden die üblicherweise eingesetzten Verfahren ausschließlich Genexpressionsdaten und sehen Gene als unabhängig an. Mehrere, vor kurzem veröffentlichte, Studien konnten jedoch zeigen, dass sich die Qualität der Klassifikation steigern lässt, wenn man Netzwerkwissen in den Klassifikationsprozess einfließen läßt. Neben einem verbesserten Klassifikationsergebnis wurde auch gezeigt, dass die ausgewählten Gene besser zu interpretieren sind und dass die Selektion der Gene stabiler wird.

Aus diesen Gründen beschäftigt sich die vorliegende Arbeit mit Methoden, die die Vorhersagegenauigkeit verbessern indem sie neben Genexpressionsdaten auch Netzwerkwissen für die Klassifikation berücksichtigen. Die Arbeit gibt einen Überblick über bestehende Methoden, die in der Lage sind, Netzwerkwissen in die Klassifikation einfließen zu lassen sowie über Datenbanken die solches Wissen speichern.

Außerdem beschreibt die Arbeit die Entwicklung einer neuen, netzwerkbasierten Klassifikationsmethode, die in der Lage ist, die Konnektivität der Gene zu berücksichtigen. Die 'Support Vector Machine' (SVM) wurde als Grundlage des neuen Algorithmus ausgewählt. Normalerweise ist die SVM nicht in der Lage eine Genselektion durchzuführen, d.h. sie nutzt immer alle Gene um einen bestimmten Endpunkt vorherzusagen. Man kann die SVM allerdings mit dem 'Recursive Feature Elimination' (RFE) Algorithmus kombinieren, um eine Genselektion zu ermöglichen. RFE selektiert Gene anhand ihres Einflusses auf die, von der SVM gefundenen Hyperebene.

Das Sortierkriterium von RFE wurde mit einer modifizierten Version von Google's PageRank-Algorithmus verändert. Die abgewandelte Version von PageRank nennt sich GeneRank und errechnet, basierend auf einem Graphen der aus einer Protein-Protein Interaktionsdatenbank erstellt wurde, ein Gewicht für jedes Gen. Dieses Gewicht wurde mit dem Sortierkriterium

von RFE kombiniert, um das Netzwerkwissen in die Sortierung der Gene und damit in die Klassifikation zu integrieren. Wegen dieser Neugewichtung wurde der neuentwickelte Algorithmus 'Reweighted Recursive Feature Elimination' (RRFE) genannt.

RRFE verfolgt die Annahme, dass Gene, die nur eine geringe Änderung in ihrer Expression aufweisen, die Chance haben sollten einen gesteigerten Einfluss auf die Klassifikation zu nehmen, wenn sie stark vernetzt sind. Diese Annahme wurde durch die Kombination von GeneRank und RFE umgesetzt. Dadurch hilft RRFE den zugrundeliegenden, biologischen Vorgang besser zu verstehen. Außerdem trägt RRFE dazu bei, den Anteil an ungenutzen Informationen in den Daten zu verringern und funktionell wichtige Gene zu identifizieren.

RRFE wurde auf einem integrierten und vier unabhängigen Brustkrebsdatensätzen getestet. Die Datensätze bestehen zusammen aus fast 800 Patieten. RRFE wurde verwendet, um den ERBB2-Status sowie das Risiko eines Brustkrebsrückfalls vorherzusagen. In den Analysen zeigte sich eine verbesserte Interpretierbarkeit und Stabilität der selektierten Gene. Desweiteren konnte auch die Genauigkeit der Klassifikation gegenüber standard- sowie netzwerkbasierten Klassifikatoren gesteigert werden.

Neben den theoretischen Grundlagen von RRFE stellt die Arbeit auch ein neues R-Paket vor, welches die Implementierungen von RRFE und weiterer netzwerkbasierter Klassifikationsmethoden enthält. Ziel war es, die Nutzung von RRFE und anderen Methoden zu vereinfachen, um Entwicklern die Möglichkeit zu geben, die Güte ihrer neuentwickelten Algorithmen mit bereits bestehenden Verfahren zu vergleichen. Das Software-Paket beinhaltet Funktionen, welche zum Vergleichen von Klassifikationsmethoden, dem Erstellen von Grafiken und zur Indentifizierung von Genen, die maßgeblich zur Klassifikation beigetragen haben, nötig sind.

Table of Contents

List of Figures

List of Figures

List of Tables

List of Tables

List of Abbreviations

AUC	Area under the ROC Curve
cf.	confer
CPDB	ConsensusPathDB
CV	Cross-validation
ERBB2	human epidermal growth factor receptor 2
GEO	Gene Expression Omnibus
GO	The Gene Ontology
HPRD	Human Protein Reference Database
i.e.	*id est*
KEGG	Kyoto Encyclopedia of Genes and Genomes
KKT	Karush-Kuhn-Tucker
loo	leave-one-out
RFE	Recursive Feature Elimination
ROC	Receiver Operator Characteristic
RRFE	Reweighted Recursive Feature Elimination
SVM	Support Vector Machine
w.r.t.	with respect to

List of Abbreviations

Chapter 1

Introduction

1.1 Clinical Cancer Research

The genomes of mammalian cells carry all information needed to create a molecular machinery that regulates proliferation, differentiation and apoptosis (Ponting, 2008). However, genomes of cells are altered by various mechanisms which can lead to mutations of encoded genes. These mutations range from point-mutations to translocation of whole chromosomes. Due to these changes, cells can acquire new phenotypes which progressively drive the transformation of normal cells into malignant neoplasms (Preston-Martin et al., 1990). Furthermore, it is anticipated that tumorigenesis is a multi-step process that needs several alterations to take place (figure 1.1, Hanahan and Weinberg 2000, 2011). Once cells have overcome the defense mechanisms that usually work against these characteristics, malignant growth arise.

Two main classes of tumors are known: benign– and malignant tumors. Benign tumors grow only locally confined and do not invade adjacent tissues, whereas malignant tumors grow more aggressive and do invade the nearby tissue. Furthermore, malignant tumors might release cancer cells into the blood stream which can reach the lymph nodes as well as distant sites of the body to finally form secondary tumors known as metastases. These

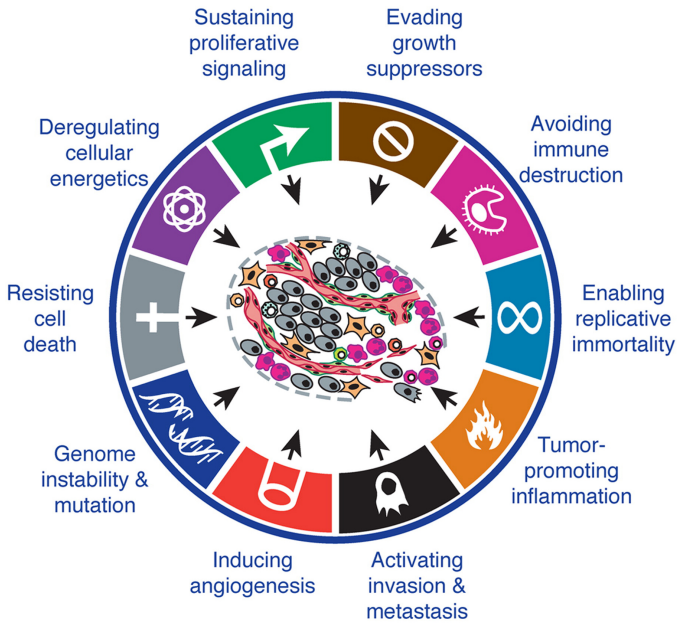

Figure 1.1: Capabilities acquired during tumorigenesis. (Reprinted from Hanahan and Weinberg (2011), Copyright (2011), with permission from Elsevier)

metastases spawned by the primary tumor are responsible for approximately 90% of cancer-related deaths (Pisani et al., 1999).

Tumors are further classified dependent on the tissue they arise from (figure 1.2, Weinberg 2006). The majority of human cancers are carcinomas that emerge from epithelial tissue (Pisani et al., 1999; Jemal et al., 2010). Carcinomas are further classified into two subgroups: squamous cell carcinoma and adenocarcinoma. Squamous cell carcinoma arise from epithelial cells that form protective cell layers, i.e. they seal the cavity or channel that they line in order to protect underlying cells. The second class are adenocarcinoma which originate in epithelial cells that are specialized in secreting substances into the ducts or cavities they line.

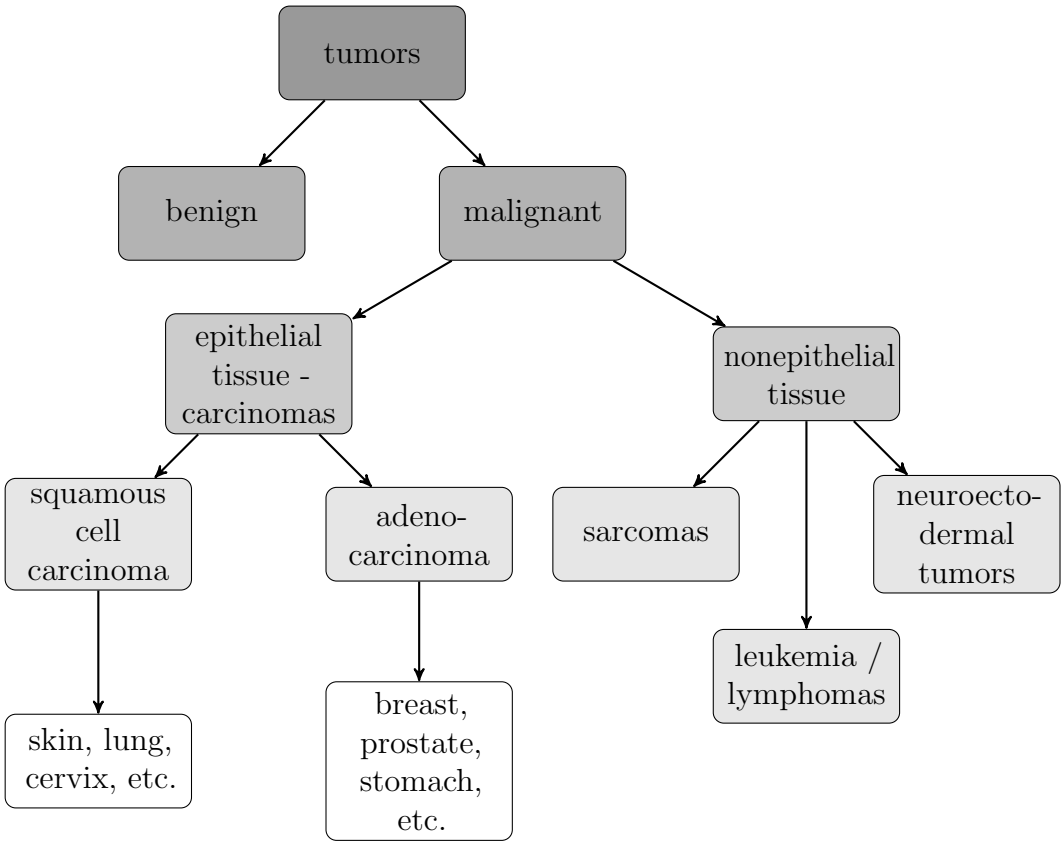

Figure 1.2: Classification of most common tumor types (according to Weinberg 2006). Most malignant tumors arise from epithelial tissues (approx. 80%). These so-called carcinomas are split into two groups: squamous cell carcinoma and adenocarcinoma. All other, non epithelial tumors, are assigned into three major groups: sarcomas, leukemia/lymphomas and neuroectordermal tumors like gliomas, etc.

1.1.1 Breast cancer

Breast cancer belongs to the class of adenocarcinoma and is by far the most common form of cancer in women (Jemal et al., 2010). Breast cancer often forms metastases and this has made it the second leading cause of cancer-related death in women (Weigelt et al., 2005).

Breast cancer is mostly diagnosed by mammography or breast examination. Once diagnosed, patients usually undergo surgery to remove the primary tumor. After surgery, clinico-pathological parameters are used to estimate

the progression status and the risk of recurrence. These estimates are used to decide whether a patient needs to undergo adjuvant treatment, which usually consists of systemic chemotherapy, radiotherapy or targeted treatment. The adjuvant treatment aims at eliminating all microscopic cancer cells and decreasing the risk of recurrence. However, due to the heterogeneity of breast cancer current clinico-pathological markers like lymph node status, tumor size or differentiation status are not appropriate for predicting the aggressiveness of the disease (Tavassoli and Devilee, 2003; Carter et al., 1989; Elston and Ellis, 1991). Indeed, women with the same clinico-pathological characteristics can have a notably different courses of disease. Hence, lots of patients are overtreated and suffer from the substantial side-effects of chemotherapy (Eifel et al., 2001). Therefore, new prognostic markers, that are able to estimate the probability of recurrence at the time of diagnosis are urgently needed.

It is widely accepted that molecular alterations lead to cancer-development (Garnis et al., 2004). Microarray technologies allow to measure the expression of thousands of genes in parallel. By associating these expression profiles with the clinical outcome of patients, new biomarkers can be discovered. Due to the heterogeneity of breast cancer this approach is more promising than just correlating a few clinico-pathological markers or combinations thereof to the course of disease (Weigelt et al., 2005). The aim is to use gene expression profiles for tailored adjuvant therapy.

Retrospective studies are commonly applied to identify novel prognostic markers for improving risk stratification of breast cancer patients (figure 1.3). To this end, biomolecules are extracted from surgical tumor specimens of cancer patients. In particular, extracted RNA is mainly used to study large scale gene expression profiles using DNA-microarrays. In the retrospective study design, RNA profiles can be correlated to long-term (5-10 years) clinical follow-up data of breast cancer patients. Relapse events (local recurrence, distant metastases) are commonly used as the primary clinical endpoint to identify novel molecular biomarkers using bioinformatic analyses. These analyses commonly consist of unsupervised (clustering) or supervised learning (classification) approaches. In the ideal case, markers identified by

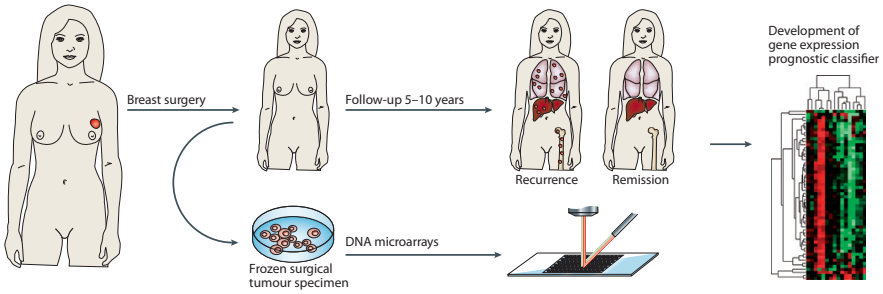

Figure 1.3: Course of a retrospective study for the development of a prognostic marker. (Adapted by permission from Macmillan Publishers Ltd: Nature Reviews Cancer (Sotiriou and Piccart, 2007), copyright (2007))

retrospective studies should also be tested in an independent prospective study (Ransohoff, 2005).

In the early twenty-first century, microarray studies led to the discovery of breast cancer subgroups. By applying unsupervised bioinformatic analyses to gene expression data, Perou et al. (2000) found portraits of four molecular different breast cancer subtypes: ERBB2-positive, normal breast-like, luminal and basal-like. ERBB2 positive breast cancer patients carry a characteristic amplification of a region on chromosome 17 that includes the *ERBB2* gene. Due to this amplification *ERBB2* itself but also adjacent genes as well as the ERBB2 pathway are overexpressed. The normal breast-like subtype expresses genes of non-epithelial cell origin. Luminal breast cancer samples are characterized by a high expression of the estrogen receptor (ER) and coregulated genes as well as other specific markers of luminal epithelial cells. The luminal subtype was later divided into luminal A and B, where subtype B has a lower expression of the ER-coregulated genes and a higher rate of proliferation associated with an adverse prognosis (Sorlie et al., 2001). The basal-like subtype commonly shows high EGFR expression and a loss of expression of the ERBB2–, progesterone– and estrogen receptors, respectively. It is important to note, that independent gene expression studies have confirmed that these breast cancer types are clinically distinct subgroups (Sorlie et al., 2003), since they show substantially different clinical outcome and response to treatment (Rouzier et al., 2005).

1.1.2 Existing markers for breast cancer prognosis

Several gene-signatures for breast cancer prognosis have been suggested in recent years. Van 't Veer et al. (2002) from the Netherlands Cancer Institute (NKI) in Amsterdam reported a multigene signature consisting of 70 genes (Mammaprint) that reliably predicts the likelihood of distant metastases in lymph node-negative tumors. Subsequent use of a cohort of 295 patients could validate the signature as being the best predictor for metastasis-free survival. Additionally, it has been shown that the signature is independent of factors like histological grade, age, tumor size and adjuvant treatment (van de Vijver et al., 2002).

Two years later, Wang et al. (2005) published a 76-gene signature obtained by a related approach as the one used by the group from Amsterdam. Although, both gene-signatures only had three genes in common, they showed similar performance and could be validated in an independent study with 302 patients conducted in the framework of the translational research network of the Breast International Group (Buyse et al., 2006; Desmedt et al., 2007).

It needs to be stressed, that one of the main reasons for the small degree of concordance in gene-signatures are correlation structures, inherently present in microarray measurements. Briefly, if a gene is highly correlated to the clinical outcome and thus is a good marker, all other genes correlated to that gene are in turn also good predictors of clinical outcome. However, depending on the patients present in individual training sets, this correlation might vary and hence the rank of correlated genes is highly unstable. This leads to unstable gene signatures that have only a few genes in common (Ein-Dor et al., 2005). In addition, the utilization of different microarray platforms for measuring the gene expression might also lead to a decreased reproducibility. Further sources of variation might be differences in bioinformatic algorithms used for normalization and marker discovery. Another reason for the small overlap is the limited statistical power, i.e. too small sample sizes for training and testing of algorithms in order to identify disease-associated genes (Ein-Dor et al., 2006).

Although, individual signatures for breast cancer prognosis contain different genes, their ability to predict the outcome on independent patient cohorts is similar (Tan et al., 2003; Fan et al., 2006). Nevertheless, individual genes, present in these signatures, are not necessarily connected to the underlying disease. This hampers our understanding of underlying mechanisms. Hence, there is a pressing need to develop new algorithms capable of identifying gene-signatures correlated to the intrinsic biology that are still able to predict the course of cancer with high accuracy.

1.2 Biomarker discovery using bioinformatics

Microarray analyses have become a standard means for assessing genome-wide gene expression measurements of biological systems. Bioinformatics uses statistical, mathematical and computational methods for analyzing and processing the resulting data. Bioinformatic analyses are a crucial step for achieving biological understanding. Gene expression measurements of different classes of samples raise the vital question of how to discriminate these classes and how to determine meaningful biomakers, i.e. signatures of genes. Therefore, the development of novel bioinformatic algorithms is essential for increasing the accuracy of biomarkers and guide the biological understanding. In clinical cancer research, for example, it is known that most cancer treatments are only suited for a specific subgroup of patients. Therefore, bioinformatic algorithms can facilitate the development new *predictive* biomarkers that help to identify patients that would benefit from a certain treatment. Other classes of biomakers are *diagnostic* biomarkers, that help identifying the absence or presence of a disease, and *prognostic* makers determine the likelihood of a relapse (Biomarkers Definitions Working Group, 2001).

Given these diverse types of biomarkers and applications, an impressive collection of bioinformatic tools has been developed for identification and validation of new markers. These methods are either *supervised*, i.e. the

classes are known, or *unsupervised* when the class of individual samples is unknown. A well known class of unsupervised methods are *cluster algorithms* that can, for instance, be used for the identification of tumor subtypes. The class of supervised learning algorithms include *classification methods* that use patterns of carefully phenotyped samples to learn the characteristics of individual groups. Examples of these tools include algorithms like the support vector machine (SVM, Boser et al., 1992), k-nearest neighbors (kNN, Duda and Hart, 1973), the nearest shrunken centroid classifier (PAM, Tibshirani et al., 2002), decision trees (Quinlan, 1986) and many others (Dudoit et al., 2002).

However, an intrinsic problem that usually occurs when conducting microarray analyses is that the number of genes, present on the chip, is much larger compared to the number of patients included in the study. This problem is well known in the field of machine learning and sometimes referred to as the *curse of dimensionality* (Bellman, 1961). The large number of genes present on the microarray makes these analysis prone to the curse of dimensionality, since the classifier will most probably find a decision rule which works well on the training data. However, since most of the genes used by the decision rule are probably not, or only by chance, associated with the disease state, the performance of the decision rule is overestimate and it will perform worse on new samples.

One possibility to tackle this problem is merging the many available covariates into some few by using so-called *dimensionality reduction* algorithms. The most famous of these methods is probably the principal component analysis (Pearson, 1901). However, when molecular markers are sought this is not the preferable approach.

Another possibility to overcome the curse of dimensionality is to build the classifier exclusively on those genes that are of importance to the disease. However, these genes are not known *a priori* and, thus have to be selected by the learning algorithm. The task of selecting only a subset of genes is known as *feature selection* (see Guyon and Elisseeff, 2003, for an overview). However, genes composing the final signature are usually selected independently of

each other, although proteins are known to interact within protein complexes, signaling pathways, and higher-order cellular processes. The reason for this independent selection is that *standard* classification methods merely rely on gene expression data and score each gene individually for how well it discriminates different classes of a disease. Therefore, the final classifier may contain unnecessarily many genes with redundant information which may lead to decreased classification performance on new samples (Lee et al., 2008). This is also one possible explanation, why gene signatures have only a few genes in common, even if they are designed to predict the same clinical outcome (Ein-Dor et al., 2005). Despite the instability, these signatures are usually not easy to interpret since the membership in the gene-signature is not necessarily a indicator of the importance of that gene in, for example, cancer pathology (Weigelt et al., 2005).

1.2.1 Pathway-based classification methods

Recent studies have demonstrated that standard classification methods can be improved in terms of accuracy as well as stability of selected genes by including *a priori* knowledge of interactions into the classification process. Here, the term 'interactions' is rather loosely defined, i.e. it refers to any kind of interacting biological entities that might form a network, pathway or signalling-cascade. These pathways are used to build a graph structure with biological entities (i.e. genes or proteins) as vertices and edges representing any kind of interaction. The field of pathway-based classification is rapidly growing and several methods have already been described.

Chuang et al. (2007) integrate pathway knowledge from protein-protein interaction networks. Their algorithm randomly chooses sub-networks and assigns an activity score based on the expression level of the genes from the sub-net. Afterwards, sub-networks which are able to discriminate between the clinical endpoints are identified and subsequently used to build a classifier based on these networks.

Rapaport et al. (2007) define a new metric for gene expression measurements by using the matrix exponential function, which is similar to the *diffusion kernel* (Kondor and Lafferty, 2002). Their assumption is that most biologically relevant information is captured in the low-frequency component of expression profiles. Hence, the projection of the low-frequency component of an expression vector on the gene metabolic network should reveal areas of positive and negative expression on the graph that are likely to correspond to the activation or inhibition of specific branches of the graph.

The approach introduced by Zhu et al. (2009) is called network-based SVM and uses a network-based penalty which leads to a grouped variable selection. This variable selection is achieved by penalizing the SVM objective function with an F_∞-norm (Zou and Yuan, 2008), instead of the commonly used L_1 or L_2 penalization. This norm forces the simultaneous selection or elimination of a group of features from the same pathway. Zhu et al. (2009) treat neighboring genes in a graph as a group and construct their network-based penalty as the sum of F_∞-norms of groups of neighboring genes-pairs.

Yousef et al. (2009) introduced an algorithm which uses the Gene Expression Analysis Tool (GXNA, Nacu et al. 2007) to build clusters of genes which are connected. They use these clusters as input to a linear SVM, assign a weight to the clusters based on the importance to the classification and then remove the least informative clusters. The process of training the SVM and removing unimportant clusters is repeated until the maximum classification performance is reached. This algorithm is called Recursive Network Elimination (RNE), as it removes clusters of genes instead of removing single genes.

A method called PathBoost which is based on likelihood-based boosting was recently proposed by Binder and Schumacher (2009). Still others have been published by Bellazzi and Zupan (2007); Lee et al. (2008); Su et al. (2009).

1.2.2 Pathway databases for building graphs

The knowledge on interacting biological entities is usually stored in databases. Depending on the system described, different levels of background knowledge exist. The Gene Ontology (GO, Ashburner et al., 2000) consortium represents an initiative which provides a vocabulary on gene functions to facilitate the systematic usage of this knowledge. The graph associated with GO is a directed acyclic graph where the nodes are the vocabulary and the edges represent relations like 'is a' and 'part of'. The GO is structured hierarchically, it has three top level annotations, being:

molecular function describing the function of a gene, e.g. kinase, phosphatase or transcription factor;

biological process describing the process or pathway a molecule is involved in, e.g. cell death, cell cycle or MAP kinase pathway;

cellular component describing the part of a cell or cell structure in which a molecule is active, e.g. nucleus, ribosome or cell membrane.

Several databases have been created for storing and collecting gene-specific information in GO format. This data can be used to create a matrix of pairwise similarities or dissimilarities of genes. Subsequently, the matrix can be used to score the gene-gene interactions and incorporated into the biomarker discovery process. Several methods have been developed for this purpose (Fröhlich et al., 2007). An overview of methods for accessing and mining these annotations is given in Beißbarth (2004).

Although the GO initiative has been founded ten years ago, most information on gene function is still hard to mine, since it is not stored systematically. However, first attempts have been made to add GO-based meta-tags to publications (Vanteru et al., 2008; Doms and Schroeder, 2005). Another way is to manually curate the published information, as done by TransPath (Choi et al., 2004), Ingenuity (Ganter and Giroux, 2008) or Metacore (Ekins et al., 2007);

still others rely on text-mining tools for automated information extraction (Jensen et al., 2006; Agarwal and Searls, 2008).

Databases like KEGG (Kanehisa and Goto, 2000) or consensusPathDB (Kamburov et al., 2009), focus on the biological interactions defining the processes of living cells and summarize these in manually curated pathway models. There also exist more focused databases representing molecular interactions obtained by genomic techniques like transcription factor binding based on chip-chip data, e.g. TRANSFAC (Wingender, 2008) or JASPAR (Portales-Casamar et al., 2010), or protein-protein interactions based on co-immunoprecipitation or yeast two-hybrid screening, e.g. HPRD (Prasad et al., 2009), MINT (Ceol et al., 2010) or IntAct (Aranda et al., 2010).

1.3 Aim and organization of the thesis

The focus of this thesis was the development of methodology that enables classification algorithms to use graphs in combination with patient specific data for building decision rules and detecting biomarkers for risk prediction. We used the SVM, which is a supervised learning method that has shown its predominance over other methods and can easily handle high dimensional data (Furey et al., 2000; Brown et al., 2000). In combination with the recursive feature elimination algorithm (RFE, Guyon et al., 2002), the SVM is able to narrow down the number of genes needed to build the decision rule. However, this feature selection is merely based on mathematical criteria. Here, we tried to incorporate *prior* biological knowledge in the form of a graph structure to improve the classification performance and the interpretability of selected genes. The graphs needed for this algorithm can be build from any of the databases mentioned in the previous subsection (see Porzelius and Johannes et al., (2011)). The assumption was, that the novel algorithm benefits from the pathway knowledge since genes are no longer treated as independent.

To make the work more self-contained, the prerequisites and theoretical foundations needed to understand the results are outlined in chapter 2. In

section 2.1 the basics of SVMs are briefly introduced, before section 2.2 shows how feature selection is performed when using SVMs. Sections 2.3 and 2.4 deal with model assessment and introduce the Receiver Operator Characteristic for evaluating classifiers. Afterwards (in section 2.5), we show how genes can be ranked by using a modified version of Google's PageRank algorithm deployed on gene networks. The remainder of the chapter shows how the gene networks were created and introduces the breast cancer gene expression data sets, that were used for evaluating the algorithm.

Chapter 3 shows the results, starting with our newly developed algorithm called Reweighted Recursive Feature Elimination (RRFE, Johannes et al., 2010). Section 3.1.1 outlines the results obtained by using RRFE, i.e. that RRFE selects interpretable genes (section 3.1.1.1), increases the classification performance as well as the overlap between marker-genes in gene-signatures obtained from different experiments (section 3.1.1.2) and that it is predominant over other classifiers (3.1.1.5). The second part of the Results chapter deals with a software package that was developed to facilitate the usage of pathway-based classification methods (Johannes et al., 2011). We implemented the novel RRFE methods as well as two other methods that are able to use *prior* knowledge.

Chapter 2

Material and Methods

2.1 Support Vector Machines

2.1.1 Introduction

The goal of this section is to introduce support vector machines, the classification algorithm that has been used in this work. The support vector machine (SVM, Boser et al. 1992) is a statistical learning method for building classification models. It belongs to the separating hyperplane classifiers. These algorithms try to find a linear decision boundary which separates the data as well as possible.

The section is organized as follows: 2.1.2 introduces separating hyperplanes and section 2.1.3 will outline the maximum margin principle which is key for support vector machines. Afterwards, section 2.1.4 will show how to solve the support vector classification problem.

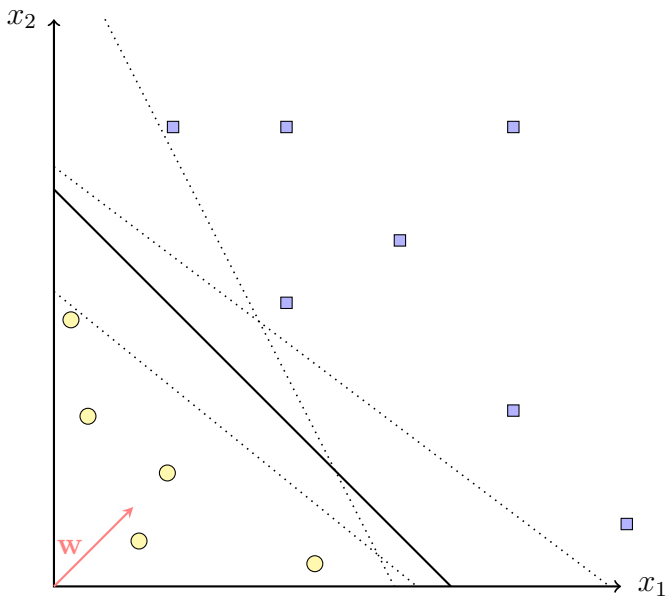

Figure 2.1: Example of a linear separating hyperplane in $\mathbb{R}^2$. The dotted lines indicate, that even when the data is perfectly separated by the black line, there are infinitely many other solutions.

2.1.2 Linear hyperplanes

Assume $\mathcal{H}$ being a Hilbert space and $\mathbf{x}_i, \ldots, \mathbf{x}_n \in \mathcal{H}$ is a set of pattern vectors with labels $y_i, \ldots, y_n \in \{\pm 1\}$. In $\mathcal{H}$ any hyperplane can be defined as

$$\{\mathbf{x} \in \mathcal{H} : f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0\}, \quad \mathbf{w} \in \mathcal{H}, b \in \mathbb{R}. \tag{2.1}$$

Here, $\mathbf{w}$ is the vector normal to the hyperplane and b is the offset from the origin. An example of such a hyperplane in $\mathbb{R}^2$ is given in figure 2.1. Since (2.1) separates $\mathcal{H}$ in two half-spaces of points classified as positive $\mathcal{H}^+ = \{\mathbf{x} : f(\mathbf{x}) \geq 0\}$ and negative $\mathcal{H}^- = \{\mathbf{x} : f(\mathbf{x}) < 0\}$ it corresponds to a *decision function*. Thus, a new sample with input vector $\mathbf{x}$ is assigned to class $\operatorname{sgn}(f(\mathbf{x})) = \operatorname{sgn}(\mathbf{w}^T\mathbf{x} + b)$. For a candidate function f one can check for each example $(\mathbf{x}_i, y_i)$ if it was correctly classified, i.e. $0 \leq y_i f(\mathbf{x}_i)$ or not. One possibility to choose f is called *empirical risk minimization*, that is, one tries to minimize the amount of wrongly made decisions on the whole set of examples $(\mathbf{x}_i, y_i), \quad i = 1, \ldots, n$. However, there are infinitely many

of such linear hyperplanes that perfectly separate the toy data as shown in figure 2.1. Therefore, one could additionally demand that the examples should be classified with strong confidence, which leads to the principle of maximum-margin hyperplanes.

2.1.3 Maximum margin principle

If one considers (2.1) it is obvious that there is still the freedom to multiply $\mathbf{w}$ and b with the same non-zero constant in order to recover an equivalent hyperplane. In order to remove this scaling freedom, the SVM searches for the *canonical* hyperplane. It maximizes the margin between both classes, i.e. the distance to the points closest to it. The canonical hyperplane is defined by the pair $(\widehat{\mathbf{w}}, \hat{b}) \in \mathcal{H} \times \mathbb{R}$ which is scaled such that the point closest to the hyperplane has a distance of $1/\|\mathbf{w}\|$:

$$\min_{i=1,\ldots,n} \left|\mathbf{w}^T \mathbf{x}_i + b\right| = 1 \tag{2.2}$$

for all $\mathbf{x}_i, \ldots, \mathbf{x}_n \in \mathcal{H}$. Thus, the width of the margin is exactly equal to $2/\|\mathbf{w}\|$. An example of a maximum-margin hyperplane is shown in figure 2.2. The idea behind maximum-margin hyperplanes is that making the margin as big as possible minimizes the bound on the risk (cf. Boser et al. 1992; Vapnik and Cortes 1995).

2.1.4 Support vector classification

The last two sections have introduced some of the foundations of SVMs. In this section it will be shown how the support vector classifier can be computed. In the ideal case one wants to classify all examples correctly with high confidence using a linear function like (2.1) with the constrains of (2.2). In a mathematical formulation this corresponds to maximizing $2/\|\mathbf{w}\|$ under the constraints $1 \leq y_i f(\mathbf{x}_i) = y_i(\mathbf{w}^T \mathbf{x}_i + b)$ for $i = 1, \ldots, n$. However, it is intuitive that this perfect separation of the data by a linear function might

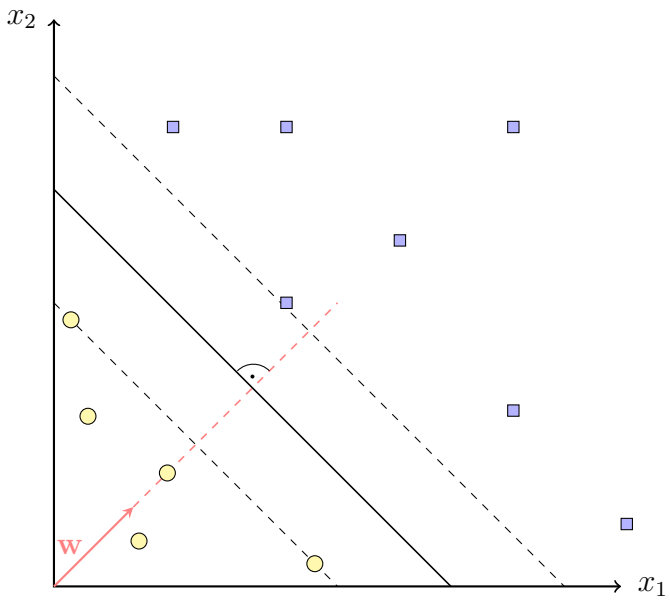

Figure 2.2: Maximum margin hyperplane. The dashed lines show the borders of the margin. Its width is exactly $^2/_{\|\mathbf{w}\|}$. The red line indicates the weight-vector of the hyperplane which is orthogonal to it. The examples that lie on the margin are called support-vectors.

not always be possible. This is particularly true for biological data, which is known to be quite noisy (Tilstone, 2003; Febbo and Kantoff, 2006). Thus, this section focuses on the *soft-margin* SVM implementation. In contrast to the *hard-margin* version it allows for some fraction of misclassification.

Even though a perfect classification might not be possible one wants to find the best possible solution. Therefore, a criterion to assess the quality of the estimate is needed. This is assessment is usually done by optimizing some functional. However, this type of function should fulfill certain criteria like having its minimum at zero, since a correct prediction should result in a zero penalty. Additionally, it should not only count misclassifications but also take into account the confidence of the estimate. A well known class of functions which is well suited for this type of problem is known as *loss functions*. They measure the loss generated by a function f for a given training example $\mathbf{x}$ with known class-label y.

Definition 2.1.1 (Loss Function)**.** Assume the triplet $(\mathbf{x}, y, \operatorname{sgn}(f(\mathbf{x}))) \in \mathcal{H} \times \{\pm 1\} \times \{\pm 1\}$ being a training example, a class label and a prediction, respectively. Then a map $\ell : \mathcal{H} \times \{\pm 1\} \times \{\pm 1\} \to [0, \infty)$ with the property that $\ell(\mathbf{x}, y, y) = 0$ for all $x \in \mathcal{H}$ and $y \in \{\pm 1\}$ is called a loss-function.

The most intuitive way to measure the loss is to simply count the fraction of misclassified examples. This is achieved by the binary or 0–1 loss:

$$\ell(\mathbf{x}, y, f(\mathbf{x})) = \begin{cases} 0 & \text{if } y = \operatorname{sgn}(f(\mathbf{x})) \\ 1 & \text{otherwise.} \end{cases} \tag{2.3}$$

However, one might also want to involve the confidence with which the classification was carried out. That leads to the *hinge loss* (Bennett and Mangasarian, 1992):

$$\ell(\mathbf{x}, y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x})) = \begin{cases} 0 & \text{if } yf(\mathbf{x}) \geq 1 \\ 1 - yf(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{2.4}$$

The hinge loss already takes into account the belief in the prediction. However, Cristianini and Shawe-Taylor (2000) have shown that the squared version of (2.4) can be minimized more easily:

$$\ell(\mathbf{x}, y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))^2. \tag{2.5}$$

Examples of these loss functions are given in figure 2.3. The 0–1 loss function only punishes erroneous predictions. In contrast, the hinge and the squared hinge loss incur no penalty as long as the example is classified correctly with high belief but increase the penalty slowly when the belief decreases.

Having introduced the squared loss function, the final optimization problem can be defined:

$$\underset{\mathbf{w},b}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \ell(\mathbf{x}_i, y, f(\mathbf{x}_i)) \tag{2.6}$$

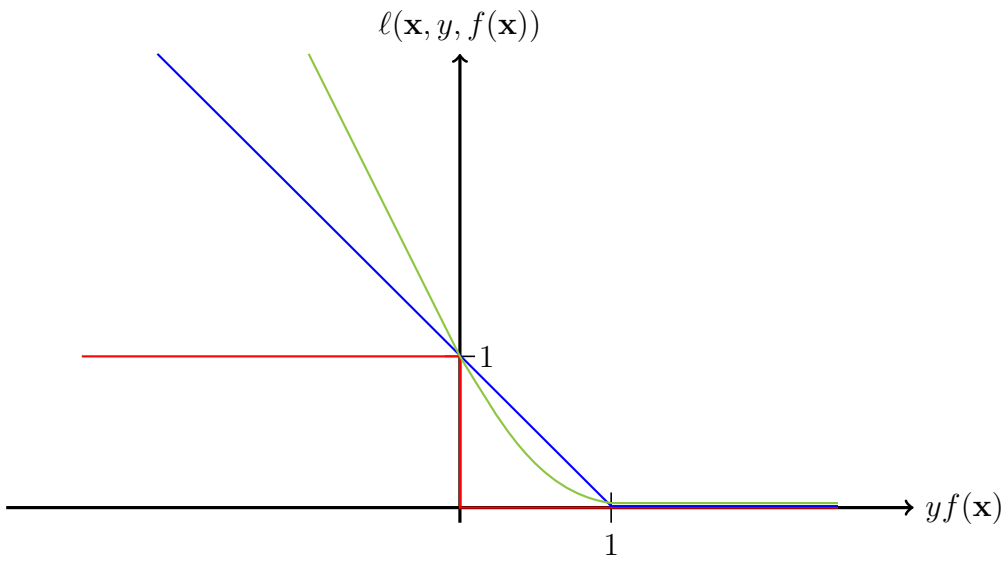

Figure 2.3: Example of three different loss functions (cf. Schölkopf et al. 2004). Blue: Hinge Loss; Red: 0–1 Loss; Green: squared Hinge Loss.

Equation (2.6) maximizes the margin (by minimizing $\|\mathbf{w}\|$) and minimizes the hinge loss. Hence, $C > 0$ is a tuning parameter that controls the tradeoff between loss induced by ℓ and size of the margin. Thus, a large value of C leads to a smaller margin but increases the number of correctly classified examples with high belief. For $C \to \infty$ the hard-margin SVM, that allows no errors, is achieved.

However, there are several advantages to not directly minimize (2.6) since neither (2.4) nor (2.5) is differentiable (Chapelle, 2007). Therefore, one usually reformulates (2.4) and introduces n so-called slack variables $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$ that measure the degree of misclassification. This leads to the primal optimization problem:

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.7}$$

subject to

$$\xi_i \geq \ell(\mathbf{x}_i, y, f(\mathbf{x}_i)), \quad \forall i = 1, \ldots, n.$$

To see that (2.6) and (2.7) are equivalent one needs to understand that the minimum of (2.7) with respect to ξ_i is reached when ξ_i takes its minimal value, which is $\ell(\mathbf{x}_i, y, f(\mathbf{x}_i))$. The next step is to split the squared hinge loss function (2.5) into two constraints, namely $\xi_i \geq 0$ and $\xi_i \geq 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)$. Thus (2.7) becomes:

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.8}$$

subject to

$$\xi_i - 1 + y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 0 \text{ and } \xi_i \geq 0 \quad \forall i = 1, \ldots, n. \tag{2.9}$$

Equation (2.8) is called an *objective function* and (2.9) are called *inequality constraints*. The combination of both (2.8) and (2.9) is known as a *constrained optimization problem* and is subject to convex optimization theory. One convenient way to solve such constrained optimization problems is to introduce *Lagrange multipliers*. For each of the constraints (2.9) a positive Lagrange multiplier, sometimes also referred to as dual variable, has to be introduced. Hence, for each training example, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \geq 0$ represents the constraint $\xi_i - 1 + y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 0$, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n) \geq 0$ denotes $\xi_i \geq 0$. Note, that the rule is that for constraints of the form $c_i \geq 0$, the constraint equations are multiplied with positive Lagrange multipliers and subtracted from the objective function. Therefore, the Lagrangian has to be formulated

as:

$$\begin{aligned} L_P(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ - \sum_{i=1}^{n}\alpha_i[\xi_i - 1 + y_i(\mathbf{w}^T\mathbf{x}_i + b)] \\ - \sum_{i=1}^{n}\beta_i\xi_i \end{aligned} \tag{2.10}$$

L_P has to be minimized w.r.t the *primal* variables $(\mathbf{w}, b, \boldsymbol{\xi})$ and maximized with respect to the *dual* variables $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq 0$. Therefore, for fixed $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, L_P is minimized as a function of $(\mathbf{w}, b, \boldsymbol{\xi})$ by setting the respective partial derivatives to zero:

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0 \tag{2.11}$$

$$\frac{\partial L_P}{\partial b} = \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{2.12}$$

$$\frac{\partial L_P}{\partial \boldsymbol{\xi}} = C - \alpha_i - \beta_i = 0 \qquad \forall i \tag{2.13}$$

together with positivity constraints $\xi_i \geq 0$, $\alpha_i \geq 0$ and $\beta_i \geq 0$. Substituting (2.11) – (2.13) into (2.10) recovers Wolfe's dual (Wolfe, 1961):

$$L_D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \tag{2.14}$$

which has to be maximized subject to $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq 0$ under the constraints (2.12) and (2.13). Compared to (2.10) equation (2.14) is a simpler convex quadric optimization problem an can be solved with standard algorithms like Gill et al. (1981). However, $\boldsymbol{\beta}$ does not occur in L_D. Thus, it can be maximized as a function of $\boldsymbol{\alpha}$. However, it must be ensured that for some $\boldsymbol{\beta} \geq 0$ the constraint (2.13) is met. This is the case if and only if $\alpha_i \leq C$ for all $i = 1, \ldots, n$, since only then a $\beta_i \geq 0$ can be found such that $C = \alpha_i + \beta_i$. This leads to the

following constraints on the dual optimization problem (2.14):

$$0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \forall i = 1, \ldots, n. \tag{2.15}$$

Since this is a convex optimization problem the Karush-Kuhn-Tucker (KKT, Kuhn and Tucker 1951) conditions apply. In addition to (2.11) – (2.13) the KKT are:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \geq 0 \tag{2.16}$$

$$\alpha_i[\xi_i - 1 + y_i(\mathbf{w}^T \mathbf{x}_i + b)] = 0 \tag{2.17}$$

$$\beta_i \xi_i = 0 \tag{2.18}$$

$$\forall i = 1, \ldots, n.$$

In combination equations (2.11) – (2.18) uniquely define the solution to the SVM optimization problem.

After having found the $\widehat{\boldsymbol{\alpha}}$ that maximizes (2.14) , $\widehat{\boldsymbol{\beta}}$ can be calculated as:

$$\widehat{\beta}_i = C - \widehat{\alpha}_i \quad \forall i = 1, \ldots, n. \tag{2.19}$$

After rearranging equation (2.11) the solution for the weight vector $\widehat{\mathbf{w}}$ of the hyperplane is given by:

$$\widehat{\mathbf{w}} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i \mathbf{x}_i \tag{2.20}$$

It is, however, worth mentioning that $\widehat{\mathbf{w}}$ is a linear combination of solely the *support vectors*, that is, only those points with Lagrange multiplier $\widehat{\alpha}_i \neq 0$. Hence, the hyperplane found by the SVM does not change when non-support vectors are removed from the training set. Moreover, it is important that all support vectors with a slack variable $\xi_i = 0$ lie on the margin (*in-bound support vectors*) and due to (2.13) and (2.18) are defined by $0 < \widehat{\alpha}_i < C$. All others ($\xi_i > 0$, *bound support vectors*) have $\widehat{\alpha}_i = C$. Equation (2.17) shows that any of the in-bound support vectors can be used to calculate $\hat{b}$.

To predict the class membership of a new sample $\mathbf{x} \in \mathcal{H}$ the linear function

$$\widehat{f}(\mathbf{x}) = \widehat{\mathbf{w}}^T\mathbf{x} + \hat{b} = \sum_{i=1}^{n} y_i \widehat{\alpha}_i \mathbf{x}_i^T \mathbf{x} + \hat{b} \tag{2.21}$$

has to be formed. Subsequently, $\operatorname{sgn}\left(\widehat{f}(\mathbf{x})\right)$ can be used to predict the class of $\mathbf{x}$ as $+1$ or -1.

2.1.5 Kernels

Another advantage of Wolfe's dual (2.14) that has not been mentioned in the previous section, is that the pattern vectors occur as dot products. This makes possible the use of so-called kernels that represent a similarity measure of the input patterns in a much higher (possibly infinite) dimensional *feature space*. The map from the input– into the feature space is usually defined as:

$$\begin{aligned} \boldsymbol{\phi} : \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \mathbf{x} := \boldsymbol{\phi}(\mathbf{x}). \end{aligned} \tag{2.22}$$

Given this map, the kernel itself is defined as:

$$k(x, x') := \langle \mathbf{x}, \mathbf{x}' \rangle = \langle \boldsymbol{\phi}(x), \boldsymbol{\phi}(x') \rangle \tag{2.23}$$

The important point is, that the kernel (2.23) allows calculation of the dot product in the feature space $\mathcal{H}$ without having to explicitly compute the map $\boldsymbol{\phi}$. This is also known as the *kernel trick*.

Even if the data already exists in a dot product space, as assumed in previous sections, it is still possible to apply a nonlinear map $\boldsymbol{\phi}$. This might change the representation of the data into one that better fits the problem. Additionally, a change of the map $\boldsymbol{\phi}$ leads to a new similarity measure, which allows one to create a large variety of learning algorithms. However, biological data, coming from microarray experiments, is already very high dimensional. Hence, this data is most often linear separable and there is usually no need

to map the features into a space with even more dimensions.

The interested reader is referred to Vapnik (1995); Burges (1998); Schölkopf and Smola (2001); Schölkopf et al. (2004) for more detailed introductions.

2.2 Feature selection using support vector machines

2.2.1 Introduction

In machine learning applications, data usually consists of measurements of some quantity. Frequently, the measurements are referred to as features or variables. Each data point is represented as a vector of dimensionality n, where n is the number of features. For each feature vector there exists a class label, defining to which class the vector belongs to. For simplicity, only two-class problems will be considered here. The challenge now is to select a classifier which assigns correct class labels to the training patterns. Furthermore, it should also be able to predict the class membership of future examples with low error rate.

It is, however, unclear if the learning algorithm needs all features to unravel the dependency between data points and class labels. A large amount of uninformative measurements might indeed mask the relationship between informative features and class labels. Additionally, the performance of a learning algorithm is strongly dependent on the quality of the data and thus, noisy, redundant or unreliable measurements impair the learning process.

There are, at least, two types of preprocessing methods that can be used to improve machine learning techniques: *feature construction* and *feature selection* methods. Feature construction methods use existing measurements and combine those to reduce the dimensionality of the problem. A well known linear example of such a method is the *principal component analysis* (PCA, Pearson 1901). There also exist non-linear methods which are based

on kernels (Schölkopf et al., 1998).

Feature selection methods, on the other hand, try to select the best subset of features to solve the classification task. Feature selection is performed in order to eliminate uninformative variables which should, in turn, lead to a better generalization performance, that is a better classification performance on previously unseen patterns. In addition to this, a reduced set of features may also give a better insight into the underlying model to be learned and a computational speedup. Other benefits might be cost reduction, in biological applications for example where only a smaller subset of genes has to be measured to detect a particular disease with the same accuracy as before (Rakotomamonjy, 2003). The goal of cost reduction cannot be reached with feature construction methods. The remainder of the section will focus on the task of feature selection.

2.2.2 Heuristics for feature selection

The task of identifying the optimal feature subset can be viewed as a search problem. Each state of this search consists of one possible feature subset. Due to the large number of features, this search space is usually high dimensional. Thus, it is obvious that this task can only be accomplished by using *heuristics*, since there are 2^n possible subsets for n features. Nevertheless, the nature of this heuristic needs to be defined by the following four characteristics:

First, the direction of the search has to be determined. One can either start with an empty model and iteratively take new features into the model, this process is known as *forward selection*. The reverse of this process, called *backward elimination*, starts with the complete model and discards one variable after the other (Neter et al., 1990).

Since it is known that an exhaustive search through the whole space is impractical the second issue is to organize the search. One possibility to do this is by using *greedy methods* that traverse the space. One possible approach is known as *stepwise* selection or elimination, and consider both

adding and discarding variables at each step of the search. By doing so, it is possible to undo previous decisions without explicitly keeping track of the search path. In the end, all states generated can be considered and the one with best performance can be selected. Alternative methods, that are not greedy but still tractable, are best-first search or beam search, for example.

The third point is related to the approach used to evaluate the performance of the subsets. Again, two distinct strategies can be distinguished (figure 2.4, John et al. 1994). *Filter methods* treat the features independent of the selected learning algorithm. Thus, these methods merely rely on characteristics of the training set to include certain features and discard others. One example of such methods are statistical techniques, that compute a dependency between features and class labels, like Pearson's correlation coefficient, wilcoxon– or t–statistics (Golub, 1999; Furey et al., 2000; Tusher et al., 2001; Hastie et al., 2009). *Wrapper methods*, on the contrary, select a certain amount of features and use this subset to run the learning algorithm on the training data. Afterwards the performance of each feature-subset is evaluated, which makes necessary the choice of a proper goodness-of-fit measure.

The last aspect to consider is a proper criterion to end the search through the space of feature subsets. When using filter methods this criterion might be to order features according to some relevance score and try different breakpoints. For wrapper methods the search could be continued until the accuracy starts to decrease or the search reaches the other end of the search space and select the best subset. For more details confer Langley (1994).

To this end, no assumption has been made about the underlying learning method. However, lots of progress has been made in the field of SVMs which are not equipped with an embedded feature selection (cf. section 2.1). However, several groups have developed feature selection algorithms for SVMs. For biological data, Moler et al. (2000) for instance, introduced a naive Bayes relevance (NBR) score to select informative features. Given the value of a gene and using Gaussian assumptions, the NBR score calculates a features' probability of belonging to class one or two. The larger the probability, the more distinct is the expression of that feature and the more likely it is to be

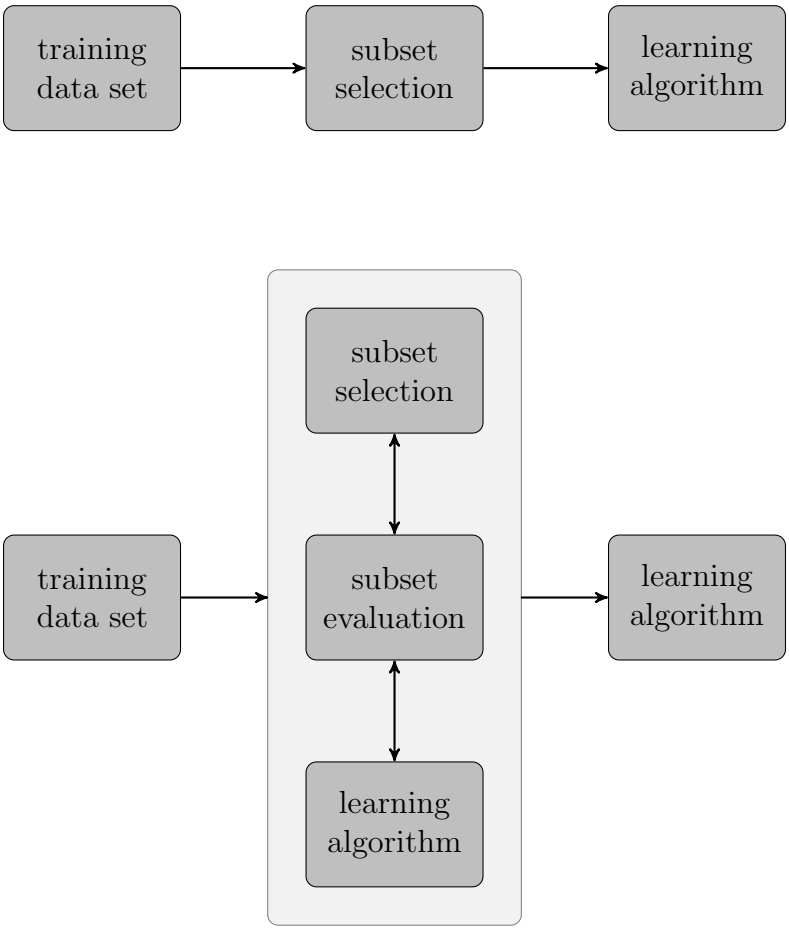

Figure 2.4: Comparison of filter- and wrapper methods for subset selection (cf. John et al. 1994). The upper panel shows a filter method; it selects the subset of features independent of the learning algorithm. The lower panel shows a wrapper method. Here, the selected subset is evaluated using the learning algorithm. It is worth noting, that this inner evaluation has to be performed on an independent test set using cross-validation, for example.

a good marker. In another work, Segal et al. (2003) used p-values obtained from students t-test to rank genes and subsequently choose a certain number of most important genes to train the model. Both methods follow the goal of selecting certain features, but none of them considers important interactions but rather treats features as independent.

2.2.3 Recursive Feature Elimination

To overcome the above-mentioned problems, Guyon et al. (2002) introduced a wrapper method called Recursive Feature Elimination (RFE). For SVMs with a linear kernel, RFE uses $\|\mathbf{w}\|^2$, the squared norm of the weight vector of the SVM hyperplane, as a ranking criterion for the importance of a feature. The authors proposed the following 4 steps (figure 2.5):

1. Train SVM on training data.
2. Rank features according to $\|\mathbf{w}\|^2$.
3. Discard the feature with smallest impact from the training data.
4. If more than one feature is left go to 1, otherwise stop.

To formally calculate the influence of the kth feature on the squared weight vector norm, equation (2.14) can be used:

$$\begin{aligned}\left|\|\mathbf{w}\|^2 - \|\mathbf{w}^{(k)}\|^2\right| &= \left|\sum_{i=1}^{n}\widehat{\alpha}_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\widehat{\alpha}_i\widehat{\alpha}_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \right. \\ &\quad \left. - \sum_{i=1}^{n}\widehat{\alpha}_i^{(k)} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\widehat{\alpha}_i^{(k)}\widehat{\alpha}_j^{(k)} y_i y_j \langle \mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}\rangle\right| \qquad (2.24) \\ &= \frac{1}{2}\left|\sum_{i=1}^{n}\sum_{j=1}^{n}\widehat{\alpha}_i\widehat{\alpha}_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \right. \\ &\quad \left. - \sum_{i=1}^{n}\sum_{j=1}^{n}\widehat{\alpha}_i^{(k)}\widehat{\alpha}_j^{(k)} y_i y_j \langle \mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}\rangle\right| \qquad (2.25)\end{aligned}$$

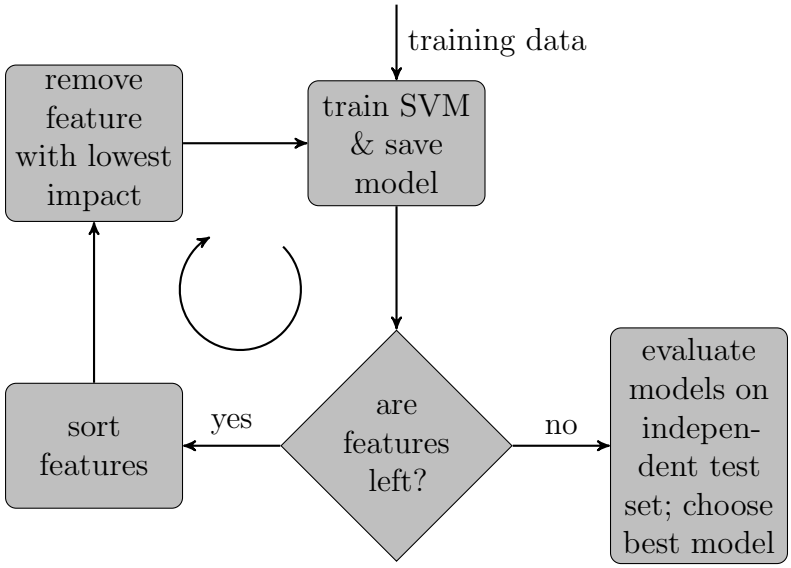

Figure 2.5: Recursive Feature Elimination workflow. This chart explains the workflow of the RFE algorithm. First, the SVM is trained on the training data set, as long as features are left these are ordered and iteratively removed. In the end the performance of all models is evaluated on an independent test set in order to find the best one.

The notation $\mathbf{c}^{(k)}$ denotes that kth feature has been removed from vector $\mathbf{c}$. Note, that the vector multiplication $\mathbf{x}_i^T\mathbf{x}_j$ in (2.14) has been exchanged by the dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. To simplify the calculation and reduce calculation time, Guyon et al. assume $\widehat{\alpha}_i^{(k)}$ to be equal to $\widehat{\alpha}_i$.

After calculating (2.25), the features can be ordered according to their importance (high value means more important). Guyon et al. recommended removing chunks of genes to speed up the procedure. Hence, as a next step a specific amount of features from the bottom of the ordered list needs to be discarded. The process of training the SVM, calculating (2.25) and removing a specific amount of potentially uninformative features is repeated until the set of surviving features is empty. In practice, all trained classifiers obtained at step 1 are saved in order to afterwards examine their performance on an independent test set and thus identify the optimal number of features. This can, for example, be done by cross-validation or by using a theoretical concept like the span estimate (cf. section 2.3).

2.3 Assessment and selection of models

2.3.1 Introduction

The performance of a classifier or any other learning algorithm on an independent test data set is known as its *generalization performance*. A detailed examination of this quantity is a prerequisite in practical applications, since it guides the choice of the learning algorithm or model. Additionally, it allows the estimation of its classification capability on yet unknown data. In the following section some details will be given in order to introduce a process called cross-validation that estimates the expected test error in section 2.3.3. The reader interested in more details is referred to Hastie et al. (2009).

2.3.2 Training- and test error

Let $\mathcal{T}$ denote a training set. The *training error* is defined as the average loss over $\mathcal{T}$:

$$\overline{err} = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i)) \qquad \forall\ \mathbf{x}_i \in \mathcal{T} \tag{2.26}$$

where ℓ is a loss function, for example the hinge loss (2.4). Usually one is more interested in the expected test error than in $\overline{err}$. However, the training error unfortunately is a bad estimate of the test error (figure 2.6), since the same data is used for fitting the model and assessing the loss it incurs. Thus, the estimate of $\overline{err}$ is biased downward. It is a too optimistic estimate of the expected generalization error. If the model complexity is increased enough the training error can become very small, or even zero. This will, however, lead to a highly overfitted model which will generalize only very poorly.

The *test-* or *generalization error* on an independent test sample is defined as:

$$\mathrm{Err}_{\mathcal{T}} = \mathrm{E}[\ell(\mathbf{x}_0, y_0, \hat{f}(\mathbf{x}_0)) \mid \mathcal{T}\ ] \qquad \mathbf{x}_0 \notin\ \mathcal{T}, \tag{2.27}$$

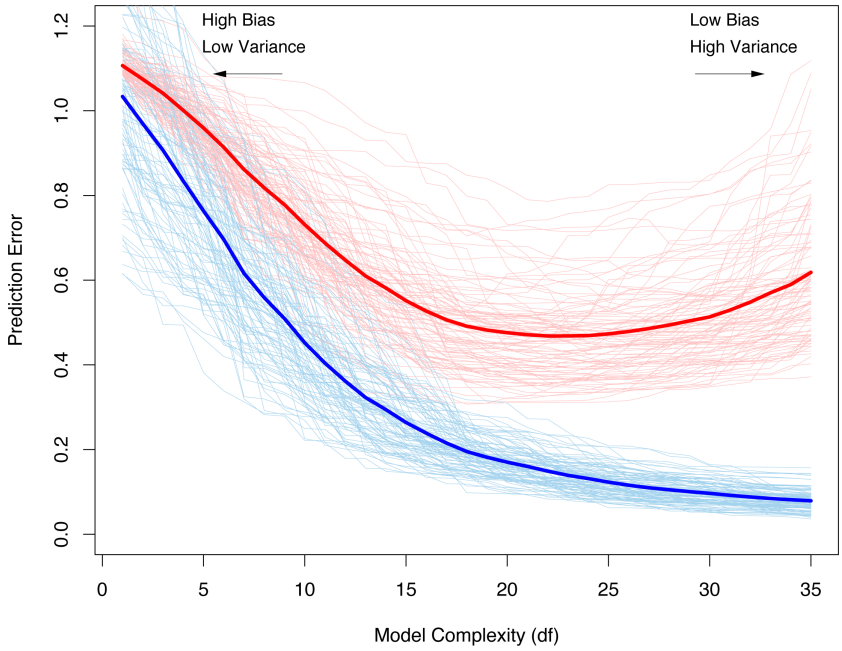

Figure 2.6: Dependency of the expected training- and test error on the model complexity, shown as solid blue and red curves, respectively. The light blue and red curves show the training- and test error for 100 training sets of size 50 each. All models have been obtained by the lasso (figure courtesy of Hastie et al. 2009).

where $\mathbf{x}_0$ and y_0 is a previously unknown test-point. Since $\hat{f}$ has been obtained by training on the fixed set $\mathcal{T}$, the estimate $\text{Err}_{\mathcal{T}}$ is only valid given this particular training set. Figure 2.6 shows the generalization error for 100 training sets as light red curves. The *expected-* or *average test error* is given by:

$$\text{Err} = \text{E}[\ell(\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i))] = E[\text{Err}_{\mathcal{T}}] \ . \tag{2.28}$$

Equation (2.28) shows that this quantity is no longer dependent on a specific training set but rather averages over all possible training sets. It is shown as solid red line in figure 2.6. In the remainder of the section the goal is to efficiently approximate the expected test error (2.28).

2.3.3 Cross-Validation

Cross-validation (CV, Mosteller and Turkey 1968; Geisser 1975; Kohavi 1995) is one possible method that efficiently re-uses the given data in order to approximate the expected test error of a previously chosen model. However, it is worth noting that there are several goals CV can be used for:

Model selection: Here, several models of the same learning algorithm are compared in terms of their estimated performance in order to choose the best one.

Model assessment: Once, the best model has been chosen, the task of model assessment is to give an estimate on its generalization error.

In this work CV was used for accomplishing the second task. Model selection was performed using a theoretical concept, which uses an analytical expression to calculate an upper-bound on the test error (cf. section 2.3.4 for more details).

In K-fold cross-validation the data is randomly split into K almost equally sized subsamples. For the kth subsample the model is fitted on the other $K-1$ parts of the data. Afterwards, the model is used to predict the class labels of the examples in the kth subsample. Thus, the estimate of the prediction error through cross-validation is given by:

$$\mathrm{CV}(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}_i, y_i, \hat{f}^{-\kappa(i)}(\mathbf{x}_i)). \tag{2.29}$$

Where $\kappa : \{1, \ldots, n\} \to \{1, \ldots, K\}$ is an indexing function that defines to which subsample of the data sample i was assigned to. Thus, $\hat{f}^{-\kappa(i)}$ denotes the fit computed on the training data after having removed the subsample i belongs to. Common choices are $K = 5$ or $K = 10$ (McLachlan et al., 2005). The condition $K = n$ leads to the *leave-one-out* (loo) cross-validation estimate, $\mathrm{CV}_{\mathrm{loo}}$, and $\kappa(i) = i$. $\mathrm{CV}_{\mathrm{loo}}$ is known to be the best approximation and an almost unbiased estimate of the expected test error (2.28) (Luntz and

Brailovsky, 1969).

However, (2.29) is still only a point estimate of the expected generalization error (2.28). To reduce the variance of the estimate it is common to repeat the K-fold cross-validation several times with different split positions. Again, common choices are 5 or 10 repeats.

2.3.4 The Span Estimate

As introduced in the last section, cross-validation is used to estimate the generalization performance of a classifier trained on some training data. However, most learning algorithms have one or more tuning parameter which need to be optimized as well. An example of such a tuning parameter is the constant C in (2.6) or the optimal number of features. The problem of finding a function with parameters that minimize the expected error on the test data is called *model selection*. However, the number of parameters determine the size of the space of possible functions. Intuitively, the model selection demands several, nested, cross-validations. The degree of nestedness of cross-validations, again, depends on the number of parameters to choose and can, thus, be a quite time-consuming method. In practice the naive strategy to exhaustively search the parameter space for the best solution becomes intractable. Thus, several authors have proposed methods to approximate an upper bound for the loo–error, $\mathrm{CV}_{\mathrm{loo}}(\hat{f})$, of a classifier (Jaakkola and Haussler, 1999; Chapelle and Vapnik, 2000a; Opper and Winther, 2000).

In this work, a quantity, called the *span* of the support vectors (Chapelle and Vapnik, 2000b) was used in order to calculate an upper bound on the number of errors made by the classifier. Let $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \ldots, \widehat{\alpha}_n)$ be the solution to the optimization problem (2.14). Chapelle and Vapnik have shown that for any support vector $\mathbf{x}_p$ the following equality is true:

$$y_p(\hat{f}(\mathbf{x}_p) - f^p(\mathbf{x}_p)) = \widehat{\alpha}_p S_p^2. \tag{2.30}$$

Here, $\hat{f}$ and f^p are the decision function (2.21) trained on the whole training

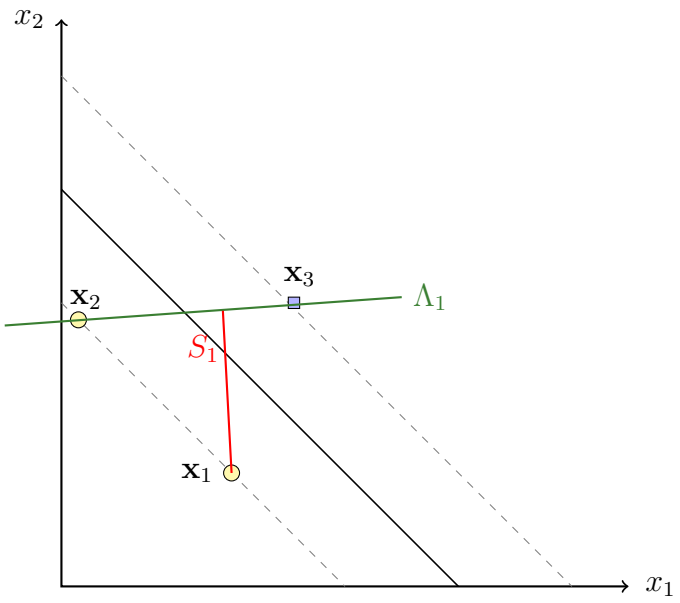

Figure 2.7: Example of the span of support vector $\mathbf{x}_1$ in $\mathbb{R}^2$. The dashed lines indicate the margin. Λ_1, shown by the green line, is the set (2.31). The red line shows the span (2.32) of support vector $\mathbf{x}_1$.

set and the set without the point $\mathbf{x}_p$, respectively. However, (2.30) is only true if in-bound and bound support vectors remain the same during the leave-one-out procedure. This limitation is obviously not always met. Nevertheless, the number of cases that violate this constraint is usually small compared to the number of support vectors. The proof of (2.30) can be found in Theorem 1 of Chapelle and Vapnik (2000a). In equation (2.30), S_p^2 is the distance of support vector $\mathbf{x}_p$ to the set of constrained linear combinations:

$$\Lambda_p = \left\{ \sum_{\{i \neq p,\ 0 < \widehat{\alpha}_i\}} \lambda_i \mathbf{x}_i, \quad \sum_{i \neq p,} \lambda_i = 1 \right\}. \tag{2.31}$$

Formally, the *span* of the support vector $\mathbf{x}_p$ is defined as:

$$S_p^2 = d^2(\mathbf{x}_p, \Lambda_p) = \min_{\mathbf{x} \in \Lambda_p} (\mathbf{x}_p - \mathbf{x})^2, \tag{2.32}$$

which is the minimum distance from $\mathbf{x}_p$ to Λ_p. A toy example is given in figure 2.7. By using the span estimate the numbers of errors made by the loo

cross-validation can be calculated as:

$$T = \frac{1}{n} \sum_{p=1}^{n} \Psi(\widehat{\alpha}_p S_p^2 - 1) \tag{2.33}$$

where Ψ is the Heaviside step function:

$$\Psi(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise .} \end{cases} \tag{2.34}$$

2.4 Receiver Operator Characteristic

Receiver Operator Characteristic (ROC) graphs have a long tradition in machine learning applications (Spackman, 1989). They are mostly used for comparison of algorithms. Initially, however, they were used in signal detection theory (Egan, 1975). Before introducing more details on the ROC space, some commonly used metrics for evaluation of classifier performance will be reviewed.

As said before, a classifier is a function which tries to map a vector $\mathbf{x}$ to a set of class labels, $\{\pm 1\}$ for example. There are models that do this in a discrete fashion, that is, produce output like -1 or $+1$. But there also exist classifiers that generate a continuous output, that is one needs to apply a cutoff in order to assign an instance to one or the other group. Assuming two classes, a classifier can create the following assignments:

***true positive* (TP):** classify a positive instance as positive.

***false negative* (FN):** classify a positive instance as negative.

***true negative* (TN):** classify a negative instance as negative.

***false positive* (FP):** classify a negative instance as positive.

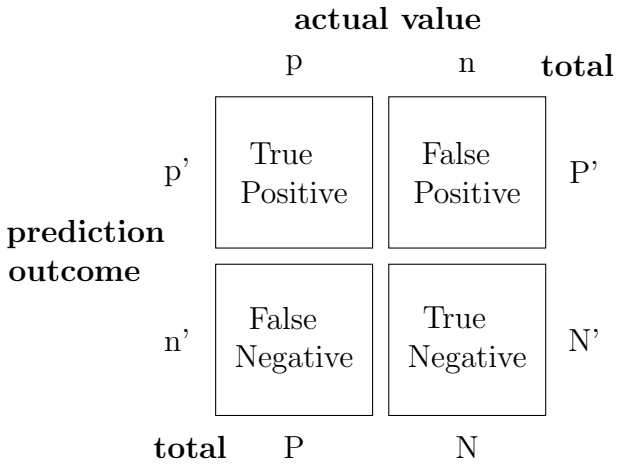

Figure 2.8: A 2 by 2 confusion table (cf. Fawcett 2004).

These class assignments can be written into a contingency table (figure 2.8). Using this information a variety of quality measures can be computed, some of which are shown in table 2.1.

The ROC space is spanned by $1 - \text{specificity}$ (FPR) on the x-axis versus the sensitivity (TPR) on the y-axis. An example of a ROC graph is given in figure 2.9. The discrete classifier, mentioned above, leads to a single contingency table and thus to exactly one point in the ROC space. Whereas the continuous method allows one to vary the cutoff for inducing the (binary) classification rule from $+\infty$ to $-\infty$ and thereby to traverse the ROC space from the lower left to the upper right corner.

The procedure of a typical ROC analysis with such a varying cutoff will be outlined by using the data from table 2.2 (see also Fawcett 2004). Figure 2.10 shows the resulting ROC graph. Table 2.2 carries information on 18 instances split into two classes (third column). The second column shows the continuous score used by a classifier to predict the class membership of each instance. This score could, for example, be the result of a SVM with *hinge loss* function (2.4). In the ROC graph a cutoff of $+\infty$ corresponds to the

Name of measure	Equation
sensitivity or true positive rate (TPR)	$\mathrm{TPR} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})}$
specificity or true negative rate (TNR)	$\mathrm{TNR} = \frac{\mathrm{TN}}{(\mathrm{FP}+\mathrm{TN})} = 1 - \mathrm{FPR}$
false positive rate (FPR)	$\mathrm{FPR} = \frac{\mathrm{FP}}{(\mathrm{FP}+\mathrm{TN})}$
false negative rate (FNR)	$\mathrm{FNR} = \frac{\mathrm{FN}}{(\mathrm{FN}+\mathrm{TP})}$
accuracy (ACC)	$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{(\mathrm{FP}+\mathrm{TN})+(\mathrm{TP}+\mathrm{FN})}$

Table 2.1: Quality measures for evaluation of classifier performance (cf. Fawcett 2004)

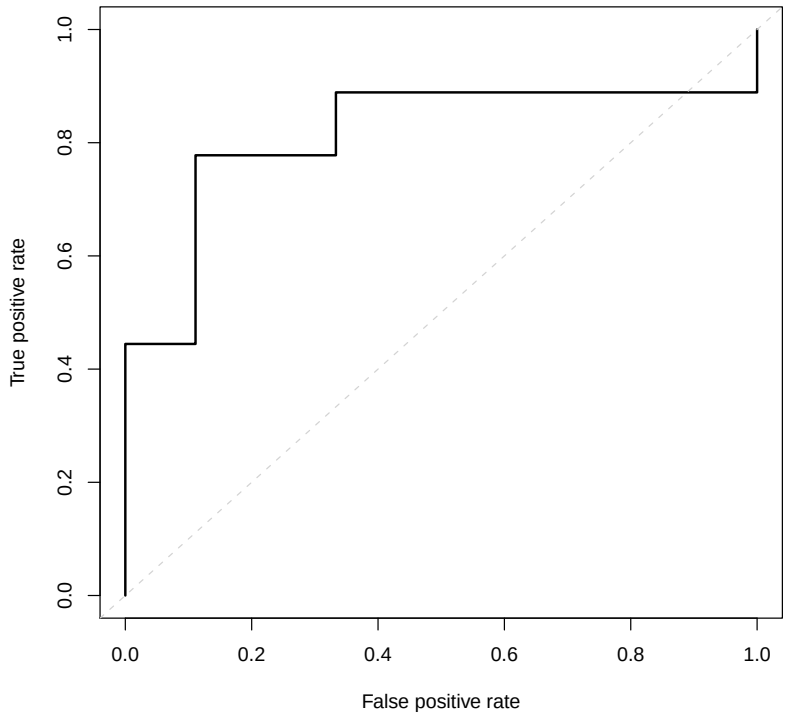

Figure 2.9: Example of a ROC graph. The x-axis denotes the false positive rate, whereas the y-axis shows the true positive rate. The dashed diagonal $x = y$ indicates the strategy of random guessing.

#	score assigned by classifier	real class label
1	0.51	+
2	0.45	+
3	0.44	+
4	0.43	+
5	0.40	-
6	0.35	+
7	0.34	+
8	0.34	+
9	0.31	-
10	0.29	-
11	0.28	+
12	0.25	-
13	0.25	-
14	0.22	-
15	0.21	-
16	0.21	-
17	0.20	-
18	0.17	+

Table 2.2: Example data for a ROC analysis. The table shows 18 instances, 9 in class + and 9 in class −. The score comes from a learning methods that uses a continuous score to predict the class membership of each instance.

point $(0, 0)$ in the lower left corner. At this point no instance is assigned to the positive class, therefore both, the number of true and the number of false positives is equal to zero. In the next step of the ROC analysis the threshold is lowered to 0.51 and thus produces 1 true positive and no false positive (table 2.3). The first false positive occurs at a threshold of 0.40 (row 6 in table 2.3). But since there are already 4 true positives the true positive rate is 0.44 and the false positive rate is 0.11. In subsequent steps the algorithm further reduces the cutoff until reaching the top right corner of the space. All intermediate results are summarized in table 2.3. The threshold belonging to the point in the top right corner of the ROC space corresponds to a classifier that assigns all instances to the positive class.

#	cutoff	TP	FP	fpr	tpr
1	Inf	0.00	0.00	0.00	0.00
2	0.51	1.00	0.00	0.00	0.11
3	0.45	2.00	0.00	0.00	0.22
4	0.44	3.00	0.00	0.00	0.33
5	0.43	4.00	0.00	0.00	0.44
6	0.40	4.00	1.00	0.11	0.44
7	0.35	5.00	1.00	0.11	0.56
8	0.34	7.00	1.00	0.11	0.78
9	0.31	7.00	2.00	0.22	0.78
10	0.29	7.00	3.00	0.33	0.78
11	0.28	8.00	3.00	0.33	0.89
12	0.25	8.00	5.00	0.56	0.89
13	0.22	8.00	6.00	0.67	0.89
14	0.21	8.00	8.00	0.89	0.89
15	0.20	8.00	9.00	1.00	0.89
16	0.17	9.00	9.00	1.00	1.00

Table 2.3: Result of the ROC analysis. The cutoffs found by the ROC algorithm are shown in column 2. The third and fourth column show the number of correspond true and false positives. The last two column indicate the respective rates that belong the counts of columns three and four.

So far, nothing has been said about random performing classifiers, i.e. classifiers that randomly assign labels to instances. In ROC space the diagonal is 'reserved' for predictors with such a performance. If, for example, a classifier would assign positive class labels to 50% of the cases it will probably reach a TPR of 0.5, likewise it will classify 50% of the negative instances as positive and thereby reach a FPR of 0.5 as well. This result corresponds to the point $(0.5, 0.5)$ in ROC space. The same applies to a predictor that guesses the positive class in 99% of the time. It will, most likely, correctly predict 99% of the positive cases, but its FPR will also increase to 0.99. Thus, any classifier that randomly predicts class memberships will produce a point on the diagonal of the ROC space. Also, it is possible that a estimator produces a ROC curve that is below the diagonal. Intuitively, this corresponds to a classifier that reliably predicts the instance to be a member of the wrong

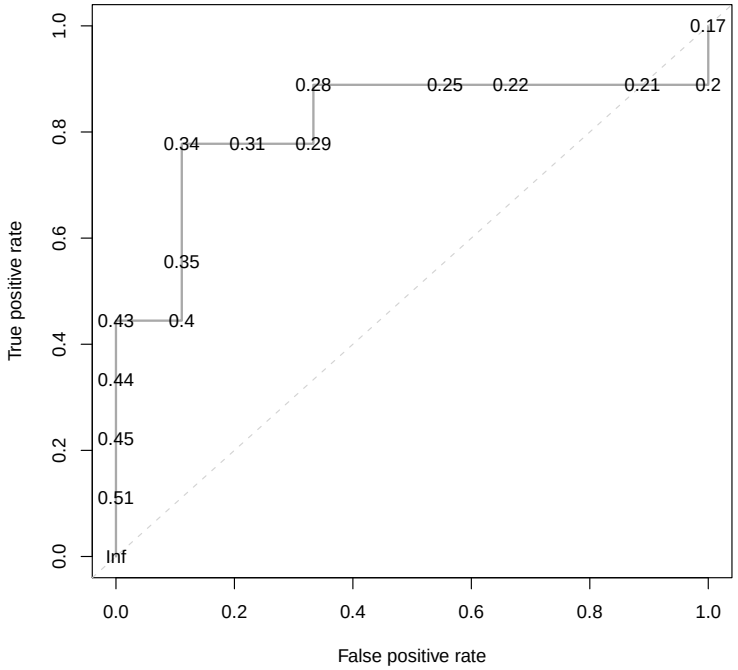

Figure 2.10: Example of a ROC graph with corresponding thresholds. The numbers indicate the threshold applied at the corresponding position of the ROC curve.

class. That is, it predicts positive instances as negative and the other way round. However, if this happens one can simply reverse all decisions made by the classifier and thus generate a mirrored ROC curve which is above of the diagonal (Flach and Wu, 2005).

2.4.1 Area under the ROC curve

For a visual comparison of different classifiers the ROC curve is a very valuable tool. Sometimes, however, it is more convenient to have a single quantity describing the classification performance. For this purpose one can calculate the area under the ROC curve (AUC, Hanley and McNeil 1982; Bradley 1997). Since the AUC is derived from the ROC space, which is a unit square, it is

always between 0 and 1. However, as said above, any classifier that lies below the diagonal in ROC space is worse than random and would have an AUC < 0.5. But still this model can be turned into a valuable classifier by simply reversing its predictions. Hence, the AUC is usually given between 1 and 0.5.

A detailed explanation of the statistical properties of the AUC is beyond the scope of this work. The interested reader is referred to the review by Fawcett (2004). The author also gives more details on averaging ROC curves in order to obtain estimations on variances to compare several classifiers. Furthermore, the review includes pseudocode to calculate ROC curves as well as AUCs. Throughout this work the software package ROCR (Sing et al., 2005) has been used for this purpose.

2.5 Gene ranking

The identification of genes that are related to a certain disease is one of the challenges in recent clinical research. The community has aggregated a huge amount of sources of knowledge which can help to accomplish this task. Examples of this gene-related knowledge are sequence information, splice variants, expression measurements, functional annotation, interacting proteins and, of course, lots of literature. A promising way is the development of bioinformatic algorithms using this data in order to rank genes which most probably play a role in a certain disease. The top ranking genes can then be followed up in subsequent biological analyses. In recent years lots of methods have been developed to rank genes based on their differential expression, see Smyth (2005) for one example. Alternatives are classification based approaches like Guyon et al. (2002).

The problem of ranking also occurs in other areas like computer sciences, where the community develops methods to rank websites. Here, a famous example is the PageRank algorithm (Page et al., 1999) used by Google (Brin and Page, 1998). When looking at the idea behind PageRank it is intuitive to transfer this algorithm to biological data and rank genes instead of web pages.

The following subsection will briefly introduce the PageRank algorithm. After that, subsection 2.5.2 will show how to modify PageRank in order to use it for biological data.

2.5.1 PageRank

Given a query, submitted by a user, PageRank has to rate the importance of each webpage on the web. This quantitative measure allows the algorithm to return the most important websites first. There exist lots of articles dealing with the theory behind PageRank (Bianchini et al., 2005; Langville and Meyer, 2004). Thus, only a brief introduction will be considered here.

In general, PageRank can be seen as a Markov model that represents the web as a matrix $\mathbf{P} = p_{ij}$ of transition probabilities. Every entry p_{ij} corresponds to the probability that the user jumps from page i to page j. The matrix $\mathbf{P}$ is defined as:

$$\mathbf{P} = \mathbf{G}\mathbf{D}^{-1}, \tag{2.35}$$

where $\mathbf{G} \in \{0,1\}^{p \times p}$ is the adjacency matrix of the underlying network. Hence, $g_{ij} = 1$ if nodes i and j are connected and $g_{ij} = 0$ otherwise. $\mathbf{D}$ is a diagonal matrix with entries

$$d_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ \deg_i & \text{otherwise} \end{cases}$$

where $\deg_i := \sum_{j=1}^{p} g_{ij}$ is the (out) degree of node i in the network.

However, building the matrix $\mathbf{P}$ from the structure of the Internet leads to some problems. First, $\mathbf{P}$ will not be stochastic, i.e. it will contain rows of all zeros ($\mathbf{0}^T$). These rows correspond to nodes in the network that have no outgoing edges, also known as *dangling nodes*. Second, the matrix is not irreducible, i.e. it is not possible to get from any node to any other node. Therefore, Brin and Page forced the matrix $\mathbf{P}$ to fulfill these criteria by first

requiring stochasticity[1]:

$$\mathbf{P}^* = \mathbf{P} + \mathbf{a}\mathbf{v}^T. \tag{2.36}$$

Where all elements $a_i = 1$ for dangling nodes in $\mathbf{P}$ and $a_i = 0$, otherwise. $\mathbf{v}^T$ is a probability vector, i.e. $\mathbf{v}^T\mathbf{e} = 1$. They further adjusted the matrix in such a way as

$$\widehat{\mathbf{P}} = d\mathbf{P}^* + (1-d)\mathbf{e}\mathbf{v}^T, \tag{2.37}$$

which ensures primitivity. Here, $\mathbf{e}$ is a vector of all ones. $d \in [0,1]$ is a fixed parameter which is called the "damping factor". Google seems to use a damping factor $d = 0.85$ (Langville and Meyer, 2004). The matrix $\widehat{\mathbf{P}}$ fulfills the criteria of being both stochastic and irreducible. Furthermore, given by the Perron–Frobenius theorem (Perron, 1907), $\widehat{\mathbf{P}}$ is primitive. This property is important, since it implies that the *power method* (Golub and van der Vorst, 2000) converges to the solution of the problem.

Using the matrix $\widehat{\mathbf{P}}$, the PageRank vector $\mathbf{r}^T$ can be found by solving the following eigenvector problem:

$$\mathbf{r}^T\widehat{\mathbf{P}} = \mathbf{r}^T. \tag{2.38}$$

However, (2.38) is subject to an additional *normalization equation*, $\mathbf{r}^T\mathbf{e} = 1$, which ensures $\mathbf{r}^T$ being a probability vector.

As already mentioned above, the solution to (2.38) is usually found by numerical methods like the power method. The power method is slow but has the advantage that it can make use of vector-matrix multiplications on the sparse matrix $\mathbf{P}$ and does not need the completely dense matrix $\widehat{\mathbf{P}}$ to be formed:

$$\mathbf{r}^{(k)T} = \mathbf{r}^{(k-1)T}\widehat{\mathbf{P}} \tag{2.39}$$

$$= d\mathbf{r}^{(k-1)T}\mathbf{P} + (d\mathbf{r}^{(k-1)T}\mathbf{a} + (1-d))\mathbf{v}^T. \tag{2.40}$$

[1] square matrix with each row consisting of nonnegative real numbers that sum up to 1

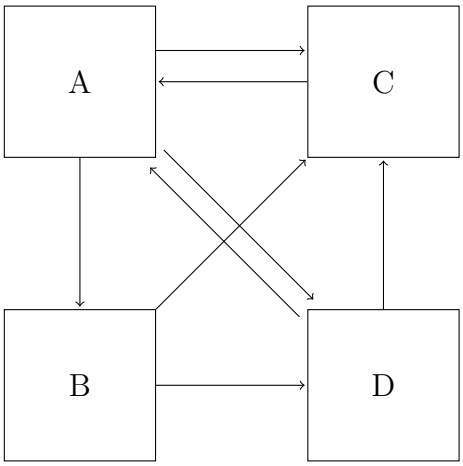

Figure 2.11: Toy network of four webpages. The arrows should illustrate links between webpages. The direction of the arrow indicates the direction of the link, i.e. which website contains a link to what other website.

Thanks to the forced irreducibility of $\widehat{\mathbf{P}}$ existence of $\mathbf{r}^T$ is guaranteed. Furthermore, the stochasticity of $\widehat{\mathbf{P}}$ leads to a spectral radius of $\rho(\widehat{\mathbf{P}}) = 1$, which ensures uniqueness of the solution (Meyer, 2001). If $\widehat{\mathbf{P}}$ would not be primitive, there might be more than one eigenvalues on the unit circle which would cause problems for the power method.

It has recently been shown (Bianchini et al., 2005), that the PageRank problem (2.38) can also be written as a linear system:

$$(\mathbf{I} - d\mathbf{P}^*)\mathbf{r} = (1-d)\mathbf{v}^T. \tag{2.41}$$

This formulation of the problem gives researchers the chance to use approaches different from Markov chain methods or the power method. Morrison et al. (2005), for example, propose to use the Jacobi method (Golub and Van Loan, 1996).

Given the toy network in figure 2.11, the adjacency matrix $\mathbf{G}$ would be

defined as:

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

Similarly, $\mathbf{D}^{-1}$ would look like:

$$\mathbf{D}^{-1} = \begin{bmatrix} 1/3 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \end{bmatrix}.$$

Solving (2.41) using the matrices above and a damping factor $d = 0.85$, leads to highest PageRank for node A, followed by C, D and B on the last place. This ranking can be understood when thinking of a confidence voting principle (Morrison et al., 2005), that is, PageRank regards a link from page i to page j as a "vote of confidence" for page j. Node C is collecting confidence from all other nodes in the network and then casts everything over to node A. Thus, with the weight of nodes B and C, A becomes the most important one.

The intention behind the damping factor can be understood when thinking on a user surfing the web. In that case, a damping factor of $d = 0.85$ corresponds to a surfer who uses in 85% of the cases a link on the webpage he or she is currently visiting (left hand side of (2.41)). And in 15% of the time the surfer selects the address line of the browser and enters the address of a page to "teleport" to (right hand side of (2.41)). Therefore, PageRank can be personalized by adapting the probability vector $\mathbf{v}^T$ in (2.41) with a personalized one. Thus, increasing the probability of "teleporting" to a page which has a larger value in this vector.

More detailed introductions to the PageRank algorithm can be found in the publications by Bianchini et al. (2005) and Langville and Meyer (2004), for example. The authors also cover topics like sensitivity and stability of the

algorithm. Furthermore, they outline speed and storage limitations and give areas of future research.

2.5.2 GeneRank

Morrison et al. (2005) have recently published a modified version of the PageRank algorithm, called GeneRank. Instead of ranking webpages on the Internet their algorithm can be used to rank biological entities in biological networks. These biological networks can, for example, be composed of gene-gene or protein-protein interactions. Thus, a node in the network corresponds to a gene or a protein and the edges encode for an interaction between those.

However, compared to the Internet these biological networks are not necessarily directed. Hence, one could argue that in the undirected case the ranking is highly correlated to the degree of a node because the weight that is transferred at time point t from node i to node j is transferred back at time point $t + 1$. This is, however, only true for a damping factor of $d = 1$. Therefore, Morrison et al. (2005) suggest to use a damping factor of $d = 0.5$ and personalize GeneRank in such a way as to use the absolute value of expression as the weight of each node. However, GeneRank still offers the freedom to use the damping factor for adapting the influence of the network structure and the expression information on the ranking. A value of $d \to 0$ results in a ranking mostly affected by the expression, whereas $d \to 1$ corresponds to a ranking that is more dependent on the network structure.

2.6 Generation of the interaction graph

Obviously, all pathway-based classification methods need a graph structure which provides information on the interaction of entities. In a recent paper we give an overview on several databases carrying such information (Porzelius and Johannes et al., (2011)). In this work information on protein-protein interactions (PPI) were used. However, the algorithms are capable of including

any other knowledge that can be represented as a graph structure.

The Human Protein Reference Database (HPRD, Prasad et al. 2009) was used as a source on interacting proteins. The flatfile of binary protein-protein interactions (Release 8, 09/06/07) was downloaded from their servers. After mapping the proteins to the genes present on the microarray used in the experiments (cf. 2.7), an interaction matrix G of dimension 7,896 $\times$ 7,896 was created with

$$g_{ij} = g_{ji} = \begin{cases} 1 & \text{if proteins } i \text{ and } j \text{ interact,} \\ 0 & \text{otherwise.} \end{cases}$$

The mapping resulted in a matrix with 59,924 non-zero elements, which represent 29,962 interactions since the matrix is symmetric.

2.7 Data sets

Gene expression profiles of 788 breast cancer patients who did not receive any systemic therapy were downloaded from the NCBI Gene Expression Omnibus[2] (GEO) data repository.

The first data set (GEO series accession number GSE11121; Schmidt et al. 2008), coming from the Department of Obstetrics and Gynecology of the Johannes Gutenberg University Mainz, was collected during 1988 and 1998. It contains 153 lymph node-negative, relapse free patients and 47 lymph node-negative patients that had a relapse (median relapse time 6.04 years).

The second data set (GSE2034; Wang et al. 2005) was produced at the Erasmus Medical Center in Rotterdam, Netherlands. The samples were collected between 1980 to 1995 and, again, did not receive any systemic therapy. The study started with a total of 436 tumors. However, due to thorough quality control 53 sample were discarded because of insufficient

[2] www.ncbi.nlm.nih.gov/geo/

tumor content, 77 samples did not have adequate RNA quality and 20 patients were lost due to poor chip quality. Therefore, this data set comprises 286 lymph node-negative breast cancer samples, and 106 of the patients experienced a relapse (median relapse time 7.17 years).

Data coming from the third and fourth data set, is sometimes referred-to as the TRANSBIG cohort (Buyse et al., 2006). TRANSBIG is a network for translational research established by the Breast International Group (BIG). They recently conducted a validation study of a 70-gene signature (van 't Veer et al., 2002) and showed reproducible prognostic value in a collection of 302 patients from five different centers. The clinical, pathological, and gene signature data were merged at the TRANSBIG Secretariat at the Institute Jules Bordet in Brussels, Belgium. The study started with a cohort of 403 samples. However, due to insufficient quality or missing annotation 101 patients had to be discarded from the study. Samples of the TRANSBIG cohort used in this work consist of data coming from Loi et al. (2007) and Desmedt et al. (2007). The data set of Loi et al. (GSE6532) consists of 125 samples including 49 relapse events (median relapse time 7.7 years). The data set published by Desmedt et al. (GSE7390) contains 177 patients, with 85 relapse events (median relapse time 7.42 years). It is, however, worth mentioning that the samples in the Loi and Desmedt data sets were selected by Geo accession numbers (GSM) according to Schmidt et al. (2008), who also analyzed the data recently.

2.7.1 Data preprocessing

The raw data were preprocessed using robust multichip average (RMA, Irizarry et al. 2003). After combining the single data sets, quantile normalization (Bolstad et al., 2003) was performed in order to remove inter-data set effects. Mapping of the protein-protein interactions to the probe sets present on the HG-U133A microarray resulted in 13,671 features with *prior* knowledge from a total of 22,283 features present on the chip. All annotation-data concerning the HG-U133A microarray was obtained from the R-package `hgu133a.db`

(Carlson et al., 2009) in the R-Bioconductor environment (Gentleman et al., 2004).

2.7.2 Determination of the *ERBB2* status

ERBB2 is a frequently amplified oncogene in breast cancer. Determination of *ERBB2* status is important, since ERBB2-positive patients have a poor prognosis and targeted treatment strategies (i.e. monoclonal antibody against *ERBB2* – Trastuzumab) are available for ERBB2-positive breast cancer patients. The *ERBB2* status is routinely detected by immunohistochemistry in the clinics. Since the receptor status was not available for all samples, *ERBB2* status was determined using Affymetrix probe set `216836_s_at` as previously described by Rody et al. (2009); Brase et al. (2010). The classification of the ERBB2 status by microarrays has recently been shown to have a concordance of 96% when compared to immunohistochemistry data (Roepman et al., 2009). The expression values of the *ERBB2* probe set showed a bimodal distribution over the 788 samples (figure 2.12). By visual inspection 11.45 was chosen as cutoff to assign the patients into ERBB2 positive and negative. Afterwards all probe sets targeting *ERBB2* were removed from the raw data which left in 22,281 features for classification.

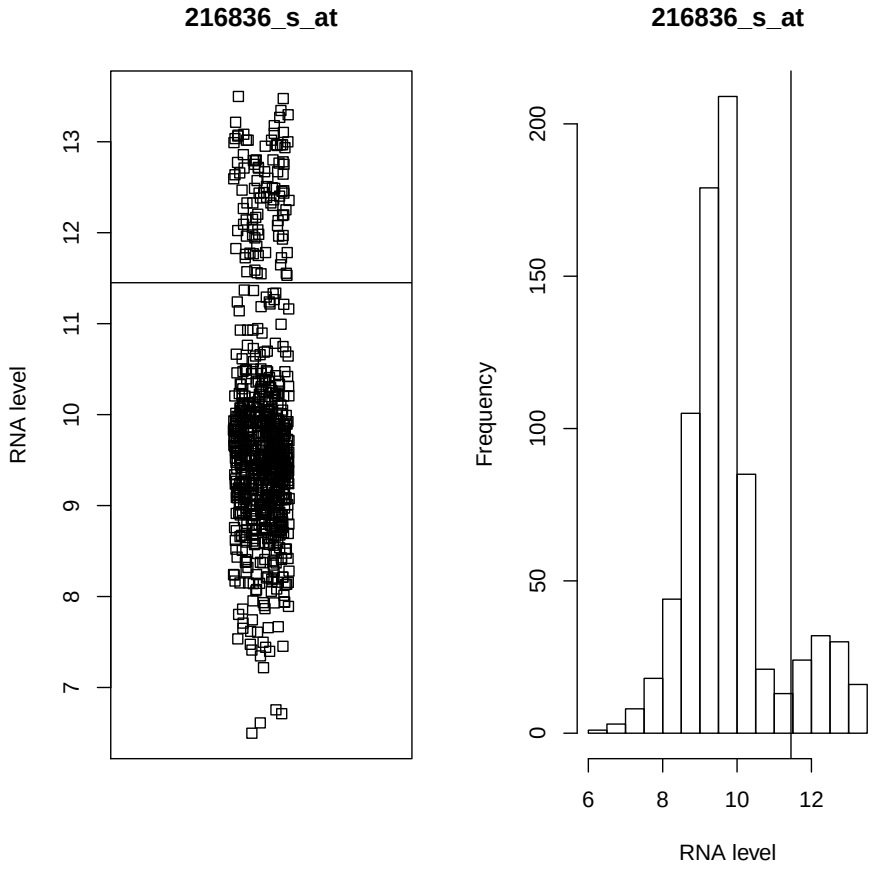

Figure 2.12: Cutoff for determining the *ERBB2* status of 788 patients. By visual inspection an RNA level of 11.45 was chosen as cutoff for assigning samples as either *ERBB2* positive or negative.

Chapter 3

Results and Discussion

3.1 Reweighted Recursive Feature Elimination

New prognostic makers of breast cancer metastasis are urgently needed in order to avoid patients from being over- or undertreated. Classification methods have shown to be a promising approach for detecting novel biomarkers. To solve the task of biomarker discovery one usually applies classification methods that are able to perform a feature selection. In risk prediction for example, these algorithms try to predict the risk group of a patient and provide information on which features were necessary for this prediction. For that, standard classification methods merely rely on one source of data like gene expression measurements, for example. A major drawback is, that these methods mostly detect genes that exhibit high fold-changes (Chuang et al., 2007), which might only be downstream effectors of the key players. It has, however, recently been shown that incorporating information on feature connectivity into the classification process can increase the classification performance (Chuang et al., 2007; Rapaport et al., 2007; Bellazzi and Zupan, 2007; Lee et al., 2008; Zhu et al., 2009; Su et al., 2009; Binder and Schumacher, 2009; Yousef et al., 2009).

Here, Reweighted Recursive Feature Elimination (RRFE, Johannes et al. 2010) was introduced. RRFE is a new algorithm, developed to accomplish the task of including *prior* knowledge into the classification process and thus identify key players in cancer development and prognosis. The *prior* knowledge needed by RRFE is a graph structure given as an adjacency matrix. This graph structure has to provide connectivity information on the features that are given in an additional expression matrix. It is, however, worth mentioning that RRFE is not limited to biological data. The only prerequisite is that the user provides a data matrix containing the same features that are present in the adjacency, which reflects the graph structure. Nevertheless, here the focus is on expression data and pathway knowledge. The expression data is represented by $P \times n$ matrix, containing measurements of P genes for n patients.

The workflow of RRFE is depicted in figure 3.1. Usually, the algorithm is run in a cross-validation (Geisser 1975; Kohavi 1995; c.f. section 2.3.3), therefore the expression data is split into a training set (90% of the samples) and a test set (10% of the samples). First, the training data is used to calculate a fold-change for each gene, i.e. the change in expression according to a specific endpoint. Subsequently, RRFE uses the graph structure and the fold-change information as input to GeneRank (Morrison et al. (2005); see section 2.5.2). GeneRank uses this information to calculate a weight for each gene that is based on the number of connected neighbours and their change in expression. This weight can subsequently be used to rank the features. GeneRank is a modified version of Google's PageRank algorithm (Brin and Page, 1998). PageRank is based on the hypothesis that a web page should be highly ranked in a search result, if other highly ranked pages contain links to it. Our motivation to use GeneRank was, that this hypothesis can also be transferred to biological networks. It is known that key players in cancer development do not necessarily exhibit high fold-changes, but are most often highly connected to other genes that change their expression level significantly (Chuang et al., 2007).

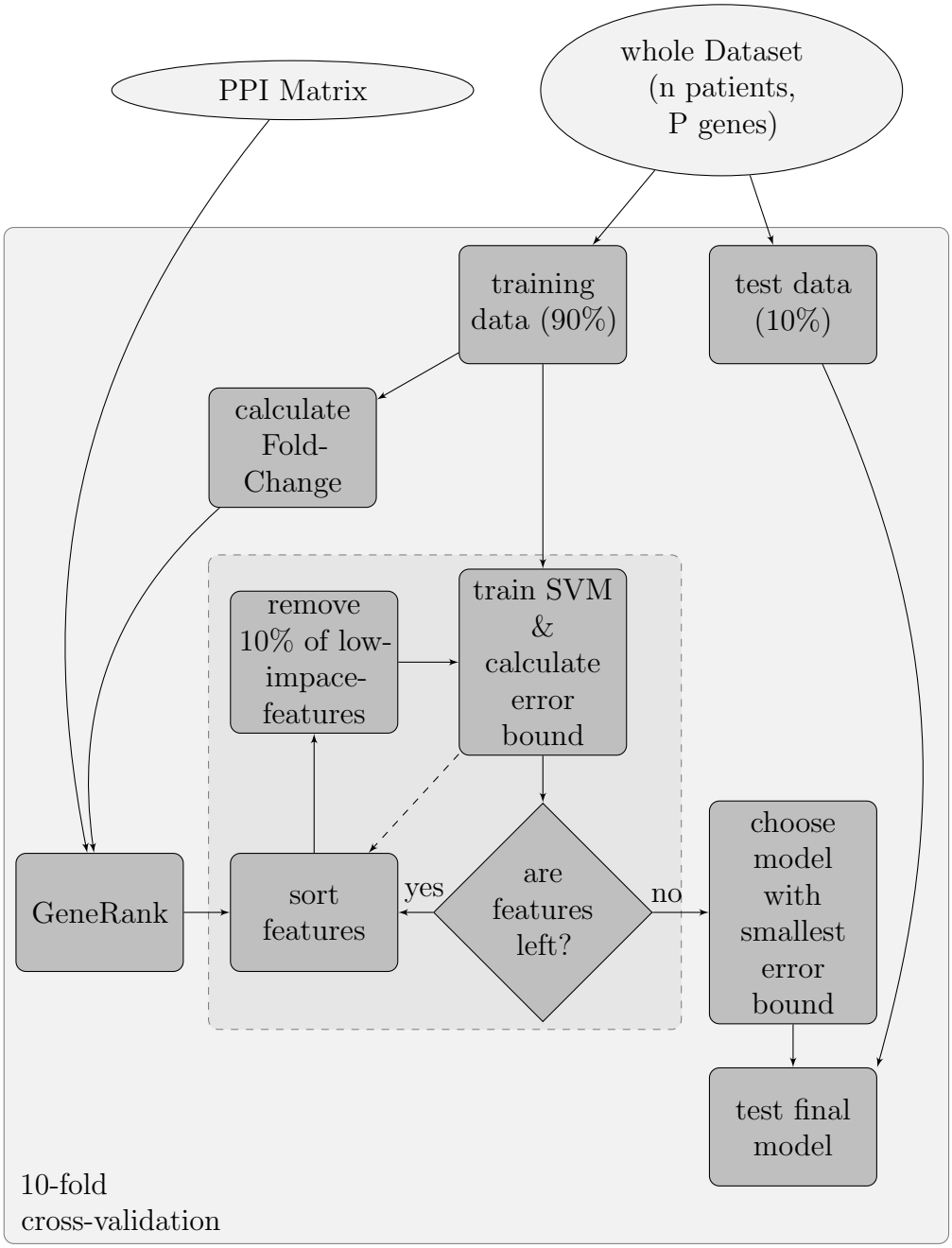

Figure 3.1: Workflow of Reweighted Recursive Feature Elimination.

However, the weight calculated by GeneRank had to be transformed in order to avoid single genes of having a weight that is much higher compared to the others. This transformation was done in a rank-based fashion. Let r_i denote the GeneRank of the ith gene, the transformed weight was calculated as:

$$r_i^* = |\{r_l | r_l \geq r_i\}|. \tag{3.1}$$

Using equation (3.1), the gene with highest GeneRank gets a weight of 1, the gene with second highest GeneRank gets weight 2, the next 3, ..., P.

Afterwards, the expression data was used to train a SVM (Boser et al. 1992; c.f. section 2.1). The generalization performance of the trained model was determined using the *span estimate* (see section 2.3). The span estimate allows to calculate an upper bound on the leave-one-out error of a SVM classifier. Subsequently, the features were ordered according to the SVM-RFE criterion (Guyon et al. 2002; section 2.2.3) coming from equation (2.25) in combination with the transformed GeneRank. This combination was performed by using an function ϕ, which was defined as

$$\phi(w_i, r_i^*) = w_i \frac{1}{r_i^*}, \tag{3.2}$$

where r_i^* is the transformed GeneRank and w_i comes from RFE.

After having ranked all genes, RRFE discards 10% of the genes from the bottom of the ordered list. This speeds up the running time of the algorithm but increases the risk of missing the perfect subset of genes. However, since the high number of features leads to a combinatorial explosion, an exhaustive enumeration of all subsets was infeasible.

Subsequently, the next SVM model was trained using only those features that were not excluded during the elimination process. Again, a bound on the leave-one-out error was calculated. This process of training the SVM, ranking the remaining features and discarding 10% of them was repeated until only one feature was left in the model (dashed square in figure 3.1). Since each SVM model was associated with a certain error, given by the span-estimate,

the model with the best performance could easily be selected. It inherently consisted of the best performing features. In the end, this model was tested on the previously defined test set.

Repeating this process ten times led to a ten-fold cross-validation. After these ten folds every sample had once been a member of the test set. Thus, the prediction of the class-label of each sample could be compared to its original one. This information was used to calculate a sensitivity and a specificity for the classifier (cf. section 2.4). Subsequently, as a measure of quality, a ROC curve was plotted and the AUC was calculated. To obtain a confidence interval for the AUC estimate the cross-validation was repeated several times with different split positions.

The combination of GeneRank and RFE increased the influence of low expressed genes on the classifier if they were highly linked to differentially expressed genes. Furthermore, this combination accounted for the fact that many functionally relevant genes were not detected by standard methods that merely rely on expression data. This, in turn, decreased the unexploited information in the data.

3.1.1 Evaluations of the method

The newly developed RRFE method was evaluated with different goals. First, the aim was to show that the genes selected by the algorithm were reproducible (see section 3.1.1.1). That is, the genes that 'survived' the feature elimination and made it into the final model should be independent of the samples that comprise the training set. Furthermore, the selected genes should be associated with the underlying disease (cf. 3.1.1.1). Second, the algorithm should improve the classification performance. That is, the AUC achieved by RRFE should be higher compared to both, standard methods (section 3.1.1.2) and methods that are capable of including *prior* knowledge (section 3.1.1.5).

All experiments were run in a five times repeated ten-fold cross-validation. This means, in each fold the model was trained using 90% of the data and

then it had to predict the class labels of the remaining 10%. Hence, after ten folds every sample had been part of the test set once and the results could be used to calculate an AUC. However, this point estimate of the AUC might not be the best. Therefore, the CV was repeated five times with different split positions to get a confidence interval for the AUC.

The graph structure needed by RRFE was produced using the HPRD (see section 2.6) and all experiments were performed using this adjacency matrix. However, to be sure that the AUC reached by RRFE is independent of the underlying graph structure, the algorithm had also been evaluated using prior knowledge coming from KEGG (Kanehisa and Goto, 2000) and the ConsensusPathDB (CPDB, Kamburov et al. 2009). Since HPRD and CPDB both contained information on PPI the nodes of the graph corresponded to proteins. In the case of KEGG the nodes of the network were genes.

It is important to mention that the density of connections among genes involved in cancer is higher compared to genes that are not so well known. Indeed, there is an annotation bias in the pathway knowledge because the parts of the network comprising disease related genes are better understood. However, we believe that with increasing amount and quality of biological data the influence of this bias will decrease. Moreover, the damping factor can be used to adjust the influence of the pathway knowledge according to its reliability (see below).

According to the analyses conducted by Schmidt et al. (2008), four gene expression data sets were downloaded from GEO (c.f. section 2.7). Evaluations on the gene list stability (subsection 3.1.1.1) were performed on the combination of all four data sets. This combination led to one large data set comprising almost 800 samples. The AUC was investigated on all single data sets as well as on the combined one. Since all experiments have been performed on the same microarray platform and non of the patients received any systemic treatment the combination of the data sets was possible. However, to avoid within-study differences from influencing the classification result, all data sets have been normalized together.

In order to calculate a fold-change for the nodes in the graph, as needed by GeneRank, the microarray data had to be mapped to the pathway data. Whenever a gene was represented by more than one probe set on the microarray, the measurements were averaged to obtain one value for the gene. When using KEGG, the averaged expression data could directly be mapped to the graph. However, in the case of HPRD and CPDB, a gene could be represented by more than one protein in the network structure. In this case, the averaged expression values were assigned to all proteins that originate from this particular gene.

Like most classification methods, RRFE has parameters that have to be adjusted carefully. As suggested by Morrison et al. (2005), the damping factor of GeneRank was set to $d = 0.5$, which led to an equal influence of the pathway knowledge and the fold-change information on the ranking of the genes. Nevertheless, an experiment was performed in order to judge the influence of the damping factor on the classification result (see below). RRFE uses a SVM with linear kernel. The soft-margin parameter in equation (2.6) was optimized using the span-estimate. However, the range was limited to $C \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. The following sections will describe the outcome of the evaluations.

3.1.1.1 Evaluation of RRFE in terms of stability and interpretability of selected features

The human epidermal growth factor receptor 2 (*ERBB2*) is a oncogene, that is frequently amplified in breast cancer patients (Slamon et al., 1987, 1989). The amplification of *ERBB2* is associated with poor prognosis and its status is routinely analyzed in the clinics, because patients with a *ERBB2* amplification benefit from a specific treatment with a monoclonal antibody (trastuzumab, Emens 2005). It is known that the elevated signalling by this protein drives cells into proliferation and protects them from apoptosis (Weinberg, 2006). Since the ERBB2 amplicon is a long stretch of chromosomal DNA it also encompasses additional genes besides *ERRB2*. Therefore, these genes are

co-amplified, which leads to an overexpression on the RNA level.

The amplification of the region around *ERBB2* on chromosome 17 and the resulting overexpression of the neighboring genes is a good example to investigate the feature selection of a classification algorithm. Therefore, this analysis aims at evaluating RRFE in terms of stability as well as the interpretability of genes selected for the classification. When discriminating ERBB2-positive from ERBB2-negative patients our assumption was that due to the high correlation with the class labels and the elevated fold-change a standard algorithm will mostly choose genes that are located in close proximity to *ERBB2*. Apparently, most of these genes need not necessarily be associated with the intrinsic biology and the adverse clinical outcome of the ERBB2 breast cancer subtype (Slamon et al., 1987; Sorlie et al., 2001). However, due to the way RRFE incorporates the pathway knowledge one would expect changes in the selected features.

Based on the expression level of the ERBB2-specific probe sets 788 patients were assigned into two groups (cf. section 2.7.2). 686 patients were defined as ERRB2-negative and 102 as ERBB2-positive. Afterwards, the ERBB2-specific probe sets were omitted from the data set. After having assigned the patients into two groups, all probe sets that could not be mapped onto the PPI network were removed, which left 13,671 features for the classification. On the basis of these features RRFE was used to predict the ERBB2 status of the patients and subsequently compared to SVM-RFE. As expected, both algorithms performed well and reached an AUC close to 1 (figure 3.2). However, as already mentioned, it is straightforward for the algorithms to achieve good results by simply choosing genes lying within the amplicon. Therefore, the classification performance was not the aim of this analysis. It is more interesting to compare the genes selected by the algorithms (table 3.1).

Table 3.1 shows for each algorithm the 10 genes that have been selected most often. Due to the cross-validation setting, a gene can be chosen 50 times at maximum. 6 out of 10 genes selected by SVM-RFE are lying within the ERBB2 amplicon (indicated by bold chromosome numbers). Most probably, RFE selects those genes because of their high fold-change as well as their

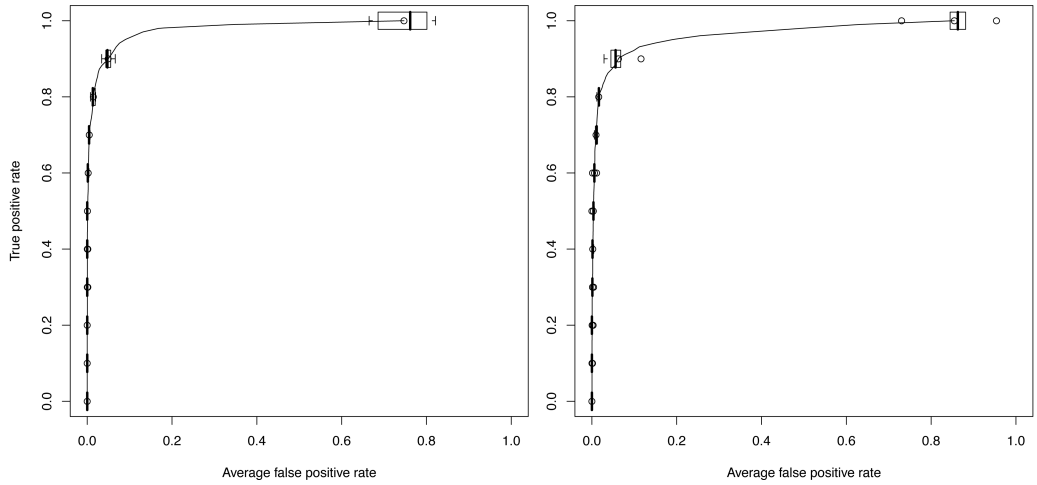

Figure 3.2: AUC for predicting the *ERBB2* status. Both algorithms reached an AUC close to 1. RFE shown on the left side obtained an AUC of 0.982. RRFE reached a slightly smaller AUC of 0.966

correlation with the receptor status. RRFE on the other hand, chooses only 3 of the genes that lie adjacent to ERBB2. This indicates that the pathway knowledge influences the choice of the genes in such a way as RRFE does not simply choose genes that are correlated with the class labels. Nevertheless, this does not prove that the genes selected by RRFE are really associated with the ERBB2-positive breast cancer subtype.

Therefore, a pathway overrepresentation analysis (Fröhlich et al., 2008) was performed using the top 100 genes selected by both algorithms. The algorithm developed by Fröhlich et al. obtains the pathway membership for each of the 13,671 features. Afterwards, it uses fisher's exact test (Fisher, 1922) to test for significantly overrepresented pathways among the genes chosen by the classifiers. This analysis revealed that, among others, the ERBB2 signaling pathway was significantly associated with the genes selected by RRFE. No (statistically significant) overrepresented pathway could be found among the genes chosen by SVM-RFE.

Furthermore, the selection of these disease-associated genes seems to be better reproducible (table 3.1). The third column shows how often a particular gene was not excluded during the feature elimination. The feature selection

A)		**SVM-RFE**		
Gene symbol	Chromosome	times chosen	log fold-change	connections
GRB7	**17**	50	-2.105	14
NDUFA7	19	31	0.308	2
MED24	**17**	29	-1.479	4
LRRC59	**17**	24	-0.729	1
CRKRS	**17**	23	-1.825	5
PHB	**17**	22	-0.512	14
CD86	3	22	-0.099	4
MED1	**17**	21	-1.627	26
ACTG1	17	21	0.024	35
NR2F1	5	20	-0.538	13
B)		**RRFE**		
Gene symbol	Chromosome	times chosen	log fold-change	connections
GRB7	**17**	50	-2.105	14
EGFR	7	49	0.032	151
WFDC2	20	48	1.044	1
EWSR1	22	48	-0.008	97
TP53	17	48	0.240	242
PRKACA	19	48	0.022	131
LRRC59	**17**	48	-0.730	1
SMAD3	15	47	0.072	166
CRKRS	**17**	47	-1.825	5
PRKCA	17	47	0.038	162

Table 3.1: Results of the *ERBB2* status prediction. Top 10 genes chosen by both methods after cross-validation, i. e. a gene would have been chosen 50 times if it was considered as important by all classifiers. (A) genes chosen by the SVM-RFE algorithm. (B) genes considered as important by RRFE. Bold means that the gene lies adjacent to the *ERBB2* gene.

of SVM-RFE can not be considered as stable: While the first gene in the list, *GRB7*, was chosen by all models, the last one was only selected by less than 50%. RRFE chooses *GRB7* 50 times as well, however the last gene, *PRKCA*, is still selected in 47 of 50 cases, that is, it was considered as important in

94% of the models.

To conclude, this analysis showed that the pathway knowledge, as used by RRFE, enables the algorithm to choose genes that are correlated to the intrinsic biology of the disease. Besides this, the features are selected with a higher reproducibility, which might decrease the doubts raised regarding the reliability of these tools in clinical applications.

3.1.1.2 Evaluation of RRFE in terms of classification accuracy

Besides a consistent and interpretable feature selection a new classification algorithm should also improve the classification performance in terms of sensitivity and specificity. Therefore, a thorough analysis of RRFE was conducted.

In a first analysis, RRFE was used to predict whether or not a breast cancer patient will suffer from a relapse, which is one of the major challenges in clinical cancer research (Ein-Dor et al., 2006). Therefore, all four gene-expression data sets were used independently as well as in a combined manner, which led to a total of 5 experiments. However, most pathway-based classification methods can only use features that are annotated in the corresponding pathway database. This is usually connected with a substantial loss in the number of features, since a huge amount of genes has not yet been assigned to a pathway (Huttenhower et al., 2009). In order to avoid this limitation, RRFE was adapted to assign the smallest weight returned by GeneRank to all features that are not annotated with pathway knowledge. Thus, we used the 5 data sets twice, first by using only those genes present in the graph structure and second by using all genes. This setup led to a total of 10 comparisons (five data sets, two analyses on each).

A detailed overview of the evaluations is given in table 3.2. In nine of these ten comparison RRFE reached a significantly (one-sided wilcoxon test, $p \leq 0.05$ was considered significant) higher AUC compared to SVM-RFE (figure 3.3, columns 3 and 6 of table 3.2). No improvements could be obtained

	SVM-RFE	RRFE	p-Value	SVM-RFE	RRFE	p-Value
Combined	0.649	0.671	0.028	0.657	0.688	0.008
GSE11121	0.614	0.667	0.016	0.588	0.657	0.016
GSE2034	0.659	0.708	0.008	0.673	0.727	0.028
GSE7390	0.528	0.516	0.345	0.519	0.536	0.016
GSE6532	0.503	0.609	0.004	0.518	0.585	0.004

Table 3.2: Median AUC obtained by cross-validation for predicting relapse events on five data sets. Columns 1 and 2 show the AUC when prior knowledge is used (RRFE) or not (SVM-RFE). Column 3 shows the p-values obtained from testing whether there is a significant difference between the AUCs. Columns 4 and 5 show the AUC for both methods when all genes were used for classification. The last column shows the p-values obtained by carrying out the same test as above.

on the data set by Loi et al. (GSE7390) when using only those genes for which pathway knowledge was available.

To further evaluate the results, the 100 genes chosen most often by both algorithms were subject of a pathway overrepresentation analysis (Fröhlich et al., 2008). Again, RRFE has most often selected genes associated with cancer. Thus, the cancer associated pathways *Cell Growth and Death* ($p = 2.656 \times 10^{-12}$), *Cancers* ($p = 1.880 \times 10^{-11}$) and *Cell cycle* ($p = 6.607 \times 10^{-08}$) were significantly overrepresented. No enriched pathways were found among the 100 most selected genes of SVM-RFE.

Several authors have pointed out, that the overlap of gene signatures obtained on different data sets is usually poor (Ein-Dor et al., 2005, 2006). To investigate this issue for gene lists produced by RRFE, the 100 most selected genes obtained on each of the 4 individual data sets, i.e. GSE2034, GSE11121, GSE7390, GSE6532, were examined in more detail. Venn diagrams were created for the gene lists obtained by RFE and RRFE (Figure 3.4) by using the VENNY software (Oliveros, 2007). It is evident that the gene lists in the upper diagram which reflects the result of RFE do not have a single gene in common. The lower panel shows the Venn diagram obtained from the RRFE gene lists. It demonstrates that the overlap increased to nine genes that are common to all four lists. Although this is still far away from being

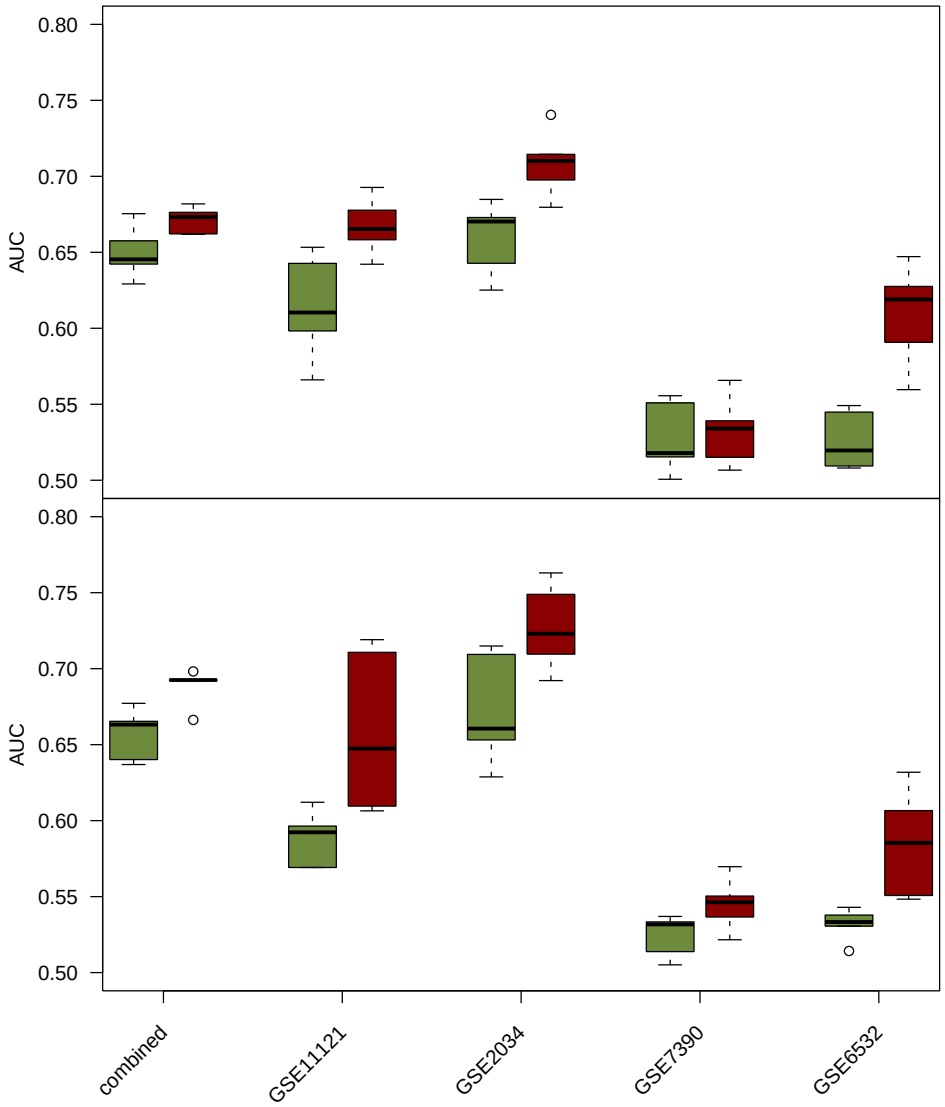

Figure 3.3: AUC for predicting if a patient will suffer from a relapse or not. The x-axis shows the 5 data sets (combined = combination of 4 data sets; GSE = GEO accession numbers.). Each boxplot consist of 5 AUCs obtained by repeating the cross-validation five times. The upper panel shows the AUC reached by SVM-RFE (red) against the AUC obtained by RRFE (green) when the algorithms were able to use only those features for which pathway knowledge was available. The lower panel has the same color code. This time, however, all features have been available for classification.

perfect it reinforces the fact the RRFE improves the gene selection in terms of stability. Furthermore, the overlap between gene signatures obtained from different data sets increases the likelihood that these gene signatures will also work well on new data sets and show the same prognostic value. Moreover, biomarkers might no longer be dependent on the microarray platform, the different probe sets and data normalization methods used. Additionally, the impact of differences in the study populations might also decrease (Sotiriou and Piccart, 2007).

In combination these results indicate that the pathway knowledge enables both increasing the classification performance and selecting biological meaningful features in a reproducible manner. Moreover, incorporating the pathway knowledge seems to decrease the susceptibility to noise in the data since the AUC could also be increased on the combined data set which is quite heterogeneous since the data was produced in different labs by different people.

Ein-Dor et al. (2005) pointed out that the presence of a gene in a gene-signature does not necessarily indicate its importance in cancer pathology. Furthermore, they said that one can produce fairly reliable multi-gene signatures by just adding enough genes, since adding lots of genes compensates for the limited predictive power of individual genes for individual patients. This is in line with the results described above, i.e. that the genes identified by RFE could not be associated to any pathway whereas RRFE used genes that are associated with cancer pathways.

It is worth mentioning, that although we could show a significant increase in classification performance when predicting relapse events the performance is still far from being perfect. Therefore, it might be better to conduct the classification in two steps: First, stratify the patients into (possibly yet unknown) molecular subgroups and subsequently perform the relapse prediction within the subgroups. Bair and Tibshirani (2004) made an attempt in this field and it might be worth to incorporate such a 'stratification'-step into pathway-based classifiers to further increase classification performance. However, we leave this point open for future research.

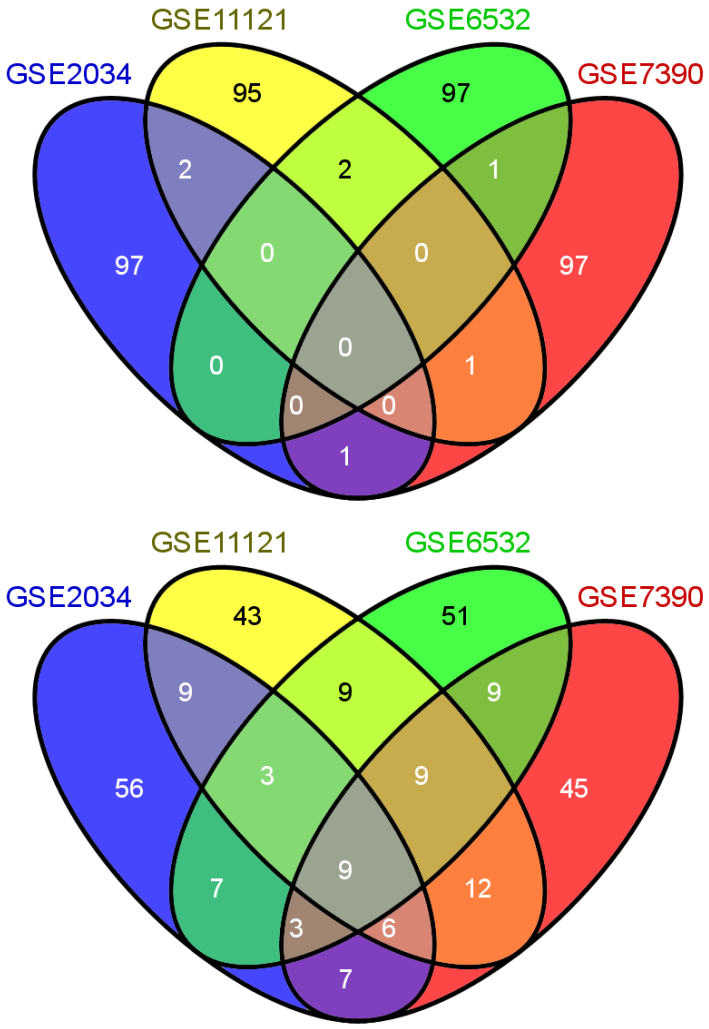

Figure 3.4: Venn diagram showing the overlap of genes between different data sets. The upper diagram shows the result of RFE and the lower panel that of RRFE.

3.1.1.3 Assessing the influence of the damping factor

As mentioned above, all experiments were performed using a damping factor (cf. equation (2.37) and Morrison et al. 2005) of $d = 0.5$. To investigate the influence of the damping factor on the classification result, it was varied from 0.1 to 0.9 on all data sets using the genes with pathway knowledge (see figure 3.5). Interestingly, $d = 0.5$ seems not to be the best choice for all data sets. The fluctuations on the individual data sets is quite high and a different choice of the damping factor might lead to better results. However, it is also evident, that the confidence intervals are mainly overlapping. Nevertheless, on the combined data set the damping factor seems not to have such a high influence since the AUC is constantly around 0.67 and the confidence intervals are narrow. Anyway, for data sets GSE6532 and GSE7390 the AUC seems to decrease as the damping factor is increased. This increase of d corresponds to a ranking mostly influenced by the pathway information (see section 2.5.2). On data set GSE2034, however, RRFE reaches its best result with a damping factor of $d = 0.6$, i.e. with higher influence of the pathway data. Also, this result has a quite narrow confidence interval.

Based on these results one can draw the conclusion, that due to its dependence on the data set the damping factor should be tuned in the cross-validation as well. However, it is always a trade-off between the number of tuning parameters and the time that is needed for training an algorithm. Therefore, if the user can accept a rather small decrease in AUC a fixed damping factor of $d = 0.5$ should be a good choice. Furthermore, it is not recommendable to use the extreme values, i.e. 0.1 and 0.9.

3.1.1.4 Assessing the influence of different pathway databases

To asses the performance of RRFE with respect to the dependence on a specific pathway database, information coming from different databases were used. As mentioned before, the pathway or interaction database is used to create an adjacency matrix, which is used internally by RRFE to rank the

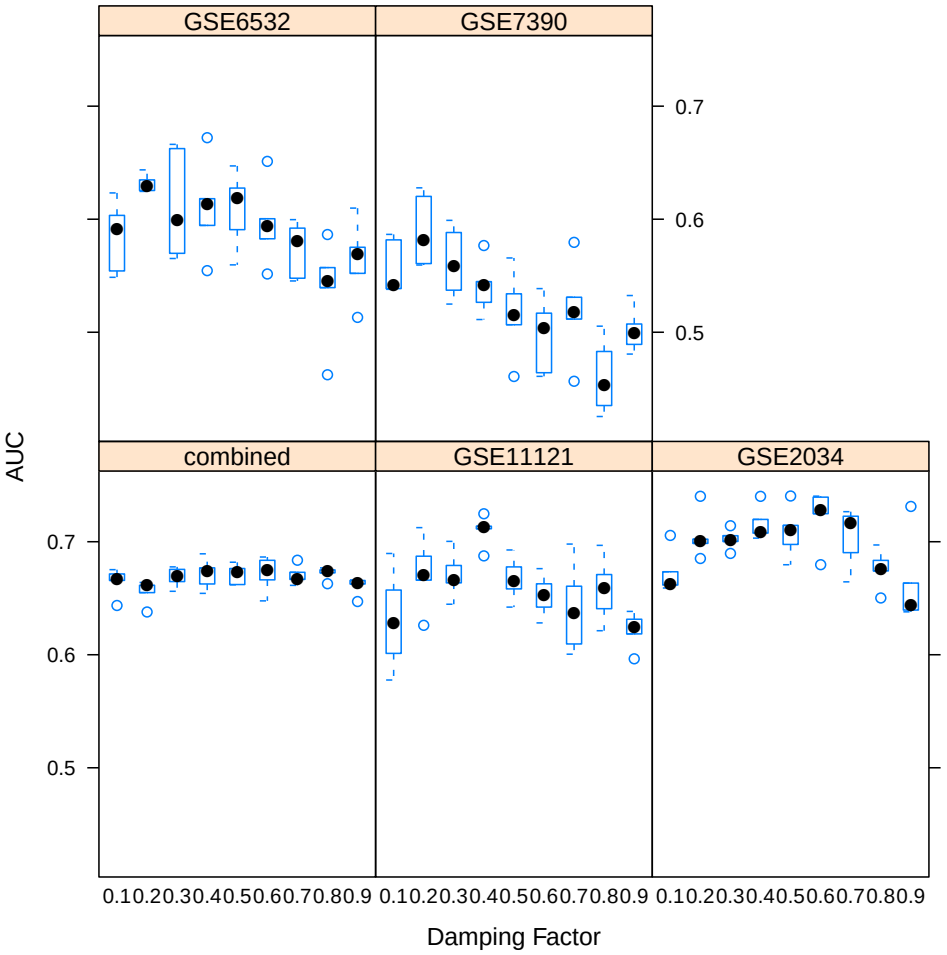

Figure 3.5: Influence of the damping factor on the AUC. The x-axis shows the value of the damping factor and the y-axis indicates the AUC obtained by five times repeated cross-validation.

genes according to their 'importance' in the network. This analysis aims at showing that the results are independent of a particular database, since the database might be changed to meet the needs of the user. All results reported so far were obtained by using PPI data coming from the HPRD. The data therein is known to be of high quality since it is manually curated (Prasad et al., 2009).

Data from CPDB and KEGG were obtained from their servers and ad-

jacency matrices were created (cf. section 2.6). Subsequently, the graph-structures and the expression data using the combined data set were used as input to RRFE. The results indicate that a change of the underlying pathway knowledge did not influence RRFE too much, since the AUC was similar for all three databases (figure 3.6). A statistical test (one-sided wilcoxon test, $p \leq 0.05$ was considered significant) did not show significant differences among the AUCs. Thus, all three databases seem to contain data that is of high quality. However, it is important to note that only the PPI part of CPDB was available for download. Therefore, the data of HPRD and CPDB can be considered similar.

Nevertheless, it is important to mention that the pathway knowledge is biased towards genes associated with common diseases. Additionally, the databases are under permanent change. Therefore, the results of pathway-based classifiers might change as the databases change. The fact that some parts of the networks are better understood than others leads to a higher number of connections in these areas compared to others. This in turn gives genes from these highly connected parts of the network a higher chance to 'survive' the feature elimination process of RRFE. However, we believe that with increasing amount and increasing quality of pathway data, the classification results of recent pathway-based classifiers should increase as well.

3.1.1.5 Comparison to other classifiers

All comparisons shown this far, have compared RRFE to its 'progenitor', i.e. SVM-RFE. However, it is also a necessity to evaluate the performance of newly developed algorithms to well established state-of-the-art methods. Therefore, RRFE was compared to other methods that are capable of incorporating *prior* knowledge into the classification as well as to algorithms that do not use any prior knowledge.

In this analysis we restricted our attention to multivariable methods that associate a large amount of features with a clinical endpoint. This kind of

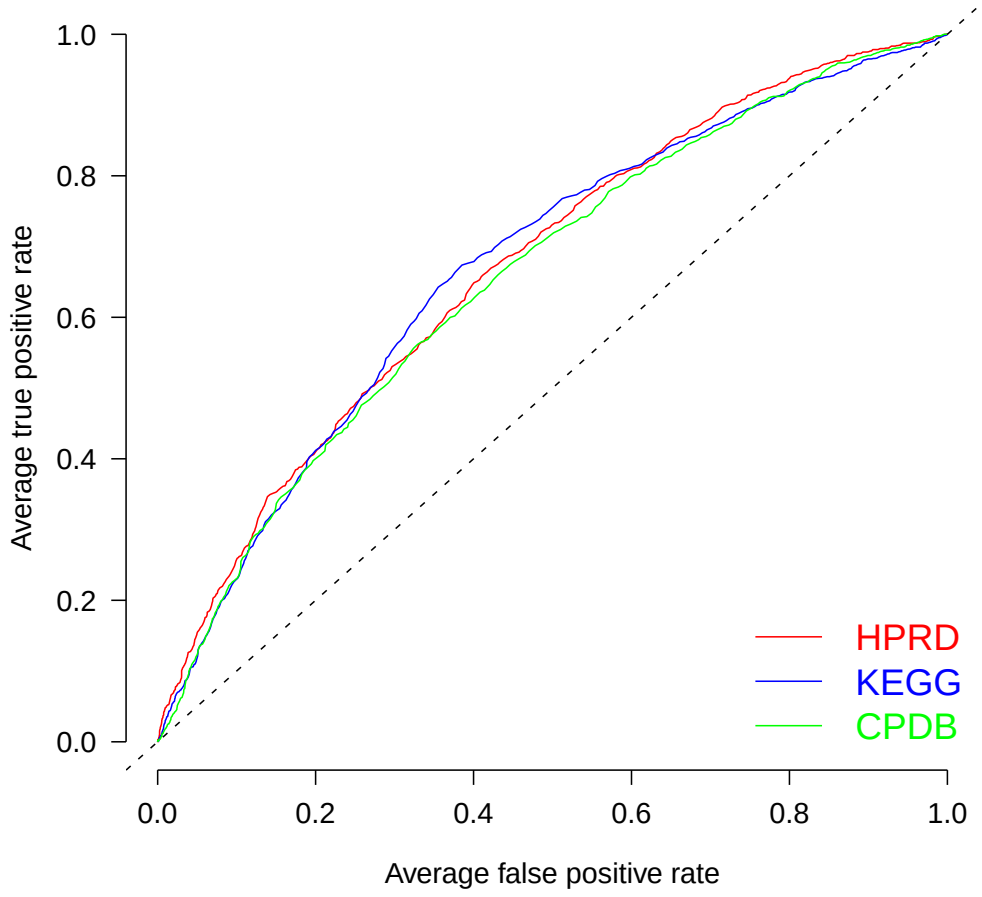

Figure 3.6: Influence of different databases on the AUC. RRFE was run on all genes which were annotated in one of the three different databases: HPRD, CPDB, KEGG.

model is not the only way for linking molecular measurements with clinical characteristics or events. There also exist a lot of test-based methods, that investigate each measurement in a univariat fashion. Wu and Lin (2009) give an overview of methods for incorporating external knowledge in such a test-based setting. However, for the purpose of identifying prognostic gene signatures multivariable modeling approaches are advantageous (Porzelius et al., 2011).

Among the numerous methods available, the nearest shrunken centroid method (PAM, Tibshirani et al. 2002) and GAMBoost (Tutz and Binder, 2005, 2006) were chosen as methods that do not use any prior knowledge. To evaluate RRFE in comparison to other methods capable of incorporating additional knowledge, the network-based SVM (Zhu et al., 2009) and PathBoost (Binder and Schumacher, 2009) were selected. In the following, the methods used will be briefly explained.

The PAM method uses conventional nearest-centroid classification (cf. Hastie et al., 2009) as a starting point. First, the squared distance of a test sample to all class centroids is calculated. Afterwards the test sample is assigned to the class whose centroid is closest. This method has the disadvantage that it needs all genes for the classification. Therefore, Tibshirani et al. (2002) modified the conventional method in such a way as it shrinks the class centroids of each gene towards the gene's overall centroid. Since most gene-measurements are noisy, their class-centroid is considered of being close to their overall centroid. Thus, after the distance between both centroids was shrunk to zero the gene does no longer contribute to the classification. This is the way how PAM performs a feature selection. The amount of shrinkage has to be determined by cross-validation.

The boosting approach GAMBoost (Tutz and Binder, 2006) is based on a penalized log-likelihood. Starting with all regression coefficients equal to zero, the coefficients of selected covariates are updated in each boosting step. For that, candidate models are fitted. The coefficient of the covariate whose candidate model resulted in the largest log-likelihood is updated. The coefficients of all other covariates remain unchanged. The main parameter, responsible for the model complexity, is the number of boosting steps. This parameter can be determined by cross-validating the predictive log-likelihood.

The PathBoost approach is an extension of GAMBoost. However, due to the pathway knowledge the connectivity of covariates is known. Therefore, if a feature is connected to another feature that already received a non-zero coefficient (i.e. entered the model), it is more likely to also receive a non-zero parameter estimate.

Method	AUC	95% CI
RRFE	0.671	0.662 - 0.681
Network-based SVM	0.607	0.574 - 0.645
PathBoost	0.564	0.547 - 0.601
PAM	0.590	0.569 - 0.623
GAMBoost	0.559	0.534 - 0.588

Table 3.3: Method comparison on the combined data set. Shown are the average AUC and the 95% confidence interval (CI) after cross-validation.

Zhu et al. (2009) developed a method, which they called network-based SVM. It uses a network-based penalty which leads to a grouped variable selection. This variable selection is achieved by penalizing the SVM objective function with an F_∞-norm (Zou and Yuan, 2008) instead of the commonly used L_1 or L_2 penalization. This norm forces the simultaneous selection or elimination of a group of features from the same pathway. Zhu et al. (2009) treat neighboring genes in a graph as a group and construct their network-based penalty as the sum of F_∞-norms of groups of neighboring genes-pairs.

All evaluations were performed in the above-mentioned cross-validation setting, i.e. five-times repeated ten-fold CV. All methods were used to predict if the patients will suffer from a relapse or not. To do so, the combined data set with 788 observations was use. Methods capable of using prior knowledge used the adjacency matrix created from HPRD.

The ROC curves obtained by the analysis are given in figure 3.7. RRFE reached the highest AUC in all comparisons (table 3.3). Indeed, it seems that predicting relapse events is a hard task for both standard and pathway-based classifiers.

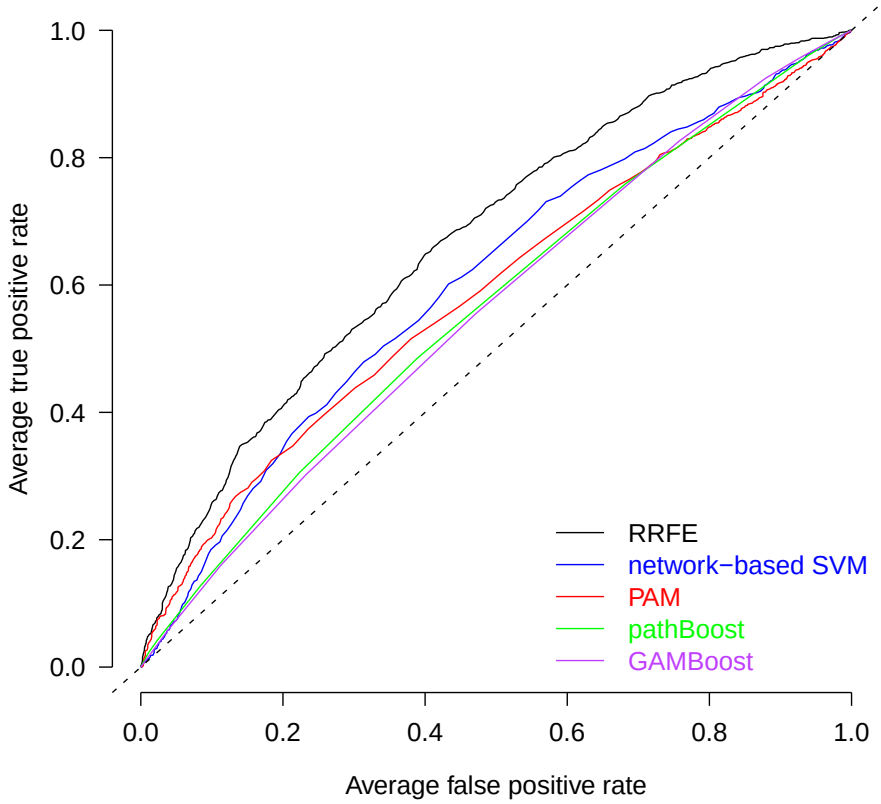

Figure 3.7: ROC Curve of RRFE compared to other classifiers.

3.2 pathClass: a software for classification with prior knowledge

We believe that publishing usable software additionally to novel methodology is an important task and valuable field. Therefore, we developed an R package, called `pathClass` (Johannes et al., 2011), that contains reference implementations of several pathway-based classifiers. The package is available at Comprehensive R Archive Network (CRAN)[1]. `pathClass` aims at providing the user with comprehensive implementations of these methods in a unified

[1] http://cran.r-project.org/

framework in order to allow easy and transparent benchmarking. To our knowledge it is the first package implementing several SVM-based algorithms that are capable of incorporating network knowledge into the classification process. It is, however, worth mentioning, that all methods available in `pathClass` have previously been published and shown their predominance over standard algorithms. For benchmarking of the more 'classical' methods not using *prior* knowledge the user is referred to the packages `CMA` (Slawski et al., 2008) and `MCRestimate` (Ruschhaupt et al., 2004). A boosting approach that is capable of using *prior* knowledge (Binder and Schumacher, 2009) can be found in the package `GAMBoost`, which is also available on CRAN.

3.2.1 Package Features

As already mentioned, all methods implemented in `pathClass` are capable of using *prior* knowledge. This knowledge is represented as a network structure or graph, i.e. it carries information on the connectivity of features. R (R Development Core Team, 2009) was chosen for the implementation of `pathClass`, since it is open source and widely applied for statistical analysis. Furthermore, the package is accompanied with a vignette which gives a detailed explanation of a typical workflow when using `pathClass`. Also, the vignette contains a benchmark of all three algorithms. In the following, the different algorithms and the way they use the network data are briefly explained.

The first algorithm implemented in `pathClass` is SVM-RFE (Guyon et al., 2002). As mentioned before, this algorithm does not use any *prior* knowledge, it rather uses SVM-based criteria to rank and remove features. RFE is included in the package to provide the user with the possibility to compare its performance to the performance of classification methods which integrate pathway knowledge.

We, of course, added RRFE, our recently proposed extension of SVM-RFE. A detailed description of the algorithm is found in section 3.1.

The next algorithm that is implemented is called network-based SVM and was recently proposed by Zhu et al. (2009). This method was already explained in the previous section, when RRFE was compared to other classifiers (cf. section 3.1.1.5). However, it is important to note that this method uses a linear program solver (`lpSolve`) for optimizing the SVM objective function. Therefore, a constraints matrix has to be created. Depending on the number of genes, this matrix can become very big. Thus, it might be necessary to discard features with smallest variability as suggested by the authors.

Additionally, we implemented the algorithm by Rapaport et al. (2007). This method defines a new metric for gene expression measurements by using the matrix exponential function. Their assumption is that most biologically relevant information is captured in the low-frequency component of expression profiles. Hence, the projection of the low-frequency component of an expression vector on the gene metabolic network should reveal areas of positive and negative expression on the graph that are likely to correspond to the activation or inhibition of specific branches of the graph.

All methods are implemented in a unified framework. That is, they can be used directly or in a cross-validation setting. `pathClass` is able to use the multi-processor architecture of modern PCs or computing clusters and run the CV in parallel, which decreases the running time tremendously. Moreover, the package provides methods to plot the results and automatically extract the features used by the classifiers.

In conclusion we hope that this software can help scientists to easily compare their newly developed methods to already existing ones. Since a thorough benchmark is required by almost all journals. Further, we hope to extend the package with additional methods as soon as possible. Given the recent growing interest in computations on powerful graphic cards (GPU), it might be possible to move some of the matrix computations from the processor to the GPU, which works in a highly parallel fashion. However, we left this open for future research.

Chapter 4

Conclusions

Clinico-pathological parameters are commonly used to predict the clinical course of breast cancer patients. However, due to the heterogeneity of breast cancer these markers have shown only limited success and patients are frequently overtreated. Therefore, new markers for breast cancer prognosis are urgently needed to avoid unnecessary treatment of patients.

Classification methods have shown to be a promising approach for detecting novel biomarkers based on microarray measurements. Given the urgent need for new biomarkers and the illustrated drawbacks of standard classifiers, we developed a new classification algorithm capable of using *prior* biological knowledge. The new method, called Reweighted Recursive Feature Elimination (RRFE), is based on the Support Vector Machine and the Recursive Feature Elimination (RFE) algorithm. When developing RRFE, our assumption was that highly connected genes should have an increased influence on the decision rule even if their fold-change is rather small. This assumption was implemented by modifying the ranking criterion of RFE by using GeneRank, a derivative of Google's PageRank algorithm.

Evaluations of RRFE showed that its advantages are three-fold: First, it showed a reproducible feature selection, meaning that the selected features were chosen independently of the training set. Second, the selected features

were associated with the disease in question, which was proven by a pathway overrepresentation analysis among the selected genes. The third advantage is that RRFE was able to predict the risk of recurrence in several breast cancer data sets with significantly higher sensitivity and specificity.

Moreover, our evaluations showed that the accuracy of RRFE was independent of the database providing the *prior* knowledge. However, it is worth noting, that the databases used in the illustrated evaluations are known to contain data that is of high quality, i.e. manually curated or experimentally validated. Additionally, it was also shown, that the overlap of gene signatures identified on different data sets was increased compared to a standard classification algorithm.

Finally, it can be concluded that RRFE might help to improve the issues that have hampered successful biomarker discovery and patient stratification. Nevertheless, the classification accuracy is still not optimal. Therefore, I hope to encourage others to follow the promising way of including *prior* knowledge into the classification to further improve classification results in several aspects.

Acknowledgements

Ich danke Herrn Prof. Dr. Roland Eils für die freundliche Übernahme des Erstgutachtens.

Mein besonderer Dank gilt Herrn Prof. Dr. Tim Beißbarth, der die vorliegende Arbeit betreut und aktiv begleitet hat. Herr Prof. Dr. Beißbarth hat mir alle Freiheiten gegeben, eigene Ideen umzusetzen und mich stets unterstüzt.

Ebenfalls möchte ich meinem Gruppenleiter, Herrn PD Dr. Holger Sültmann, für seine Unterstützung sowie die Finanzierung meiner Arbeiten und Konferenzbesuche danken.

Weiterer Dank gebührt meinen Kollegen Prof. Dr. Holger Fröhlich, Christian Bender, Stephan Gade, Dr. Ruprecht Kuner sowie Christine Porzelius für viele Diskussionen und eine tolle Arbeitsatmosphäre. Ganz besonders möchte ich mich bei Jan C. Brase bedanken, der durch seine innovativen Ideen maßgeblich zu dieser Dissertation beigetragen hat. Durch eine tolle Zusammenarbeit konnten wir gemeinsam viele Projekte erfolgreich abschließen. Och vidare tackar jag Dr. Maria Fälth Savitski för hennes outtröttliga stöd och många goda råd.

Ein ganz besonders herzlicher Dank gilt meinen wunderbaren Eltern für ihre Unterstüzung, Motivation und dafür, dass sie immer an mich geglaubt haben. Außerdem möchte ich mich ganz herzlich bei Lisa Schmidt bedanken, die immer zu mir stand und für mich da war.

References

Agarwal, P. and Searls, D. B. (**2008**). "Literature mining in support of drug discovery." *Brief Bioinform*, 9(6): 479–492. URL http://dx.doi.org/10.1093/bib/bbn035.

Aranda, B., Achuthan, P., Alam-Faruque, Y., Armean, I., Bridge, A., Derow, C., Feuermann, M., Ghanbarian, A. T., Kerrien, S., Khadake, J., Kerssemakers, J., Leroy, C., Menden, M., Michaut, M., Montecchi-Palazzi, L., Neuhauser, S. N., Orchard, S., Perreau, V., Roechert, B., van Eijk, K., and Hermjakob, H. (**2010**). "The IntAct molecular interaction database in 2010." *Nucleic Acids Res*, 38(Database issue): D525–D531. URL http://dx.doi.org/10.1093/nar/gkp878.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (**2000**). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." *Nat Genet*, 25(1): 25–29. URL http://dx.doi.org/10.1038/75556.

Bair, E. and Tibshirani, R. (**2004**). "Semi-supervised methods to predict patient survival from gene expression data." *PLoS Biol*, 2(4): E108. URL http://dx.doi.org/10.1371/journal.pbio.0020108.

Beißbarth, T. (**2004**). "GOstat: find statistically overrepresented Gene Ontologies within a group of genes." *Bioinformatics*, 20(9): 1464–1465.

Bellazzi, R. and Zupan, B. (**2007**). "Towards knowledge-based gene expression data mining." *J Biomed Inform*, 40(6): 787–802. URL http://dx.doi.org/10.1016/j.jbi.2007.06.005.

Bellman, R. (**1961**). *Adaptive Control Processes*. Princeton University Press.

Bennett, K. P. and Mangasarian, O. L. (**1992**). "Robust linear programming discrimination of two linearly inseparable sets." *Optimization Methods and Software*, 1(1): 23–34. URL http://dx.doi.org/10.1080/10556789208805504.

Bianchini, M., Gori, M., and Scarselli, F. (**2005**). "Inside PageRank." *ACM Trans. Internet Technol.*, 5: 92–128. URL http://doi.acm.org/10.1145/1052934.1052938.

Binder, H. and Schumacher, M. (**2009**). "Incorporating pathway information into boosting estimation of high-dimensional risk prediction models." *BMC Bioinformatics*, 10: 18. URL http://dx.doi.org/10.1186/1471-2105-10-18.

Biomarkers Definitions Working Group (**2001**). "Biomarkers and surrogate endpoints: preferred definitions and conceptual framework." *Clin Pharmacol Ther*, 69(3): 89–95.

Bolstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. (**2003**). "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias." *Bioinformatics*, 19(2): 185–193.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (**1992**). "A training algorithm for optimal margin classifiers." In "COLT '92: Proceedings of the fifth annual workshop on Computational learning theory," pages 144–152. ACM, New York, NY, USA. URL http://doi.acm.org/10.1145/130385.130401.

Bradley, A. P. (**1997**). "The use of the area under the ROC curve in the evaluation of machine learning algorithms." *Pattern Recognition*, 30: 1145–1159.

Brase, J. C., Schmidt, M., Fischbach, T., Sültmann, H., Bojar, H., Koelbl, H., Hellwig, B., Rahnenführer, J., Hengstler, J. G., and Gehrmann, M. C. (**2010**). "ERBB2 and TOP2A in Breast Cancer: A Comprehensive Analysis of Gene Amplification, RNA Levels, and Protein Expression and Their Influence on Prognosis and Prediction." *Clin Cancer Res*. URL http://dx.doi.org/10.1158/1078-0432.CCR-09-2471.

Brin, S. and Page, L. (**1998**). "The anatomy of a large-scale hypertextual Web search engine." *Computer Networks and ISDN Systems*, 30(1-7): 107 – 117. URL http://www.sciencedirect.com/science/article/B6TYT-3WRC342-2N/2/63e7d8fb6a64027a0c15e6ae3e402889. Proceedings of the Seventh International World Wide Web Conference.

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., and Haussler, D. (**2000**). "Knowledge-based analysis of microarray gene expression data by using support vector machines." *Proc Natl Acad Sci U S A*, 97(1): 262–267.

Burges, C. (**1998**). "A Tutorial on Support Vector Machines for Pattern Recognition." *Data Mining and Knowledge Discovery*, 2(2): 121–167. URL http://www.springerlink.com/index/Q87856173126771Q.pdf.

Buyse, M., Loi, S., van't Veer, L., Viale, G., Delorenzi, M., Glas, A. M., d'Assignies, M. S., Bergh, J., Lidereau, R., Ellis, P., Harris, A., Bogaerts, J., Therasse, P., Floore, A., Amakrane, M., Piette, F., Rutgers, E., Sotiriou, C., Cardoso, F., Piccart, M. J., and Consortium, T. R. A. N. S. B. I. G. (**2006**). "Validation and clinical utility of a 70-gene prognostic signature for women with node-negative breast cancer." *J Natl Cancer Inst*, 98(17): 1183–1192.

Carlson, M., Falcon, S., Pages, H., and Li, N. (**2009**). *hgu133a.db: Affymetrix Human Genome U133 Set annotation data (chip hgu133a)*. R package version 2.2.12.

Carter, C. L., Allen, C., and Henson, D. E. (**1989**). "Relation of tumour size, lymph node status, and survival in 24,740 breast cancer cases." *Cancer*, 63: 181–187. URL http://dx.doi.org/10.1002/1097-0142(19890101)63:1%3C181::AID-CNCR2820630129%3E3.0.CO;2-H.

Ceol, A., Aryamontri, A. C., Licata, L., Peluso, D., Briganti, L., Perfetto, L., Castagnoli, L., and Cesareni, G. (**2010**). "MINT, the molecular interaction database: 2009 update." *Nucleic Acids Res*, 38(Database issue): D532–D539. URL http://dx.doi.org/10.1093/nar/gkp983.

Chapelle, O. (**2007**). "Training a Support Vector Machine in the Primal." *Neural Comp.*, 19(5): 1155–1178. URL http://neco.mitpress.org/cgi/content/abstract/19/5/1155.

Chapelle, O. and Vapnik, V. (**2000**a). "Bounds on error expectation for support vector machines." *Neural Computation*, 12(9): 2013–2036. URL http://www.mitpressjournals.org/doi/abs/10.1162/089976600300015042.

Chapelle, O. and Vapnik, V. (**2000**b). "Model Selection for Support Vector Machines."

Choi, C., Krull, M., Kel, A., Kel-Margoulis, O., Pistor, S., Potapov, A., Voss, N., and Wingender, E. (**2004**). "TRANSPATH–a high quality database focused on signal transduction." *Comp Funct Genomics*, 5(2): 163–168. URL http://dx.doi.org/10.1002/cfg.386.

Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., and Ideker, T. (**2007**). "Network-based classification of breast cancer metastasis." *Mol Syst Biol*, 3: 10.

Cristianini, N. and Shawe-Taylor, J. (**2000**). *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press, 1 edition. URL http://www.worldcat.org/isbn/0521780195.

Desmedt, C., Piette, F., Loi, S., Wang, Y., Lallemand, F., Haibe-Kains, B., Viale, G., Delorenzi, M., Zhang, Y., d'Assignies, M. S., Bergh, J., Lidereau, R., Ellis, P., Harris, A. L., Klijn, J. G. M., Foekens, J. A., Cardoso, F., Piccart, M. J., Buyse, M., Sotiriou, C., and Consortium, T. R. A. N. S. B. I. G. (**2007**). "Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the TRANSBIG multicenter independent validation series." *Clin Cancer Res*, 13(11): 3207–3214.

Doms, A. and Schroeder, M. (**2005**). "GoPubMed: exploring PubMed with the Gene Ontology." *Nucleic Acids Res*, 33(Web Server issue): W783–W786. URL http://dx.doi.org/10.1093/nar/gki470.

Duda, R. O. and Hart, P. E. (**1973**). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY.

Dudoit, S., Fridlyand, J., and Speed, T. P. (**2002**). "Comparison of discrimination methods for the classification of tumors using gene expression data." *Journal of the American Statistical Association*, 97(457): 77–87.

Egan, J. (**1975**). *Signal detection theory and ROC analysis*. Academic Press, New York.

Eifel, P., Axelson, J. A., Costa, J., Crowley, J., Curran, W. J., Deshler, A., Fulton, S., Hendricks, C. B., Kemeny, M., Kornblith, A. B., Louis, T. A., Markman, M., Mayer, R., and Roter, D. (**2001**). "National Institutes of Health Consensus Development Conference Statement: adjuvant therapy for breast cancer, November 1-3, 2000." *J Natl Cancer Inst*, 93(13): 979–989.

Ein-Dor, L., Kela, I., Getz, G., Givol, D., and Domany, E. (**2005**). "Outcome signature genes in breast cancer: is there a unique set?" *Bioinformatics*, 21(2): 171–178. URL http://dx.doi.org/10.1093/bioinformatics/bth469.

Ein-Dor, L., Zuk, O., and Domany, E. (**2006**). "Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer." *Proc Natl Acad Sci U S A*, 103(15): 5923–5928. URL http://dx.doi.org/10.1073/pnas.0601231103.

Ekins, S., Nikolsky, Y., Bugrim, A., Kirillov, E., and Nikolskaya, T. (**2007**). "Pathway mapping tools for analysis of high content data." *Methods Mol Biol*, 356: 319–350.

Elston, C. W. and Ellis, I. O. (**1991**). "Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with long-term follow-up." *Histopathology*, 19: 403–410. URL http://dx.doi.org/10.1111/j.1365-2559.1991.tb00229.x.

Emens, L. A. (**2005**). "Trastuzumab: targeted therapy for the management of HER-2/neu-overexpressing metastatic breast cancer." *Am J Ther*, 12(3): 243–253.

Fan, C., Oh, D. S., Wessels, L., Weigelt, B., Nuyten, D. S. A., Nobel, A. B., van't Veer, L. J., and Perou, C. M. (**2006**). "Concordance among gene-expression-based predictors for breast cancer." *N Engl J Med*, 355(6): 560–569. URL http://dx.doi.org/10.1056/NEJMoa052933.

Fawcett, T. (**2004**). "ROC graphs: Notes and practical considerations for researchers." *Machine Learning*.

Febbo, P. G. and Kantoff, P. W. (**2006**). "Noise and bias in microarray analysis of tumor specimens." *J Clin Oncol*, 24(23): 3719–3721. URL http://dx.doi.org/10.1200/JCO.2006.06.7942.

Fisher, R. (**1922**). "On the Interpretation of χ from Contingency Tables, and the Calculation of Pc." *Journal of the Royal Statistical Society*, 85(1): 87–94. URL http://dx.doi.org/10.2307/2340521.

Flach, P. A. and Wu, S. (**2005**). "Repairing concavities in ROC curves." In "Proceedings of the 19th international joint conference on Artificial intelligence," pages 702–707. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. URL http://portal.acm.org/citation.cfm?id=1642293.1642406.

Fröhlich, H., Fellmann, M., Sueltmann, H., Poustka, A., and Beissbarth, T. (**2008**). "Predicting pathway membership via domain signatures." *Bioinformatics*, 24(19): 2137–2142. URL http://dx.doi.org/10.1093/bioinformatics/btn403.

Fröhlich, H., Speer, N., Poustka, A., and Beissbarth, T. (**2007**). "GOSim–an R-package for computation of information theoretic GO similarities between terms and gene products." *BMC Bioinformatics*, 8: 166. URL http://dx.doi.org/10.1186/1471-2105-8-166.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (**2000**). "Support vector machine classification and validation of cancer tissue samples using microarray expression data." *Bioinformatics*, 16(10): 906–914.

Ganter, B. and Giroux, C. N. (**2008**). "Emerging applications of network and pathway analysis in drug discovery and development." *Curr Opin Drug Discov Devel*, 11(1): 86–94.

Garnis, C., Buys, T., and Lam, W. (**2004**). "Genetic alteration and gene expression modulation during cancer progression." *Molecular Cancer*, 3(1): 9. URL http://www.molecular-cancer.com/content/3/1/9.

Geisser, S. (**1975**). "The Predictive Sample Reuse Method with Applications." *Journal of the American Statistical Association*, 70(350): 320–328. URL http://www.jstor.org/stable/2285815.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y. H., and Zhang, J. (**2004**). "Bioconductor: Open software development for computational biology and

bioinformatics." *Genome Biology*, 5: R80. URL http://genomebiology.com/2004/5/10/R80.

Gill, P. E., Murray, W., and Wright, M. H. (**1981**). *Practical Optimization*. Academic Press, London. URL http://www.apcatalog.com/cgi-bin/AP?ISBN=0122839528&LOCATION=US&FORM=FORM2.

Golub, G. and van der Vorst, H. (**2000**). "Eigenvalue computation in the 20th century." *Journal of Computational and Applied Mathematics*, 123: 35–65(31). URL http://www.ingentaconnect.com/content/els/03770427/2000/00000123/00000001/art00413.

Golub, G. H. and Van Loan, C. F. (**1996**). *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 3rd edition. URL http://www.worldcat.org/isbn/0801854148.

Golub, T. R. (**1999**). "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring." *Science*, 286(5439): 531–537.

Guyon, I. and Elisseeff, A. (**2003**). "An introduction to variable and feature selection." *The Journal of Machine Learning Research*. URL http://portal.acm.org/citation.cfm?id=944919.944968.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (**2002**). "Gene Selection for Cancer Classification using Support Vector Machines." *Machine Learning*, 46(1-3): 389–422. URL http://www.springerlink.com/index/W68424066825VR3L.pdf.

Hanahan, D. and Weinberg, R. A. (**2000**). "The hallmarks of cancer." *Cell*, 100(1): 57–70.

Hanahan, D. and Weinberg, R. A. (**2011**). "Hallmarks of cancer: the next generation." *Cell*, 144(5): 646–674. URL http://dx.doi.org/10.1016/j.cell.2011.02.013.

Hanley, J. A. and McNeil, B. J. (**1982**). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, 143(1): 29–36.

Hastie, T., Tibshirani, R., and Friedman, J. (**2009**). *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction*. Springer.

Huttenhower, C., Hibbs, M. A., Myers, C. L., Caudy, A. A., Hess, D. C., and Troyanskaya, O. G. (**2009**). "The impact of incomplete knowledge on evaluation: an experimental benchmark for protein function prediction." *Bioinformatics*, 25(18): 2404–2410. URL http://bioinformatics.oxfordjournals.org/content/25/18/2404.abstract.

Irizarry, R., Hobbs, B., Collin, F., and Beazer-Barclay, Y. (**2003**). "Exploration, normalization, and summaries of high density oligonucleotide array probe level data." *Biostatistics*. URL http://pt.wkhealth.com/pt/re/bist/abstract.00134745-200304000-00007.htm.

Jaakkola, T. S. and Haussler, D. (**1999**). "Probabilistic kernel regression models." In "Proceedings of the 1999 Conference on AI and Statistics," URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.8322.

Jemal, A., Siegel, R., Xu, J., and Ward, E. (**2010**). "Cancer statistics, 2010." *CA Cancer J Clin*, 60(5): 277–300. URL http://dx.doi.org/10.3322/caac.20073.

Jensen, L. J., Saric, J., and Bork, P. (**2006**). "Literature mining for the biologist: from information retrieval to biological discovery." *Nat Rev Genet*, 7(2): 119–129. URL http://dx.doi.org/10.1038/nrg1768.

Johannes, M., Brase, J. C., Fröhlich, H., Gade, S., Gehrmann, M., Fälth, M., Sültmann, H., and Beißbarth, T. (**2010**). "Integration Of Pathway Knowledge Into A Reweighted Recursive Feature Elimination Approach For Risk Stratification Of Cancer Patients." *Bioinformatics*, 26(17): 2136–2144. URL http://dx.doi.org/10.1093/bioinformatics/btq345.

Johannes, M., Fröhlich, H., Sültmann, H., and Beißbarth, T. (**2011**). "`pathClass`: An R-Package for Integration of Pathway Knowledge into Support Vector Machines for Biomarker Discovery." *Bioinformatics*, (submitted).

John, G. H., Kohavi, R., and Pfleger, K. (**1994**). "Irrelevant Features and the Subset Selection Problem." In "Proceedings of the 11th International Conference on Machine Learning," pages 121–129. Morgan Kaufmann.

Kamburov, A., Wierling, C., Lehrach, H., and Herwig, R. (**2009**). "ConsensusPathDB–a database for integrating human functional interaction

networks." *Nucleic Acids Res*, 37(Database issue): D623–D628. URL http://dx.doi.org/10.1093/nar/gkn698.

Kanehisa, M. and Goto, S. (**2000**). "KEGG: kyoto encyclopedia of genes and genomes." *Nucleic Acids Res*, 28(1): 27–30.

Kohavi, R. (**1995**). "A study of cross-validation and bootstrap for accuracy estimation and model selection." In "IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence," pages 1137–1143. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Kondor, R. I. and Lafferty, J. (**2002**). "Diffusion Kernels on Graphs and Other Discrete Structures." In "Proceedings of the ICML," .

Kuhn, H. and Tucker, A. (**1951**). "Nonlinear programming." In "Proc. 2nd Berkely Symposium on Mathematical Statistics and Probabilistics," pages 481–492. University of California Press.

Langley, P. (**1994**). "Selection of Relevant Features in Machine Learning." In "In Proceedings of the AAAI Fall symposium on relevance," pages 140–144. AAAI Press.

Langville, A. N. and Meyer, C. D. (**2004**). "Deeper inside PageRank." *Internet Mathematics*, 1: 2004.

Lee, E., Chuang, H.-Y., Kim, J.-W., Ideker, T., and Lee, D. (**2008**). "Inferring pathway activity toward precise disease classification." *PLoS Comput Biol*, 4(11): e1000 217. URL http://dx.doi.org/10.1371/journal.pcbi.1000217.

Loi, S., Haibe-Kains, B., Desmedt, C., Lallemand, F., Tutt, A. M., Gillet, C., Ellis, P., Harris, A., Bergh, J., Foekens, J. A., Klijn, J. G. M., Larsimont, D., Buyse, M., Bontempi, G., Delorenzi, M., Piccart, M. J., and Sotiriou, C. (**2007**). "Definition of clinically distinct molecular subtypes in estrogen receptor-positive breast carcinomas through genomic grade." *J Clin Oncol*, 25(10): 1239–1246. URL http://dx.doi.org/10.1200/JCO.2006.07.1522.

Luntz, A. and Brailovsky, V. (**1969**). "On estimation of characters obtained in statistical procedure of recognition." *Technicheskaya Kibernetica*, 3.

McLachlan, G. J., Do, K.-A., and Ambroise, C. (**2005**). "Discriminant Analysis." In "Analyzing Microarray Gene Expression Data," pages 185–220.

John Wiley & Sons, Inc. URL http://dx.doi.org/10.1002/047172842X.ch6.

Meyer, C. (**2001**). *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics. URL http://www.worldcat.org/isbn/0898714540.

Moler, E. J., Chow, M. L., and Mian, I. S. (**2000**). "Analysis of molecular profile data using generative and discriminative methods." *Physiol. Genomics*, 4(2): 109–126. URL http://physiolgenomics.physiology.org/cgi/content/abstract/4/2/109.

Morrison, J. L., Breitling, R., Higham, D. J., and Gilbert, D. R. (**2005**). "GeneRank: using search engine technology for the analysis of microarray experiments." *BMC Bioinformatics*, 6: 233. URL http://dx.doi.org/10.1186/1471-2105-6-233.

Mosteller, F. and Turkey, J. (**1968**). "Data analysis, including statistics." In G. Lindzey and E. Aronson, editors, "Handbook of Social Psychology, Vol. 2," Addison-Wesley, Reading, MA.

Nacu, S., Critchley-Thorne, R., Lee, P., and Holmes, S. (**2007**). "Gene expression network analysis and applications to immunology." *Bioinformatics*, 23(7): 850–858. URL http://dx.doi.org/10.1093/bioinformatics/btm019.

Neter, J., Kutner, M. H., and Wasserman, W. (**1990**). *Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Designs*. McGraw-Hill, 3rd edition.

Oliveros, J. (**2007**). "VENNY. An interactive tool for comparing lists with Venn Diagrams." URL http://bioinfogp.cnb.csic.es/tools/venny/index.html.

Opper, M. and Winther, O. (**2000**). "Gaussian process classification and SVM: mean field results and leave-one-out estimator." In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, "Advances in Large Margin Classifiers," pages 311–326. MIT Press.

Page, L., Brin, S., Motwani, R., and Winograd, T. (**1999**). "The PageRank Citation Ranking: Bringing Order to the Web." Technical Report 1999-66, Stanford InfoLab. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

Pearson, K. (**1901**). "On lines and planes of closest fit to systems of points in space." *Philosophical Magazine*, 2(6): 559–572.

Perou, C. M., Sorlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S. X., Lonning, P. E., Borresen-Dale, A. L., Brown, P. O., and Botstein, D. (**2000**). "Molecular portraits of human breast tumours." *Nature*, 406(6797): 747–752. URL http://dx.doi.org/10.1038/35021093.

Perron, O. (**1907**). "Zur Theorie der Matrices." *Mathematische Annalen*, 64: 248–263. URL http://dx.doi.org/10.1007/BF01449896. 10.1007/BF01449896.

Pisani, P., Parkin, D. M., Bray, F., and Ferlay, J. (**1999**). "Estimates of the worldwide mortality from 25 cancers in 1990." *Int. J. Cancer*, 83(1): 18–29. URL http://dx.doi.org/10.1002/(SICI)1097-0215(19990924)83:1<18::AID-IJC5>3.0.CO;2-M.

Ponting, C. P. (**2008**). "The functional repertoires of metazoan genomes." *Nat Rev Genet*, 9(9): 689–698. URL http://dx.doi.org/10.1038/nrg2413.

Portales-Casamar, E., Thongjuea, S., Kwon, A. T., Arenillas, D., Zhao, X., Valen, E., Yusuf, D., Lenhard, B., Wasserman, W. W., and Sandelin, A. (**2010**). "JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles." *Nucleic Acids Res*, 38(Database issue): D105–D110. URL http://dx.doi.org/10.1093/nar/gkp950.

Porzelius, C., Johannes, M., Binder, H., and Beißbarth, T. (**2011**). "Leveraging external knowledge on molecular interactions in classification methods for risk prediction of patients." *Biom J*, 53(2): 190–201. URL http://dx.doi.org/10.1002/bimj.201000155.

Prasad, T. S. K., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., Balakrishnan, L., Marimuthu, A., Banerjee, S., Somanathan, D. S., Sebastian, A., Rani, S., Ray, S., Kishore, C. J. H., Kanth, S., Ahmed, M., Kashyap, M. K., Mohmood, R., Ramachandra, Y. L., Krishna, V., Rahiman, B. A., Mohan, S., Ranganathan, P., Ramabadran, S., Chaerkady,

R., and Pandey, A. (**2009**). "Human Protein Reference Database–2009 update." *Nucleic Acids Res*, 37(Database issue): D767–D772. URL http://dx.doi.org/10.1093/nar/gkn892.

Preston-Martin, S., Pike, M. C., Ross, R. K., Jones, P. A., and Henderson, B. E. (**1990**). "Increased Cell Division as a Cause of Human Cancer." *Cancer Research*, 50(23): 7415–7421. URL http://cancerres.aacrjournals.org/content/50/23/7415.abstract.

Quinlan, J. R. (**1986**). "Induction of Decision Trees." *Mach. Learn.*, 1: 81–106. URL http://portal.acm.org/citation.cfm?id=637962.637969.

R Development Core Team (**2009**). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org. ISBN 3-900051-07-0.

Rakotomamonjy, A. (**2003**). "Variable selection using svm based criteria." *J. Mach. Learn. Res.*, 3: 1357–1370. URL http://portal.acm.org/citation.cfm?id=944919.944977.

Ransohoff, D. F. (**2005**). "Bias as a threat to the validity of cancer molecular-marker research." *Nat Rev Cancer*, 5(2): 142–149. URL http://dx.doi.org/10.1038/nrc1550.

Rapaport, F., Zinovyev, A., Dutreix, M., Barillot, E., and Vert, J.-P. (**2007**). "Classification of microarray data using gene networks." *BMC Bioinformatics*, 8: 35. URL http://dx.doi.org/10.1186/1471-2105-8-35.

Rody, A., Karn, T., Ruckhäberle, E., Müller, V., Gehrmann, M., Solbach, C., Ahr, A., Gätje, R., Holtrich, U., and Kaufmann, M. (**2009**). "Gene expression of topoisomerase II alpha (TOP2A) by microarray analysis is highly prognostic in estrogen receptor (ER) positive breast cancer." *Breast Cancer Res Treat*, 113(3): 457–466. URL http://dx.doi.org/10.1007/s10549-008-9964-x.

Roepman, P., Horlings, H. M., Krijgsman, O., Kok, M., de Mesquita, J. M. B., Bender, R., Linn, S. C., Glas, A. M., and van de Vijver, M. J. (**2009**). "Microarray-Based Determination of Estrogen Receptor, Progesterone Receptor, and HER2 Receptor Status in Breast Cancer." *Clin Cancer Res*, 15: 7003–7011. URL http://dx.doi.org/10.1158/1078-0432.CCR-09-0449.

Rouzier, R., Perou, C. M., Symmans, W. F., Ibrahim, N., Cristofanilli, M., Anderson, K., Hess, K. R., Stec, J., Ayers, M., Wagner, P., Morandi, P., Fan, C., Rabiul, I., Ross, J. S., Hortobagyi, G. N., and Pusztai, L. (**2005**). "Breast cancer molecular subtypes respond differently to preoperative chemotherapy." *Clin Cancer Res*, 11(16): 5678–5685. URL http://dx.doi.org/10.1158/1078-0432.CCR-04-2421.

Ruschhaupt, M., Huber, W., Poustka, A., and Mansmann, U. (**2004**). "A Compendium to Ensure Computational Reproducibility in High-Dimensional Classification Tasks." *Statistical Applications in Genetics and Molecular Biology*. URL http://www.klinikum.uni-heidelberg.de/fileadmin/inst_med_biometrie/pdf/compHuang.pdf.

Schmidt, M., Böhm, D., von Törne, C., Steiner, E., Puhl, A., Pilch, H., Lehr, H.-A., Hengstler, J. G., Kölbl, H., and Gehrmann, M. (**2008**). "The humoral immune system has a key prognostic impact in node-negative breast cancer." *Cancer Res*, 68(13): 5405–5413. URL http://dx.doi.org/10.1158/0008-5472.CAN-07-5206.

Schölkopf, B., Smola, A., and Müller, K.-R. (**1998**). "Nonlinear Component Analysis as a Kernel Eigenvalue Problem." *Neural Computation*, 10(5): 1299–1319. URL http://www.mitpressjournals.org/doi/abs/10.1162/089976698300017467.

Schölkopf, B. and Smola, A. J. (**2001**). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA.

Schölkopf, B., Tsuda, K., and Vert, J., editors (**2004**). *Kernel Methods in Computational Biology*. MIT Press.

Segal, N. H., Pavlidis, P., Noble, W. S., Antonescu, C. R., Viale, A., Wesley, U. V., Busam, K., Gallardo, H., DeSantis, D., Brennan, M. F., Cordon-Cardo, C., Wolchok, J. D., and Houghton, A. N. (**2003**). "Classification of Clear-Cell Sarcoma as a Subtype of Melanoma by Genomic Profiling." *Journal of Clinical Oncology*, 21(9): 1775–1781. URL http://jco.ascopubs.org/content/21/9/1775.abstract.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (**2005**). "ROCR: visualizing classifier performance in R." *Bioinformatics*, 21(20): 3940–3941. URL http://dx.doi.org/10.1093/bioinformatics/bti623.

Slamon, D. J., Clark, G. M., Wong, S. G., Levin, W. J., Ullrich, A., and McGuire, W. L. (**1987**). "Human breast cancer: correlation of relapse and survival with amplification of the HER-2/neu oncogene." *Science*, 235(4785): 177–182.

Slamon, D. J., Godolphin, W., Jones, L. A., Holt, J. A., Wong, S. G., Keith, D. E., Levin, W. J., Stuart, S. G., Udove, J., and Ullrich, A. (**1989**). "Studies of the HER-2/neu proto-oncogene in human breast and ovarian cancer." *Science*, 244(4905): 707–712.

Slawski, M., Daumer, M., and Boulesteix, A.-L. (**2008**). "CMA - A comprehensive Bioconductor package for supervised classification with high dimensional data." *BMC Bioinformatics*, 9(1): 439.

Smyth, G. K. (**2005**). "Limma: linear models for microarray data." In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, editors, "Bioinformatics and Computational Biology Solutions using R and Bioconductor," pages 397–420. Springer, New York.

Sorlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., Hastie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Thorsen, T., Quist, H., Matese, J. C., Brown, P. O., Botstein, D., Lonning, P. E., and Borresen-Dale, A. L. (**2001**). "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications." *Proc Natl Acad Sci U S A*, 98(19): 10 869–10 874. URL http://dx.doi.org/10.1073/pnas.191367098.

Sorlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., Geisler, S., Demeter, J., Perou, C. M., Lonning, P. E., Brown, P. O., Borresen-Dale, A.-L., and Botstein, D. (**2003**). "Repeated observation of breast tumor subtypes in independent gene expression data sets." *Proc Natl Acad Sci U S A*, 100(14): 8418–8423. URL http://dx.doi.org/10.1073/pnas.0932692100.

Sotiriou, C. and Piccart, M. J. (**2007**). "Taking gene-expression profiling to the clinic: when will molecular signatures become relevant to patient care?" *Nat Rev Cancer*, 7(7): 545–553. URL http://dx.doi.org/10.1038/nrc2173.

Spackman, K. A. (**1989**). "Signal Detection Theory: Valuable Tools for Evaluating Inductive Learning." In A. M. Segre, editor, "Proceedings of the Sixth International Workshop on Machine Learning (ML 1989), Cornell

University, Ithaca, New York, USA, June 26-27, 1989," pages 160–163. Morgan Kaufmann, San Mateo, CA.

Su, J., Yoon, B.-J., and Dougherty, E. R. (**2009**). "Accurate and reliable cancer classification based on probabilistic inference of pathway activity." *PLoS One*, 4(12): e8161. URL http://dx.doi.org/10.1371/journal.pone.0008161.

Tan, P. K., Downey, T. J., Spitznagel, E. L., Xu, P., Fu, D., Dimitrov, D. S., Lempicki, R. A., Raaka, B. M., and Cam, M. C. (**2003**). "Evaluation of gene expression measurements from commercial microarray platforms." *Nucleic Acids Res*, 31(19): 5676–5684.

Tavassoli, F. A. and Devilee, P., editors (**2003**). *World Health Organization Classification of Tumours. Pathology and Genetics of Tumours of the Breast and Female Genital Organs*. IARC Press, Lyon.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (**2002**). "Diagnosis of multiple cancer types by shrunken centroids of gene expression." *Proceedings of the National Academy of Sciences*. URL http://www.pnas.org/cgi/content/abstract/99/10/6567.

Tilstone, C. (**2003**). "DNA microarrays: Vital statistics." *Nature*, 424(6949): 610–612. URL http://dx.doi.org/10.1038/424610a.

Tusher, V. G., Tibshirani, R., and Chu, G. (**2001**). "Significance analysis of microarrays applied to the ionizing radiation response." *Proc Natl Acad Sci U S A*, 98(9): 5116–5121. URL http://dx.doi.org/10.1073/pnas.091062498.

Tutz, G. and Binder, H. (**2005**). "Boosting ridge regression." Discussion Paper 418, SFB 386, Ludwig-Maximilians-University Munich.

Tutz, G. and Binder, H. (**2006**). "Generalized additive modeling with implicit variable selection by likelihood-based boosting." *Biometrics*, 62(4): 961–971. URL http://dx.doi.org/10.1111/j.1541-0420.2006.00578.x.

van de Vijver, M. J., He, Y. D., van't Veer, L. J., Dai, H., Hart, A. A. M., Voskuil, D. W., Schreiber, G. J., Peterse, J. L., Roberts, C., Marton, M. J., Parrish, M., Atsma, D., Witteveen, A., Glas, A., Delahaye, L., van der Velde, T., Bartelink, H., Rodenhuis, S., Rutgers, E. T., Friend, S. H., and Bernards, R. (**2002**). "A gene-expression signature as a predictor of survival in breast cancer." *N*

Engl J Med, 347(25): 1999–2009. URL http://dx.doi.org/10.1056/NEJMoa021967.

van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (**2002**). "Gene expression profiling predicts clinical outcome of breast cancer." *Nature*, 415(6871): 530–536. URL http://dx.doi.org/10.1038/415530a.

Vanteru, B. C., Shaik, J. S., and Yeasin, M. (**2008**). "Semantically linking and browsing PubMed abstracts with gene ontology." *BMC Genomics*, 9 Suppl 1: S10. URL http://dx.doi.org/10.1186/1471-2164-9-S1-S10.

Vapnik, V. (**1995**). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Vapnik, V. and Cortes, C. (**1995**). "Support-vector networks." *Machine Learning*. URL http://www.springerlink.com/index/K238JX04HM87J80G.pdf.

Wang, Y., Klijn, J. G. M., Zhang, Y., Sieuwerts, A. M., Look, M. P., Yang, F., Talantov, D., Timmermans, M., van Gelder, M. E. M., Yu, J., Jatkoe, T., Berns, E. M. J. J., Atkins, D., and Foekens, J. A. (**2005**). "Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer." *Lancet*, 365(9460): 671–679. URL http://dx.doi.org/10.1016/S0140-6736(05)17947-1.

Weigelt, B., Peterse, J. L., and van't Veer, L. J. (**2005**). "Breast cancer metastasis: markers and models." *Nat Rev Cancer*, 5(8): 591–602. URL http://dx.doi.org/10.1038/nrc1670.

Weinberg, R. A. (**2006**). *The Biology of Cancer*. Garland Science Textbooks, 1 edition. URL http://www.worldcat.org/isbn/0815340788.

Wingender, E. (**2008**). "The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation." *Brief Bioinform*, 9(4): 326–332. URL http://dx.doi.org/10.1093/bib/bbn016.

Wolfe, P. (**1961**). "A Duality Theorem for Nonlinear Programming." *Quarterly of Applied Mathematics*, 19: 239–244.

Wu, M. C. and Lin, X. (**2009**). "Prior biological knowledge-based approaches for the analysis of genome-wide expression profiles using gene sets and pathways." *Statistical Methods in Medical Research*, 18(6): 577–593. URL http://smm.sagepub.com/content/18/6/577.abstract.

Yousef, M., Ketany, M., Manevitz, L., Showe, L., and Showe, M. (**2009**). "Classification and biomarker identification using gene network modules and support vector machines." *BMC Bioinformatics*, 10(1): 337. URL http://dx.doi.org/10.1186/1471-2105-10-337.

Zhu, Y., Shen, X., and Pan, W. (**2009**). "Network-based support vector machine for classification of microarray samples." *BMC Bioinformatics*, 10 Suppl 1: S21. URL http://dx.doi.org/10.1186/1471-2105-10-S1-S21.

Zou, H. and Yuan, M. (**2008**). "THE F_∞-NORM SUPPORT VECTOR MACHINE." *Statistica Sinica*, 18: 379–398.

List of Publications

Bender, C., Fröhlich, H., Johannes, M., Beißbarth, T. (**2008**). "Extending pathways with inferred regulatory interactions from microarray data and protein domain signatures." *Proceedings of CAMDA*

Brase, J.C., Johannes, M., Schlomm, T., Fälth, M., Haese, A., Steuber, T., Beißbarth, T., Kuner, R., Sültmann, H. (**2010**). "Circulating miRNAs are correlated with tumor progression in prostate cancer." *Int J Cancer.* 128(3):608–16. URL http://dx.doi.org/10.1002/ijc.25376

Johannes, M., Brase, J. C., Fröhlich, H., Gade, S., Gehrmann, M., Fälth, M., Sültmann, H., and Beißbarth, T. (**2010**). "Integration Of Pathway Knowledge Into A Reweighted Recursive Feature Elimination Approach For Risk Stratification Of Cancer Patients." *Bioinformatics*, 26(17): 2136–2144. URL http://dx.doi.org/10.1093/bioinformatics/btq345.

Porzelius, C.*, Johannes, M.*, Binder, H., and Beißbarth, T. (**2011**). "Leveraging external knowledge on molecular interactions in classification methods for risk prediction of patients." *Biom J.*, 53(2): 190–201. URL http://dx.doi.org/10.1002/bimj.201000155. *equal contribution

Johannes, M., Fröhlich, H., Sültmann, H., and Beißbarth, T. (**2011**). "`pathClass`: An R-Package for Integration of Pathway Knowledge into Support Vector Machines for Biomarker Discovery." *Bioinformatics*, (submitted).

Kahn, N., Meister, M., Eberhardt, R., Muley, T., Schnabel, P.A., Bender, C., Johannes, M., Keitel, D., Sültmann, H., Herth, F.JF., Kuner, R. (**2011**). "Early Detection of Lung Cancer by Molecular Markers in Endobronchial Epithelial Lining Fluid" *Am. J. Respir. Crit. Care Med.*, (submitted).

Eidesstattliche Erklärung

Die hier vorgelegte Dissertation habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

Heidelberg, den 22. März 2011

Marc Johannes