Graph based fusion of high-dimensional gene- and microRNA expression data

Dissertation

zur Erlangung
des mathematisch-naturwissenschaftlichen Doktorgrades
"Doctor rerum naturalium"
der Georg-August-Universität Göttingen

vorgelegt von
Stephan Gade
aus Jena

Göttingen 2012

Betreuungsausschuss

Prof. Dr. Tim Beißbarth
Prof. Dr. Stephan Waack

Mitglieder der Prüfungskommision

Referent:	Prof. Dr. Tim Beißbarth
Koreferent:	Prof. Dr. Stephan Waack

Weitere Mitglieder der Prüfungskommision

Prof. Dr. Carsten Damm
Prof. Dr. Burkhard Morgenstern
Prof. Dr. Wolfgang May
Prof. Dr. Dieter Hogrefe

Tag der mündlichen Prüfung: 10.12.2012

Abstract

One of the main goals in cancer studies including high-throughput microRNA (miRNA) and mRNA data is to find and assess prognostic signatures capable of predicting clinical outcome. Both mRNA and miRNA expression changes in cancer diseases are described to reflect clinical characteristics like staging and prognosis. Furthermore, miRNA abundance can directly affect target transcripts and translation in tumor cells. Prediction models are trained to identify either mRNA or miRNA signatures for patient stratification. With the increasing number of microarray studies collecting mRNA and miRNA from the same patient cohort there is a need for statistical methods to integrate or fuse both kinds of data into one prediction model in order to find a combined signature that improves the prediction.

Here, we propose a new method to fuse miRNA and mRNA data into one prediction model. Since miRNAs are known regulators of mRNAs, correlations between miRNA and mRNA expression data as well as target prediction information were used to build a bipartite graph representing the relations between miRNAs and mRNAs.

Feature selection is a critical part when fitting prediction models to high-dimensional data. Most methods treat features, in this case genes or miRNAs, as independent, an assumption that does not hold true when dealing with combined gene and miRNA expression data. To improve prediction accuracy, a description of the correlation structure in the data is needed. In this work the bipartite graph was used to guide the feature selection and therewith improve prediction results and find a stable prognostic signature of miRNAs and genes.

The method is evaluated on a prostate cancer data set comprising 98 patient samples with miRNA and mRNA expression data. The biochemical relapse, an important event in prostate cancer treatment, was used as clinical endpoint. Biochemical relapse coins the renewed rise of the blood level of a prostate marker (PSA) after surgical removal of the prostate. The relapse is a hint for metastases and usually the point in clinical practise to decide for further treatment.

A boosting approach was used to predict the biochemical relapse. It could be shown that the bipartite graph in combination with miRNA and mRNA expression data could improve prediction performance. Furthermore the approach improved the stability of the feature selection and therewith yielded more consistent marker sets. Of course, the marker sets produced by this new method contain mRNAs as well as miRNAs.

The new approach was compared to two state-of-the-art methods suited for high-dimensional data and showed better prediction performance in both cases.

Zusammenfassung

Eines der Hauptziele in der modernen Krebsforschung ist es mit Hilfe von Hochdurchsatztechnologien zum Messen von mRNA- und miRNA-Daten, Signaturen zu finden, die es ermöglichen klinische Endpunkte vorherzusagen. Sowohl für mRNA Transkripte wie auch für miRNAs ist gezeigt worden, dass Änderungen im Expressionslevel klinische Parameter wie Tumorstadium oder Prognose widerspiegeln können. miRNAs sind direkte Regulatoren der Genexpression und haben einen unmittelbaren Einfluss auf ihre Zieltranskripte in der Tumorzelle. Oft werden Vorhersagemodelle benutzt, um mRNA- oder miRNA-Signaturen zu finden, mit deren Hilfe Patienten stratifiziert werden können. Mit steigender Anzahl von Studien, die sowohl mRNA- wie auch miRNA-Daten derselben Patienten enthalten, werden Methoden zur Integration beider Datentypen in ein Vorhersagemodell immer wichtiger. Das Ziel hierbei ist eine kombinierte Signatur aus mRNAs und miRNAs zu erhalten und somit die Qualität der Vorhersage zu verbessern.

In der vorliegenden Arbeit stelle ich eine neue Methode vor, die es ermöglicht mRNA- und miRNA-Daten in einem Modell zu integrieren. Da miRNAs mRNAs direkt beeinflussen, wurden Korrelationen zwischen den Expressionsleveln sowie Datenbanken mit vorhergesagten miRNA-mRNA Interaktionen benutzt. Damit wurde ein bipartiter Graph berechnet, der die miRNA-mRNA-Relationen enthält.

Feature Selection ist ein entscheidender Teil bei Modellen für hochdimensionale Daten. Die meisten Methoden gehen von der Annahme aus, dass die einzelnen Features unabhängig voneinander sind. Dies ist eine Annahme, die gerade im Umgang mit miRNA- und mRNA-Daten aufgrund der regulatorischen Eigenschaften der miRNAs falsch ist. Um nun die Vorhersage eines Modells mit beiden Datentypen zu verbessern, bedarf es einer Beschreibung der Korrelationsstruktur in den Daten. In dieser Arbeit wurde der bipartite Graph mit der Schätzung der miRNA-mRNA-Relationen dazu benutzt, die Feature Selection zu steuern und somit die Vorhersageergebnisse zu verbessern und gleichzeitig eine stabile prognostische Signatur aus miRNAs und mRNAs zu erhalten.

Die Methode wurde an einem Prostatakrebs-Datensatz mit miRNA- und mRNA-Expressionsdaten von 98 Patienten getestet. Der klinische Endpunkt, der vorhergesagt werden sollte, war in diesem Fall BCR ("biochemical relapse"), das erneute Ansteigen des PSA-Levels (Prostataspezifisches Antigen) nach dem Entfernen der Prostata. Dieser erneute Anstieg von PSA im Blut ist ein starker Hinweis auf die Bildung eines Tumorrezidives oder einer Metastase und in der klinischen Praxis der Zeitpunkt um eine neue Therapie zu prüfen.

In dieser Arbeit wurde ein Boosting-Ansatz gewählt, um BCR vorherzusagen. Wir konnten zeigen, dass der bipartite Graph in Kombination mit den miRNA-

und mRNA-Expressionsdaten die Vorhersage verbessert. Zusätzlich wurde die Stabilität der Feature Selection verbessert und damit konsistentere Signaturen, bestehend aus miRNAs und mRNAs, produziert.

Dieser neue Ansatz wurde mit zwei modernen, für hochdimensionale Überlebensdaten geeignete Verfahren verglichen. In beiden Fällen schnitt unser Ansatz besser ab.

Dedicated to my father
Dr. Reinhold Gade

Acknowledgements

I have to thank many more people than those who are actually listed on this page. Without these people this thesis would not have been possible.

I owe deepest gratitude to my supervisor Prof. Tim Beissbarth for his support, his engagement, and for sharing his deep knowledge with me. I am obliged to my second supervisor and referee Prof. Stephan Waack for his effort and time.

I owe many thanks to my friends and former colleagues at the DKFZ, especially the leader of the Cancer Genome Research Group Prof. Holger Sültmann. I want to thank Christian Bender, Marc Johannes, Jan C. Brase, Ruprecht Kuner, and Frauke Henjes for many discussions and a great time in Heidelberg. Special thanks go to Maria Fälth for reading my thesis, many valuable tips, and being the last nerdy colleague. Also special thanks go to Daniela Wuttig for reading my thesis and explaining me a lot about prostate cancer.

For his help with boosting and time-to-event data I thank Prof. Harald Binder. And for many fruitful discussions about miRNA-mRNA integration I want to thank Prof. Holger Fröhlich.

Furthermore, I want to thank Tanja Weber for her love and understanding. And finally I want to thank my family. They always supported me and guided my way.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BCR	biochemical relapse, the renewed rise of the blood PSA level after prostatectomy
cDNA	complementary DNA
DNA	deoxyribonucleic acid
FDR	false discovery rate
GO	Gene Ontology
HPRD	Human Protein Reference Database
IPEC	integrated prediction error curve
KEGG	Kyoto Encyclopedia of Genes and Genomes
kNN	K-nearest neighbors
miRNA	microRNA, a class of non-coding RNA
mRNA	messenger RNA
PEC	prediction error curve
PID	Pathway Interaction Database
PPI	protein-protein interaction
PSA	prostate specific antigen, an enzyme secreted by the prostate that is used as diagnostic marker
RFE	recursive feature elimination
RISC	RNA-induced silencing complex, a complex of AGO proteins and the mature miRNA
RNA	ribonucleic acid

RSF ..... Random survival forests

SNP ..... single nucleotide polymorphism, a single base change in the DNA

SOM ..... Self-organizing maps

SVM ..... Support Vector Machine

UTR ..... untranslated region, the region of an mRNA which is not translated into a protein

Chapter 1

Introduction

1.1 Clinical Cancer Research

1.1.1 Tumorgenesis

A metazoan cell, as for instance a human cell, carries the complete genetic information of the whole organism. The genetic code includes all information that is needed to develop and maintain the molecular mechanism for regulating proliferation, differentiation, and at the end of the live cycle of a cell, the controlled dead called *apoptosis*.

Changes in the genomic information are caused either by erroneous replication or external factors like radiation or chemicals and range from single nucleotide changes, called point mutations, to aberrations affecting whole chromosomes. Such changes can cause an abnormal transformation of cells into malignant neoplasms which overcome the normal cell cycle mechanisms and eventually lead to uncontrolled proliferations. The transformation of normal cells into cancer cells is a complex process called *tumorgenesis*. Usually several steps, several hallmarks (Hanahan and Weinberg, 2000, 2011), are needed (figure 1.1) to complete this process.

The realization that a tumor is formed of cells that have lost the normal ability of forming tissue and have gained the abnormal ability of immortal replication was one of the most important steps in the beginning of the biomedical cancer research.

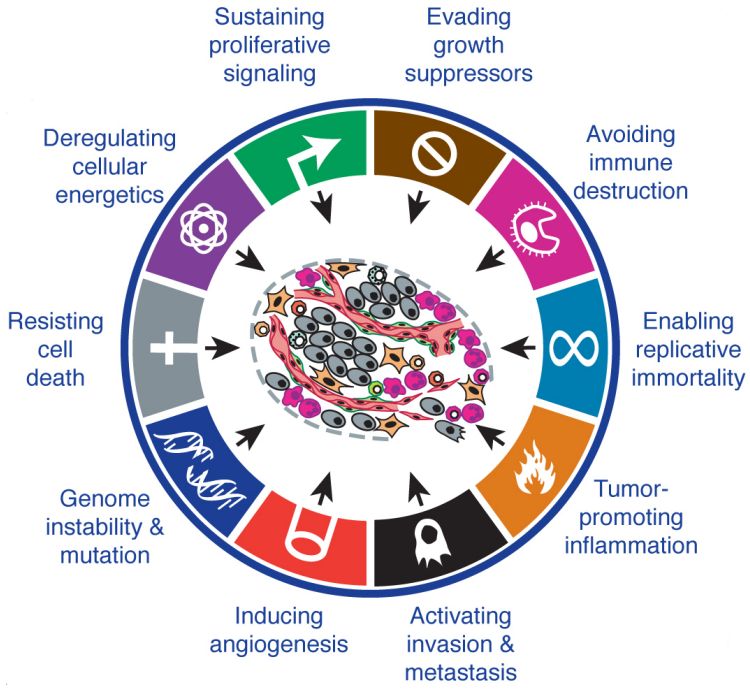


FIGURE 1.1. *The hallmarks of Cancer (Hanahan and Weinberg, 2011, courtesy of Elsevier).*

In principal every tumor can be traced back to the tissue the first tumor cells originated from. The majority of tumors grow locally within this tissue. These tumors are called *benign*. They are considered harmless for the patient. Other tumors are able to invade adjacent tissue and release cells into the blood stream spawning so called *metastases* in other organs. These metastases cause around 90% of cancer related deaths (Weinberg, 2007). These tumors are called *malignant*.

Finer categories can be made based on the original tissue. Most malignant tumors, so called *carcinomas*, arise from epithelial cells. In healthy tissue, these cells form a layer of tissue lining cavities and channels or protect organs. Epithelial tissue fulfills many important tasks in the human body ranging from protection of organs to secretion. Tumors arising from epithelial tissue can be distinguished based on these two major biological functions. Squamous cell carcinomas arise from epithelial cells serving as protecting cell sheets whereas adenocarcinomas come from secreting epithelial cells. Examples of both types can be seen in figure 1.2. Carcinomas are responsible for around 80% of cancer related deaths.

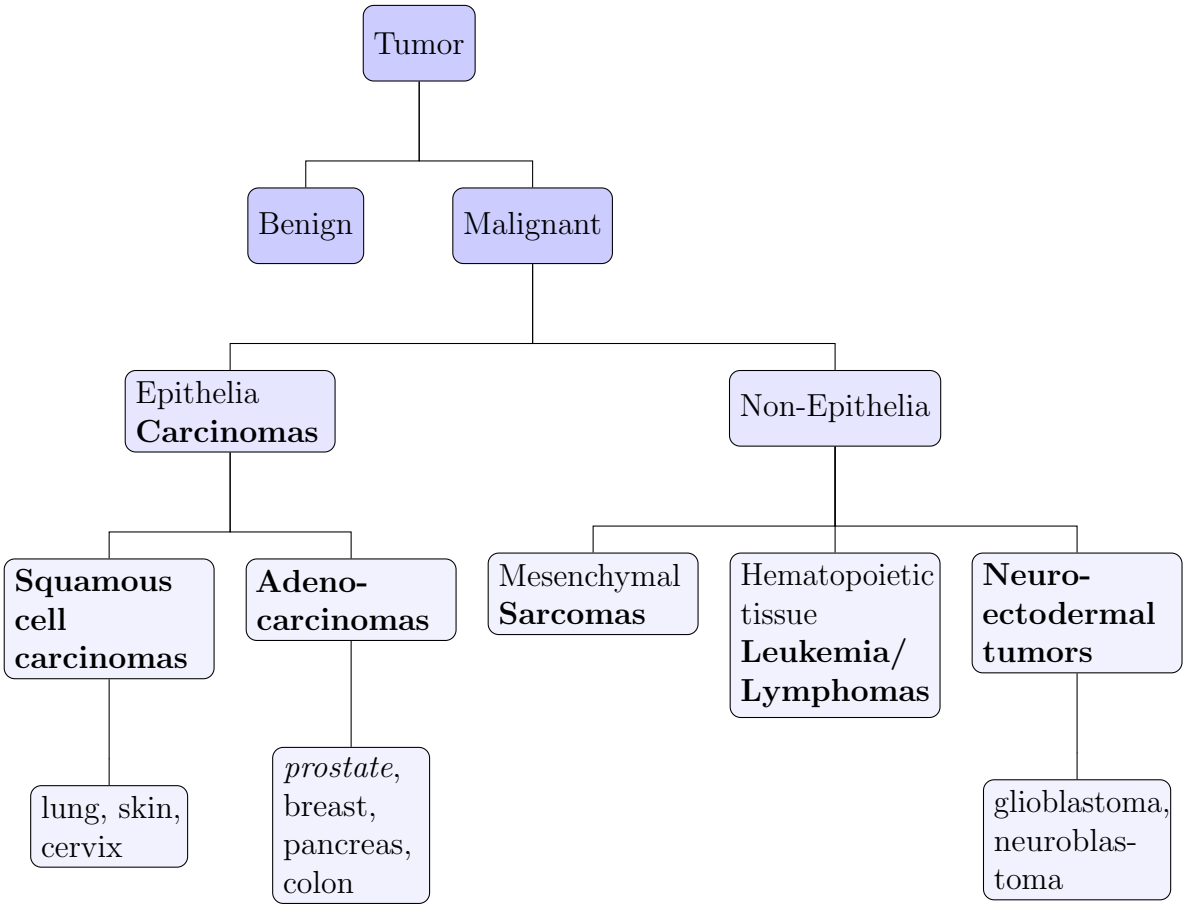


FIGURE 1.2. *A classification of cancer types based on Weinberg (2007)*

1.1.2 Prostate Cancer

The prostate is a secreting organ with a central role in the reproduction mechanism of men. Although there is still a debate about the true cellular origin (Choi et al., 2012; Goldstein et al., 2010; Wang et al., 2009), prostate cancer belongs to the class of adenocarcinomas and is assumed to arise from secreting epithelial tissue in the prostate.

Prostate cancer is one of the most frequent tumors in men and the third leading cause of death in the western hemisphere (Jemal et al., 2011). Prostate cancer patients are 65 years old on average when diagnosed with prostate cancer. Routinely several biopsies are taken to support the diagnosis.

The standard therapy for nearly all cases is (at least in Germany) the radical prostatectomy that means the complete removal of the prostate accompanied by heavy side effects. In case of a metastatic relapse additional therapies like radiotherapy and hormone therapy are used. However, nearly all patients with advanced prostate cancer eventually progress to a metastatic disease state that shows resistance to hormone therapy (Felici et al., 2012). This state has been termed castration-resistant prostate cancer. At this stage the final treatment option is chemotherapy yielding an average life expectancy of 16-18 months (Tannock et al., 2004).

In prostate cancer two risk groups can be distinguished. Around 20-30 % of the patients have an aggressive tumor with a high risk of metastatic relapse and a high mortality rate (Bill-Axelson et al., 2008). The remaining 70-80 % have a non-aggressive tumor. Considering the average age of the patients this group is over-treated with a diminished quality of life. For these patients a more conservative approach like active surveillance could be deployed.

Although, there are standard diagnostic tests indicating prostate cancer there is no established test available that is suitable to distinguish the two risk groups. One of the goals of modern clinical prostate cancer research is to identify such *prognostic markers*.

1.1.3 Biomarkers and Genomic Biomarkers

Nowadays the term *biomarker* is widely used in different terms and context. A formal definition was given by the Biomarkers Definitions Working Group

> A characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention.
>
> (Biomarkers Definitions Working Group, 2001)

In the biomedical research the term biomarker in most cases refers to *genomic biomarkers* that are markers associated to the genomic profile of a patient. Note, that the term genomic does not necessarily coins a DNA marker. It can also describe a marker on the level of RNA, e.g. mRNA and miRNA, and also on protein level.

In the past mRNA sets of markers, so called marker panels or signatures, have been described for several cancer entities. The most well known examples are several prognostic signatures described for breast cancer (Paik et al., 2004; van 't Veer et al., 2002; Wang et al., 2005). Based on these signatures multigene test like MammaPrint and Oncotype DX have found their way into clinical practise. However, despite these efforts in translational research the clinical utility of genomic signatures is still under debate (Sotiriou and Piccart, 2007).

Unfortunately, for prostate cancer a reliable risk prognosis is still a challenge and no marker or marker signature is used in clinics for this purpose (Tosoian and Loeb, 2010). However, a diagnostic marker has been used for several years: the prostate specific antigen (PSA). This is a protein secreted by the prostate and a major protein in the seminal fluid (Balk et al., 2003). Since PSA is also expressed in prostate cancer cells and it can enter the blood stream, the blood PSA level was found to be a first indicator of prostate cancer (Tosoian and Loeb, 2010).

After the removal of the prostate the blood PSA level goes down and is monitored during the follow-up time. The renewed rise of the PSA level is called biochemical relapse (BCR). It is an indicator for a local relapse or metastasis and in clinical practise the point to decide for further treatment.

1.1.3.1 The Basic Principle of Gene Expression

All gene signatures mentioned above are mRNA signatures. That means that the test measures the mRNA level of a certain gene either in the tissue, e.g.

tissue from a biopsy, or in the blood. Other types of RNA molecules have been described to be potential biomarkers in the last years. These RNA molecules belong to the class of non-coding RNAs. That means they do not code for a protein but fulfill other tasks in the cell, e.g. postranscriptional regulation.

One of the fundamental dogmas in modern cell biology describes the sequence from the genetic information contained within the DNA to the final product which is in most cases a protein (see figure 1.3). Proteins are the main effectors in the cell fulfilling a variety of tasks as e.g. structural proteins or enzymes. Especially enzymes, biocatalysts of the cell, play a central role in the lifespan of a single cell not only in catalysing metabolic reactions but building complex signal cascades used to transport external signals from the cell membrane to the cellular nucleus (a process that is called signal transduction).

Figure 1.3 shows a basic scheme of the single steps involved in the complex process of gene expression (see Strachan and Read, 2005 and Voet and Voet, 2004 for a detailed description of the expression of the genetic information in the cell) . Every step in this process underlies strict regulations.

The first step is *transcription*. The *gene* is transcribed to a one-stranded RNA molecule, the *pre-mRNA* (pre messenger RNA). The term gene coins a genomic sequence (DNA or RNA) that directly encodes a functional product, i.e. a protein or a non-coding RNA (Gerstein et al., 2007). The transcription is regulated by various mechanisms. Transcription factors are special proteins activating or repressing the transcription of their target genes. Transcription factors themselves are regulated by a complex network of signal pathways allowing the cell to dynamically change its gene expression profile to react to changing environmental conditions.

The resulting the pre-mRNA is processed further. In this *splicing* step introns, which are not part of the final protein sequence, are removed. By removing also part of the protein coding sequence, the so called *exons*, the cell can use one pre-mRNA as template to produce different proteins. This process is called alternative splicing. Several studies linked this process to various cancer types (Germann et al., 2012; Rajan et al., 2009; Venables, 2004). The splicing step results in the final mRNA.

All these steps happen in the nucleus of the cell. Afterwards the mRNA is transported through the membrane of the nucleus to the cytoplasm. Here, the

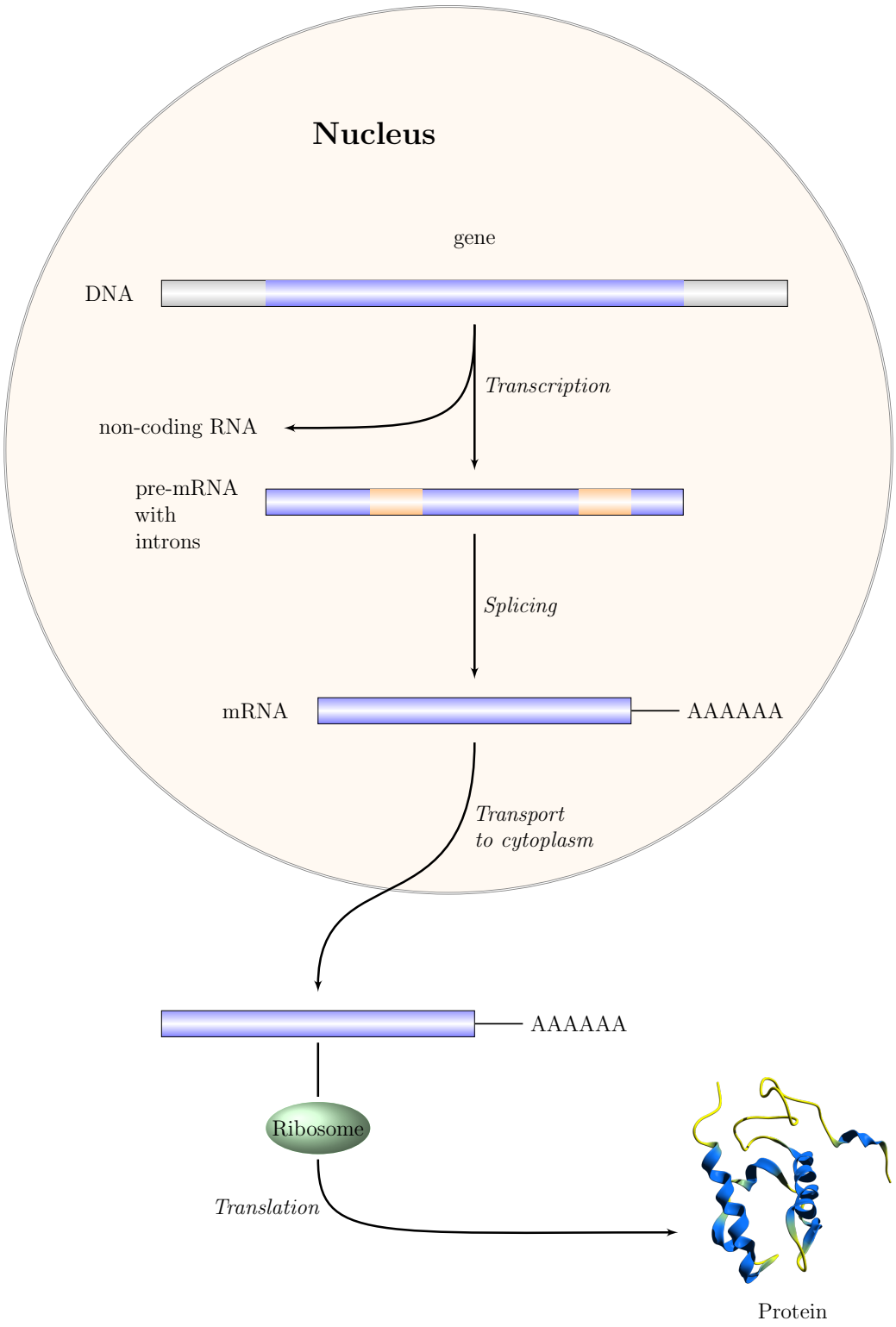


FIGURE 1.3. *The basic principle of gene expression. As the first step the part of the DNA coding strand known as gene is* transcribed *to pre-mRNA. In the second step the introns and are* spliced *out forming the mature mRNA. After the transport from the nucleus to the cytoplasm the protein is assembled from this mRNA in a process called* translation *(derived from Strachan and Read, 2005).*

mRNA is translated by ribosomes yielding the primary amino acid sequence of the protein. To protect the mRNA against degradation in the cytoplasm, to regulate nuclear export, and to allow the translation process to start, a poly-Adenyl tail (poly-A tail) is attached to its 3' and a 5' cap to its 5' end[(1)]. There is a delicate balance between the rate an mRNA is transcribed and its decay rate in the cytoplasm. Several factors can influence the stability of the mRNA and by this regulate the amount of protein. These factors include for examples enzymes responsible for removing the poly-A tail (specialized exonucleases) of the mRNA making it vulnerable to degradation. More intrinsic factors are microRNAs (miRNAs).

1.1.3.2 MicroRNAs - Essential Regulators of Gene Expression

miRNAs are short (around 22 nucleotides long), single stranded RNA molecules. They bind sequence mediated to the 3' end of a target mRNA (Bartel, 2009; Bartel and Chen, 2004). Around 30% of the human protein coding genes underlie regulation of miRNAs (Lewis et al., 2005). Around 2,000 human miRNAs are known so far[(2)] and, similar to mRNAs, miRNAs can be measured in a genome-wide manner.

In animals, binding of a miRNA to its target mRNA does not need to be perfect. A match in the seed region of the miRNA (nucleotide 1 to 8), however, seems to be important (Filipowicz et al., 2008). The binding leads to a translational block either by degradation of the target mRNA, headed by a decapping/deadenylation of the mRNA , or by inhibiting the binding of the ribosomes and, consequently, inhibiting of protein biosynthesis. Cleavage of the target mRNA or destabilization and subsequent degradation influence the abundance of the mRNA levels which is measurable with RNA screening methods (Giraldez et al., 2006; Wu et al., 2006). In any case, the miRNA represses the translation of their target mRNAs into proteins. miRNAs are therefore *negative regulators of gene expression.*

[(1)] The notation 3' and 5' for single DNA strands and RNA molecules is based on the free carbon atoms of the desoxyribose or ribose, the sugar that is the basis of DNA and RNA. This notation allows to assign a direction to DNA and RNA molecules. For example, during the transcription the pre-mRNA is built up from 5' to 3'.

[(2)] The miRBase database (www.mirbase.org, release 19, last checked August 15th, 2012) list 1600 precursors and 2042 mature miRNAs.

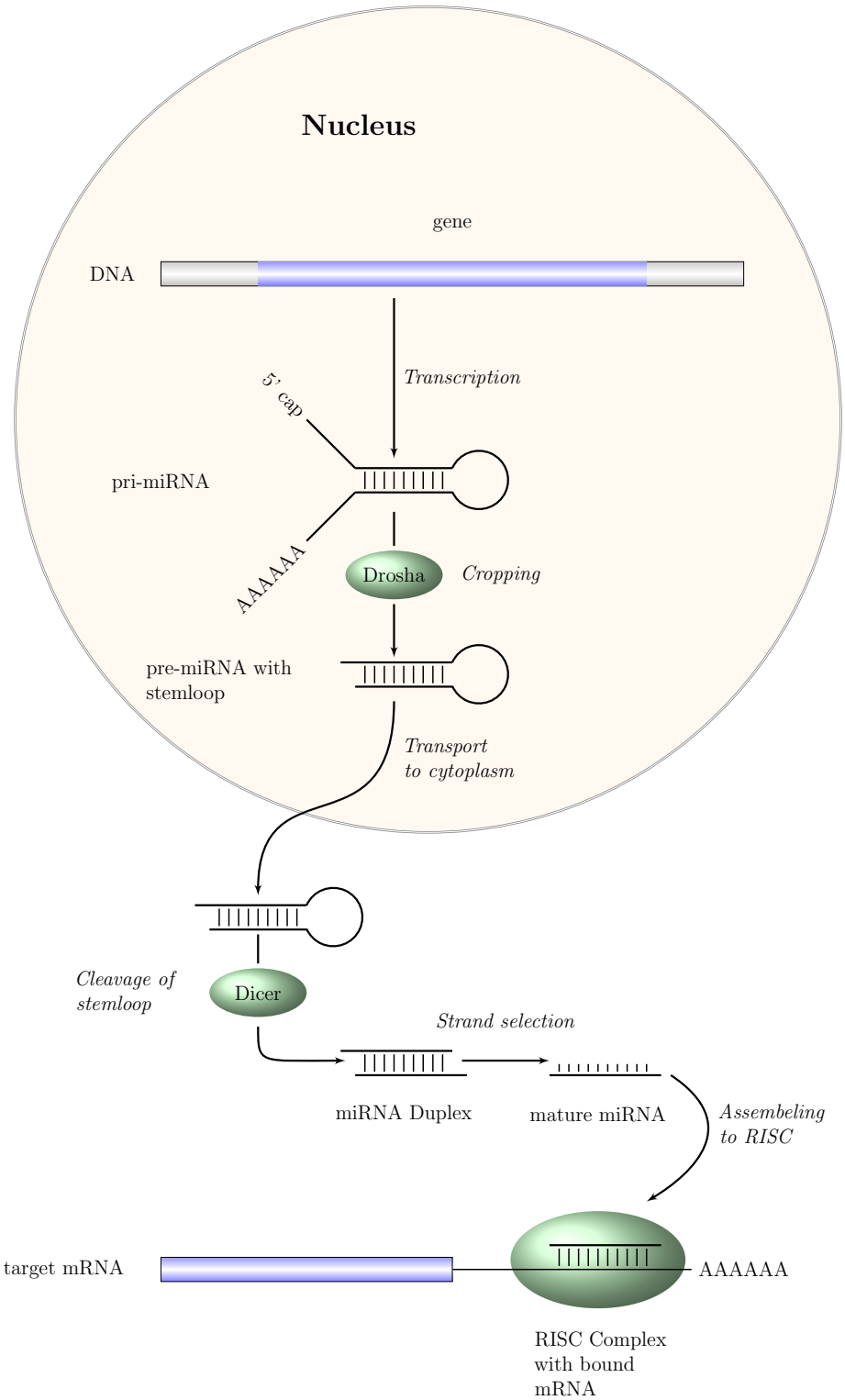


FIGURE 1.4. *The biogenesis of a miRNA beginning with the miRNA gene that is transcribed to the pri-miRNA (for simplicity only one precursor is shown in the primary transcript). Processing via Drosha and Dicer yields the mature miRNA that is incorporated into the RISC complex and finally binds to the target mRNA (Filipowicz et al., 2008; Kim et al., 2009).*

A miRNA can be encoded by a separate gene or be a part of the introns of protein coding host genes. Figure 1.4 shows the basic principle of the miRNA biogenesis from a miRNA gene (cf. Filipowicz et al., 2008; Kim et al., 2009 for more details). Transcription of the miRNA coding region leads to the pri-mRNA, the primary transcript that is usually several kilobases long and can contains several precursors. The miRNA precursors are stem-loop structures that are cleaved out by an enzyme called Drosha. The resulting pre-miRNA is a double stranded small RNA with the characteristic stem-loop. This double stranded miRNA precursor is transported from the nucleus where the transcription and cleavage takes place into the cytoplasm where the miRNA will accomplish its primary task. To do this one final step is needed. A protein called Dicer cleaves the stem-loop. The resulting duplex unwinds yielding the mature miRNA and its passenger strand. The thermal stability of both strands determines which strand is incorporated as mature miRNA into the RNA-induced silencing complex (RISC) that eventually binds to the target mRNA. The other strand is degraded. Strand selection, however, is not a stringent process and for some precursors both strands occur in the cell as mature miRNAs (Kim et al., 2009).

Similar to mRNA, miRNA transcription and processing underlie a complex regulation. Disturbance of this regulation can have a large effect since it directly affects the target genes of this miRNA. It is therefore not surprising that deregulation of miRNAs has been linked to development and progression of several diseases including cancer (Brase et al., 2011; Groce, 2009; Lu et al., 2005).

Since miRNAs are rather small and the sequence complementary to the target mRNA does not need to be perfect, one miRNA can have several (up to several hundreds) targets. Besides the pure sequence complementary the thermal stability of the miRNA-mRNA complex is an important factor. Since the experimental validation of a miRNA-mRNA pair is an elaborate issue miRNA target prediction algorithms try to find novel miRNA targets among known genes. Several different target algorithms exists taking into account not only sequence information but also theoretical thermal stability and information about homologue binding sites of other species[(3)].

[(3)] Since miRNA binding sites are an important aspect of gene expression regulation, they are evolutionary highly conserved.

With the miRBase database a central repository for miRNA related information has been created (Griffiths-Jones et al., 2008). Besides sequence information of mature miRNA as well as of miRNA precursors, miRBase describes the naming conventions of miRNAs (Ambros et al., 2003). A miRNA name consist of the species identifier (e.g. hsa for human miRNAs) followed by "mir" for miRNA genes or "miR" in case the mature miNRA is described. The single miRNA is identified by a unique number. The mechanism behind the strand selection of the double stranded precursor is not yet fully understood. If both strands of one precursor occur in the cell as mature miRNAs, the unique number is followed by either a "3p" or "5p" indicating the strand. An example of a mature miRNA name would be "hsa-miR-375-5p".

1.1.3.3 Other Types of Biomarkers

Besides RNA marker like mRNA and miRNAs other types of genomic markers are available and in standard practise in the biomedical research. DNA based markers comprise e.g. single nucleotide polymorphism (SNP) or point mutations as well as large chromosomal aberrations like deletions, amplifications, and fusion genes (Chung and Chanock, 2011). There are epigenetic markers like changes in the methylation profile of the DNA or histones[4] (cf. Mikeska et al., 2012 for an overview).

Besides these traditional genomic markers, genetic activity or diregulation can be measured directly on the protein level. This can be accomplished either in large scale for many proteins at the same time by e.g. mass spectrometry or protein arrays. Another way, and more simple, are measurements via immunochemistry for single markers. A well known example is here the ERBB2 receptor which is measured in standard clinical practise for breast cancer patients (Penault-Llorca et al., 2009).

Finally, specific metabolites, e.g lipids in the blood, can also be used as biomarkers. For example it is known that a tumor changes the metabolic profile of its cells during development to cope with its rapid growing energy requirements. In case of an undersupply with oxygen the switch to anaerobic

[4] Methylation denotes the attachment of a methyl group ($-CH_3$) to a cytosine in DNA or to an arginine or lysine amino acid in histones. Methylation of DNA as well as methylation of histones has a crucial influence on transcriptional activity and is therewith a very important factor in gene expression regulation.

metabolic processes is a logical consequence. These changes can be measured by certain metabolites (see i.e. Chajès et al., 2011).

1.1.4 Microarray Technology

miRNAs as well as mRNAs can be measured genome-wide that means all known miRNAs or mRNAs can be measured simultaneously. In the past twenty years *microarrays* (Schena et al., 1995) have become the defacto standard for large scale biomarker measurements. Besides genome-wide microarrays there are also specialized custom microarrays designed to measure a well defined set of markers.

Thereby, the basic working principle is rather simple. Genomic probes (approximately 30 up to 150 nucleotides long) are attached to a solid slide. The probes are packed at high density. Every probe has a specific sequence and is used to detect a specific mRNA or DNA part.

Since the probes can be designed to match any given sequence , microarrays can cover almost all types of genomic biomarker. SNP and tiling arrays cover DNA based markers. They are used to measure SNPs and genomic aberrations (insertions, deletions, and amplifications of specific chromosomal regions).

However, by far the most often used microarrays are microarrays for RNA quantification especially gene expression microarrays. Basically two types of gene expression microarrays can be distinguished.

cDNA- (complementary DNA[5]) or two-color arrays (Duggan et al., 1999; Schena, 1999) were mostly used in the beginning of the microarray era. The probes (cDNA, hence the name) were spotted to a solid glass slide. The mRNA of two distinct samples was labeled with two different dyes and afterwards hybridized to the array in a competitive manner. Afterwards the fluorescent intensities are scanned in two channels, one for each dye. Based on the intensities conclusion could be drawn which sample contained more or less of a specific mRNA.

[5] Complementary DNA or short cDNA denotes single stranded DNA that is gained from mRNA via a process called reverse transcription. As the name suggests it is simply the inversion of transcription: from mRNA the complementary DNA is constructed. This is catalyzed by an enzyme called reverse transcriptase that can be found in various RNA viruses.

Nowadays these kind of microarrays are hardly used anymore. The more precise one-color arrays have been established allowing a higher density of probes (and hence a larger number of mRNAs measurable at once) and more stable measurements. In order to allow density the probes are not spotted but shorter oligos are synthesized directly at the slide (Lipshutz et al., 1999) or are attached to silica beads assembled in microwells (Gunderson et al., 2004; Walt, 2000). While for two-color arrays it was necessary to hybridize the control at the same slide to eliminate slide effects the high reproducibility of modern microarrays make it possible to hybridize each sample (including possible controls) to an independent slide.

The principle of a one-color microarray experiment is illustrated in figure 1.5. Starting with several tissue samples, usually from a condition of interest and a reference (a typical example is a comparison of tumor against normal tissue), the mRNA of these samples is extracted and purified (and in most cases amplified to get more starting material). In a first step this mRNA is reversely transcribed to cDNA (complementary DNA) and at the same time labeled with biotin.

The biotin labeled cDNA is than hybridized to the array. The probes attached to the arrays bind to the cDNA matching their sequence. One spot on the array contains several probes with the identical sequence. The higher a gene is expressed the more mRNA and eventually the more cDNA is contained in the sample, and consequently, the more of the corresponding probes are occupied with cDNA molecules.

After scanning the array the accumulation of biotin labeled molecules cause a bright spot at the image where the cDNA has bound to the array. The signal intensity is then a measure for the gene expression. The higher the intensity of the spot the higher the expression of the corresponding gene[(6)].

After scanning the array and transforming the image to signal intensity values there are several pre-processing steps (cf. Stekel, 2003; Wit and McClure, 2004 for and overview on microarray analysis). Modern microarrays are designed with a certain degree of redundancy. Since the probes are rather small compared to an mRNA it is possible to design several different probes targeting the same

[(6)] Of course, other factors like the RNA sequence and hybridization efficacy can also influence in intensity of the spot.

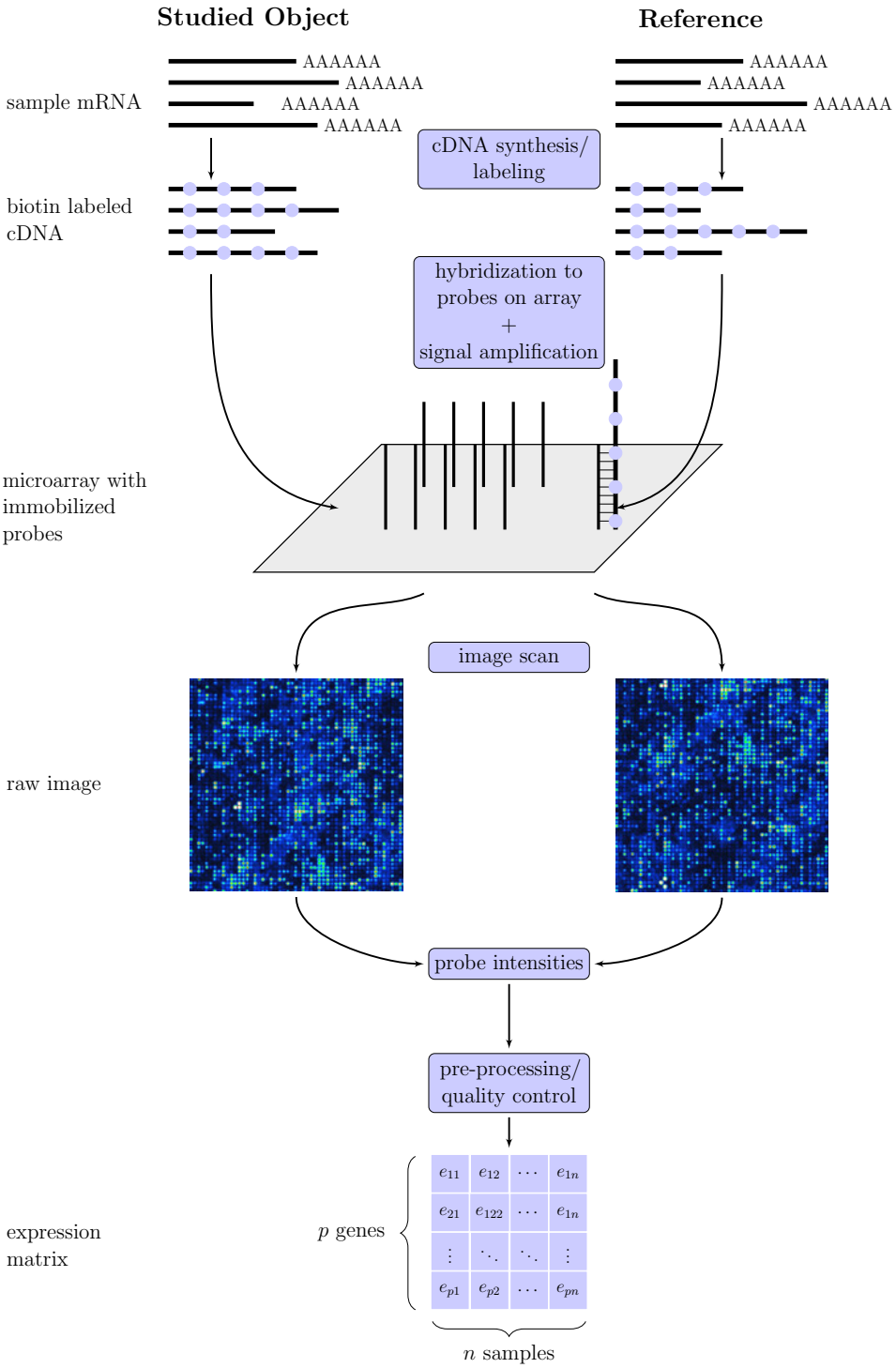


FIGURE 1.5. *The figure shows the basic workflow of a microarray experiment. Shown is a one-color mRNA (gene expression) microarray.*

mRNA. The combination of the signal intensities of all these several probes to a so called *expression value* of the gene is one of these pre-processing steps. Other steps include background correction and normalization steps. Background correction procedures are used to eliminate possible unspecific background signals caused by e.g. reflections on the slide. Normalization steps include in-array and between-array normalization. In-array normalization should remove spatial effects on the array e.g. caused by a distinct dispersion of the sample on the slide. Between-array normalization is used to eliminate technical variance between the samples (e.g. slight differences in the purification or labeling process) and biological variance (e.g. general higher mRNA level in one sample).

After preprocessing the normalized expression values can be displayed in a so called gene expression matrix which is the starting point of the actual analysis and statistical inference. The rows of the gene expression matrix correspond to the genes, the columns to the samples[7]. Similar to the statistical notation the number of genes is denoted with p and the number of samples with n. The expression matrix is therefore a $p \times n$ matrix. It is common to use the log_2 transformed expression values for further analysis due variance stabilization properties of this transformation and an improved visualization of the transformed expression values.

The described experimental workflow is explained using the example of gene expression microarrays. However, the same principle holds true for microarrays for miRNAs and SNP arrays.

[7] Since in statistical terms the genes are the variables (the expression value of a gene would be the value of that varible) and the samples are the observations, this is contradictory to the traditional statistical notation where the variables are usually the columns and the observations the rows.

1.2 Machine Learning Approaches in Bioinformatics

1.2.1 Methods

In the last years the price for a microarray experiment has dropped constantly allowing a large number of experiments which give rise to a vast amount of gene expression data especially in the cancer research. Besides data storing and sharing, e.g. standards for describing a microarray experiment, as well as the afore mentioned pre-processing steps, microarray bioinformatics is especially concerned with the analysis of the resulting gene expression data.

Assuming an expression matrix as introduced in the previous section, several questions arise naturally from such kind of experiment. Usually, several microarray experiments are conducted comparing two groups (e.g. samples from tumor tissue and as controls samples from normal tissue). When the samples of one group are considered biological replicates testing for differences between the two groups breaks down to testing for a difference between the two distributions the single experiments were sampled from. A first question is of course which genes show different expression values between the two groups. Another question that arises is how well these two groups can be separated based on the gene expression measurements.

Of course the outcome does not have to be binary. A continuous endpoint is possible and in real world problems this is often the case, e.g. certain clinical parameters of a patient can be measured on a continuous scale. If the samples were gained from patients for whom the time to a certain event was monitored, the outcome is a time-to-event endpoint. Despite the nature of the endpoint, the underlying question remains the same in all these cases: How well can the outcome be explained by the expression measurements ?

While the classical statistic knows methods to tackle all these different scenarios there is a crucial difference to problems arising there. Microarray data are high dimensional that means the number of genes (or markers in general) is usually much higher than the number of samples and thus $p \gg n$.

Many bioinformatics methods have their origins in machine learning and pattern recognition. According to a common terminology they can be divided into *supervised* and *unsupervised* learning methods. Supervised denotes algo-

rithms where the outcome, i.e. the class labels for a classification problem, is known. The goal is now to learn the underlying rule (or function) connecting the features, in this case the biomarkers, and the outcome based on the training data set. For samples with unknown outcome the learned rule can be used for prediction. Well known examples for supervised learning algorithms are Support Vector Machines (SVM, Boser et al., 1992; Schölkopf and Smola, 2002; Vapnik, 1999), boosting (Freund and Schapire, 1996), the nearest shrunken centroids classifier (Tibshirani et al., 2002), K-nearest neighbors (kNN, Cover and Hart, 1967; Fix and Hodges, 1951), and Random Forests (Breiman, 2001). Other methods are originated in classical regression models. Prominent examples are Lasso (Tibshirani, 1996) and the elastic net (Zou and Hastie, 2005).

If no outcome is known, no class label or continuous score, the only information left are similarities between the samples. In the case of a gene expression matrix this is the similarity between the expression profiles. Unsupervised learning methods, also known as cluster methods, try to discover these similarities. Based on such patterns the samples can be grouped, i.e. in order to define new subclasses. Especially for cancers where no molecular subclasses are known *a priori* this is a valuable approach. Examples for clustering methods are K-means (Lloyd, 1982; MacQueen, 1967) , Self-organizing maps (SOM, Kohonen, 1982), and Neural Gas (Cottrell et al., 2006; Martinetz et al., 1993).

1.2.2 Feature Selection and the Curse of Dimensionality

In the classical statistics a simple linear regression model can be formulated (in matrix notation) as

$$\mathbf{y} = \boldsymbol{\beta} X + \varepsilon$$

where y is the n-dimensional outcome vector, X the $n \times p$ matrix of predictors, and $\varepsilon \sim N(0, \sigma^2)$ is the normally distributed error term. The famous *least squares* solution for this problem was developed by Gauss and Legendre in the early years of the 19th century and is nowadays the standard method to solve linear models

$$\hat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T \mathbf{y}$$

It requires to be $n > p$ in order to inverse $\left(X^T X\right)$. As mentioned before the strength of array based analysis methods in the biomedical field, like

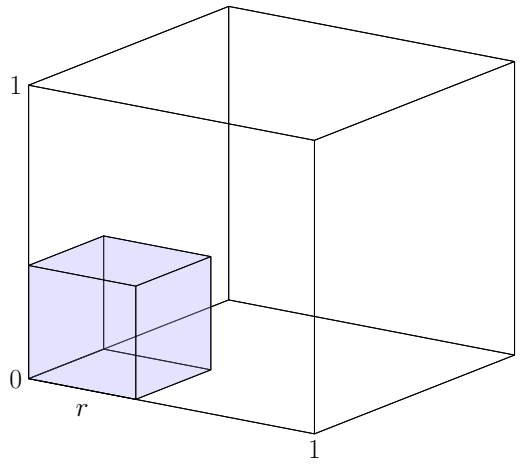


FIGURE 1.6. *Illustration of the curse of dimensionality. The large unit hypercube is the feature space. The colorized cube is the space of the training samples covering a fraction r of the range of every predictor. The fraction of the feature space (fraction of volume of the unit hypercube) and hence the predictive power of a fitted model decreases with increasing dimension p (adapted from Hastie et al., 2009).*

gene expression arrays, is to measure several ten thousands up to hundreds of thousands markers at once. In this high dimensional case usually $p \gg n$.

While methods like SVM and kNN are in principle capable of fitting a model for high dimensional data, their performance that means the prediction power on new data is usually unsatisfactory. The underlying phenomenon is sometimes referred to as *curse of dimensionality*, a term coined by Bellman (1961).

The curse of dimensionality has many facets. In bioinformatics, where models are fitted to high dimensional data, it usually manifests as a sampling problem (cf. Hastie et al., 2009; chap. 2 for details). The quality of a high dimensional model depends on how well the training data cover the feature space, that means how well is the sampling. In case of $p \gg n$ the data are sparse that means the sampling is bad and the underlying structure cannot be covered by the fitted model. Consequently, the predictive power of such a model is poor.

The situation is best explained by a p-dimensional unit hypercube (figure 1.6). The unit hypercube represents the feature space that means the space in which the fitted model will be used for prediction. It is the space the model must be valid in. The colored hypercube marks the subspace that is sampled by the training data. The fraction of the feature space covered by

the training data is r^p (note that $r \leq 1$). Hence, r has to grow exponentially with the dimensions p to cover the same fraction of the feature space. Since r corresponds to the number of training samples, n has to grow exponentially with increasing p. If the number of training samples n is fixed, the fraction of the feature space covered by the model and therewith its predictive power decrease with increasing p.

While the data are sparse in high dimensions traditionally distance metrics like the euclidean distance become useless (Friedman, 1997) and methods relying on them fell apart. Another problem in high dimensional settings is that most of the predictors have no effect on the outcome. Adding only noise to the model these features can mask the underlying relationship of informative features to the outcome.

In modern algorithms the curse of dimensionality is tackled by *feature selection* that means the selection of informative predictors (cf. section 2.2.2.1 for more details) for a specified outcome. By removing unnecessary features the curse of the dimensionality is avoided during model fitting. Of course finding informative features on the same data used for model fitting is not trivial and bears the risk of overfitting. In this case the performance of the training data would be overoptimistic while the performance on unseen data is poor. Feature selection can be a separate step or a part of the learning algorithm (cf. Guyon and Elisseeff, 2003 for an overview on feature selection methods) but most methods assume the predictors to be independent. While this might be true for other research areas it is definitely not in biomedical research.

Genetic regulation forms a complex network that leads to complicated correlation structures. The situation is even worse when using gene expression together with miRNA expression data. One miRNA can target many genes and one gene can be targeted by several miRNAs. This forms a correlation structure even more complex than for gene expression data alone. Feature selection algorithms relying merely on scoring of single features for their importance to the outcome, i.e. the disease state, produce models with probably to many, but highly correlated features. The coherent redundancy in these features causes a decreased performance on new data (Lee et al., 2008). For gene expression data this results in signatures which have a poor overlap between different studies

even if the considered outcome is identical (Michiels et al., 2011; Sotiriou and Piccart, 2007).

1.2.3 Pathway Based Approaches

In the last years several methods have been developed to overcome these shortcomings, at least when dealing with gene expression data. The key idea is to include *prior* biological knowledge of regulation structures in order to resolve co-linearity between the features. For protein coding genes there are several databases covering information about interactions and common *pathway memberships*. A *pathway* is an abstraction made in systems biology. It is thereby defined as a biological network, a set of interactions or functional relationships between molecular entities, i.e. genes or proteins of the cell (Cary et al., 2005). Genes involved in the same pathway, if not having a direct interaction, at least contribute to the same cellular process. Therefore, the assumption that these genes are co-regulated is reasonable.

A variety of databases cover biological pathways or gene and protein interactions (cf. Cary et al., 2005 for an overview). One of the most famous among them is the KEGG database (Kyoto Encyclopedia of Genes and Genomes, Kanehisa et al., 2004) that maps genes to manually curated pathway maps, focusing on molecular interactions of genes in signalling and metabolic networks. A similar approach is followd by PID (Pathway Interaction Database, Schaefer et al., 2009). It is also a manually curated repository but focused on genes with a role in signalling pathways, mostly cancer related. Besides ongoing efforts in the field there are still no consistent standards to report newly found interactions in the biomedical literature. Therefore, Transpath (Choi et al., 2004), a commercial database, contains manually curated interactions from peer-reviewed literature.

The HPRD database (Human Protein Reference Database, Keshava Prasad et al., 2009) comprises information about protein-protein interactions (PPI data) gained from yeast two-hybrid screens. Another database worth mentioning in this context is the MINT database (Licata et al., 2012), also focussing on experimentally verified protein interaction data.

The ConsensusPathDB (Kamburov et al., 2011, 2009) differs from the afore mentioned databases as it is a meta-database. It integrates different pathway and PPI databases, i.e. KEGG, MINT, HPRD, PID, INAct, and others, to draw a more complete picture of regulatory mechanisms in the cell.

Besides these general interaction databases there are databases focussing on special interactions, most notable are transcription factor bindings. As outlined in the former section transcriptions factors are proteins binding to the DNA and therewith promoting or inhibiting the transcription of the target gene. Transcription factor binding sites are key elements in the understanding of transcriptional regulations and hence, databases like Transfac (Matys et al., 2006) and JASPAR (Portales-Casamar et al., 2010) deal with this kind of regulatory interactions.

Besides the databases several efforts have been made to develop formats for storing and sharing pathway information, for example the BioPax language (Biological Pathway Exchange, Demir et al., 2010).

Another structured knowledge resource for gene functions and products is the Gene Ontology (GO, The Gene Ontology Consortium et al., 2000). In a less technical sense the term *ontology* is used for an area of formalized knowledge. An ontology defines items from a specific domain and relationships connecting these items in a structured and hierarchical manner (Bard and Rhee, 2004). In case of the Gene Ontology three domains are considered: biological processes, molecular functions, and cellular components. Biological processes is the domain that can be most likely compared to pathway information contained in databases like KEGG or PID. The hierarchical structure comprises broad terms, i.e. *cell cycle*, on top to more refined terms at the bottom, all of them describing biological processes. A gene (or more precisely a gene product) can be assigned to several of these GO Terms. Since the structure follows a hierarchical order a gene can always be assigned to the parent terms of an assigned term, too[(8)]. Consequently, more general terms on top of the hierarchy contain more genes (that means more genes are assigned to that term) than more specialised terms at the bottom.

[(8)] Note, since a term in the GO can have several parents, GO is not a tree but a directed acyclic graph. Also note, it is sufficient to state the most explicit term valid for a certain gene. The parents terms are included implicitly.

Different methods have been developed to check for overrepresented GO terms in a list of genes, i.e. genes that are differentially regulated between two conditions (Beissbarth, 2006; Beissbarth and Speed, 2004). These GO terms give a hint on altered processes in the cell caused by a deregulation of these genes. Also, GO terms can be used, such as biological pathways, to conclude similar functions and expression patterns. Genes assigned to similar GO terms are likely to contribute to similar processes in the cell.

In recent years, an increasing number of methods incorporated *prior* biological knowledge in model building to overcome the afore mentioned flaws for high-dimensional gene expression data and retrieve stable and highly predictive gene signatures (cf. Porzelius et al., 2011a for an overview).

There are methods incorporating pathway knowledge in a test based setting, i.e. examine each gene separately to retrieve candidate genes for a signature (Wu and Lin, 2009). Of course, more elegant and more useful in the field of biomarker discovery are methods that integrate the biological knowledge in the model fitting process and feature selection. In the following a few examples are mentioned.

Wei and Li (2007) proposed NPR (nonparametric pathway-based regression) models with an additive pathway effect. The pathway effect is estimated by the expression measurements of genes in the particular pathway via regression trees. Li and Li (2008) and Pan et al. (2010) deployed shrinkage regression methods with an altered penalty term to incorporate pathway knowledge. Both methods rely on gene interaction networks as delivered by KEGG or HPRD. In a similar fashion Binder and Schumacher (2009) used boosting to fit an additive model using a penalized likelihood. By adapting the penalization structure gene interaction graphs can be incorporated (cf. section 2.2.3 for details).

Other methods rely on SVMs and are specifically designed for classification tasks (binary endpoints). Zhu et al. (2009) proposed a network based SVM with a penalty constructed from the F_∞-norm[(9)]. Thereby, neighboring genes in a gene interaction network are grouped together, forcing the SVM to select or eliminate genes adjacent in the network, i.e. genes lying in the same pathway. Rapaport et al. (2007) used the spectral decomposition of the gene interaction network in order to compute a discrete Fourier transformation from the gene

[(9)] The infinity norm, or max norm of a vector $\mathbf{x}$ is defined as $|\mathbf{x}|_\infty = \max\{|x_1|, \ldots, |x_n|\}$.

expression profiles. Again, the transformation of the gene expression profiles was used to define a new metric for gene expression profiles. This metric was used with a standard SVM as an example for a supervised learning algorithm[10]. SVMs were also used by Johannes et al. (2010). Here, a modified version of recursive feature elimination (RFE, Guyon et al., 2002) was used to incorporate *prior* pathway knowledge. Genes are ranked according to their connectivity in a gene interaction network (Morrison et al., 2005). Subsequently, this rank is included in RFE, an iterative feature selection used for SVMs.

Finally, some Bayesian approaches exist, allowing not only to incorporate *prior* biological pathway knowledge but also a measure of uncertainty for the final model (see e.g. Hill et al., 2012; Vannucci and Stingo, 2010).

1.3 Aim and Organization of the Thesis

While there are several methods that incorporate *prior* biological knowledge into prediction models using gene expression data, there is however still a need for methods using both gene expression and miRNA expression data at the same time. For the *fusion* of these two kinds of data the description about the regulatory dependencies of the features, mRNAs and miRNAs, is of central importance.

The focus of this thesis was to develop a workflow that allows the risk prediction of cancer patients where both, gene expression and miRNA expression data are available. As a learning method we chose boosting because it has proven its usability for high-dimensional microarray data (Dettling and Buhlmann, 2003; Dudoit et al., 2012), is able to handle different types of endpoints, and has a sound statistical foundation (see section 2.1 for details). A graph representing the regulatory relationships between the miRNAs and the mRNAs can be estimated from the expression data itself in combination with a target prediction database, in this case the MicroCosm target database (Enright et al., 2003). The intention was to use this graph together with the gene and miRNA expression data to built a better prediction model and improve feature selection.

[10] Rapaport et al. (2007) noted that the derived metric, incorporating gene expression and *a priori* network knowledge, can also be used with unsupervised methods, i.e. to cluster the biological samples.

The thesis is structured as followed. Chapter 2 gives insights about the theoretical background of the methods used in this work. Section 2.1 gives a general overview about boosting and the statistical interpretation of this method which originates from the machine learning field. Section 2.2 introduces CoxBoost, an adaption of the boosting method for Cox models, and PathBoost as a possibility to include *prior* biological knowledge in form of gene interaction networks in the model fitting process. The Cox model as well as the underlying fundamentals of time-to-event data are explained in 2.2.1. Section 2.3 briefly introduces two methods suited for high-dimensional time-to-event data. These are used as benchmarks to our workflow in terms of prediction accuracy. The following section 2.4 deals with model assessments and error measurements used in this thesis to judge the quality of a method and the resulting model. In section 2.5 we present the miRNA target prediction algorithm used for building the graph in our workflow. The chapter concludes with a description of the data set we used for evaluation of our workflow and the preprocessing of this data set.

The results chapter (chapter 3) explains our new workflow how to fit a model with gene and miRNA expression in order to predict a clinical endpoint (Gade et al., 2011). The description of the new workflow is followed by a thorough evaluation of the method. This includes the evaluation of the prediction error (section 3.2.1), the stability of the feature selection (section 3.2.2), and the comparison to the benchmark methods. Furthermore, the problem of overfitting and different target prediction algorithms is discussed (section 3.2.3 and 3.2.4).

Chapter 2

Material and Methods

2.1 Introduction to Boosting

An important part of machine learning, or statistical learning as it is called sometimes, is supervised learning. Assuming training data (y_i, x_i) with $i = 1, \ldots, n$ where y_i is the output or response and x_i is the predictor or feature. The task is now to find a prediction model capable of predicting y given x with high accuracy on observations where y is unknown. If the output is discrete, e.g. $y \in \{-1, 1\}$ this task is called classification. If the response is continuous it is called regression.

Boosting is one of the most powerful machine learning methods of the last years. Similar to other ensemble learners several weak learners are combined into a powerful committee. The prediction power of these simple base learners is boosted. The first approaches of boosting were introduced by Schapire (1990) and Freund (1995). The first practical, and todays most popular, boosting algorithm was AdaBoost (short for Adaptive Boost) described by Freund and Schapire (1996) (figure 2.1).

The original AdaBoost, called "AdaBoost.M1" (see algorithm 1), was designed for a 2-class classification problem. Such a classification problem can be describes as followed. Starting with the original training data, a new weighted sample is created in every step $m = 1, \ldots, M$ and used to build a simple classifier $G_m(x)$. In order to create a new sample the weights are adapted according to the classification performance. Observations which were classified poorly in previous steps gain more weight whereas the weight of

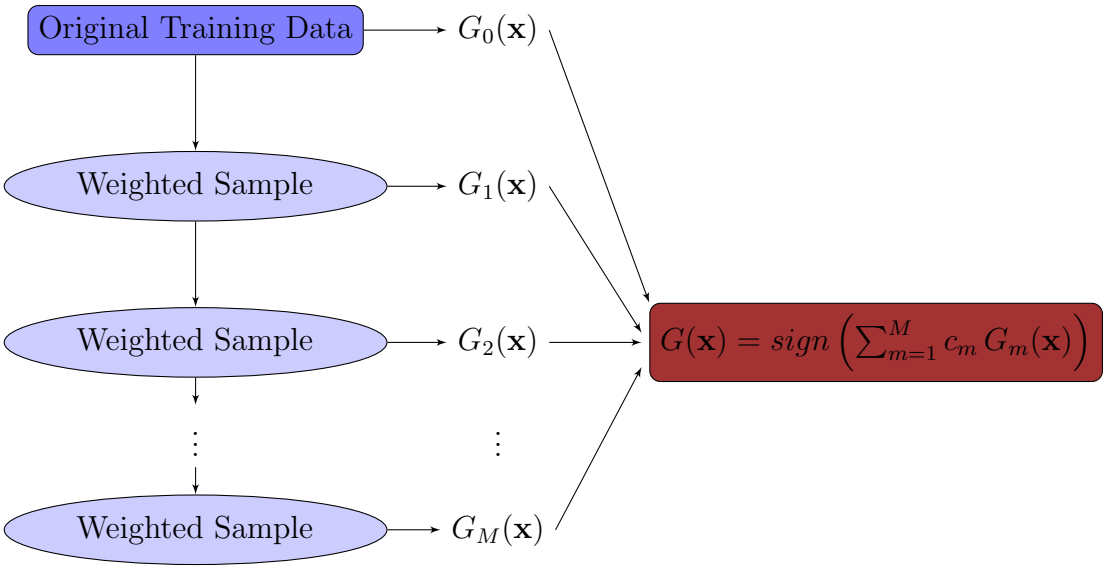


FIGURE 2.1. *The figure shows the basic principle of AdaBoost as introduced by Freund and Schapire (1996) (figure adapted from Hastie et al., 2009).*

Algorithm 1 AdaBoost.M1 (as described in Hastie et al., 2009; chap. 10)

1: initialize weights $w_i = 1/n \quad \forall i = 1, \ldots, n$
2: **for** $m = 1 \rightarrow M$ **do**
3: fit a weak classifier $G_m(x_i)$ using weights w_i
4: compute error

$$err_m = \frac{\sum_{i=1}^{n} w_i\, I(y_i \neq G_m(x_i))}{\sum_{i=1}^{n} w_i}$$

5: compute classifier weight $c_m = log\left(\frac{1-err_m}{err_m}\right)$
6: adjust weights $w_i \rightarrow w_i \exp\left(c_m\, I(y_i \neq G_m(x_i))\right)$
7: **end for**
8: final output is weighted combination $G(x) = \text{sign}\left[\sum_{m=1}^{M} c_m\, G_m(x)\right]$

correctly classified observations is decreased. Finally, the committee is built as weighted combination of the single classifiers

$$G(\mathbf{x}) = sign\left[\sum_{m=1}^{M} c_m\, G_m(\mathbf{x})\right] \tag{2.1}$$

The weights c_m are calculated from the weighted misclassification error of the single classifiers. Therefore, more accurate classifiers contribute more to the final committee.

An interesting observation is that the test error of AdaBoost decreases in most applications for a higher number of boosting steps M (Friedman et al., 2000). It seems to be resistant to overfitting. Fitting learners on samples of the training data suggests parallels to the bagging (short for bootstrap aggregation) procedure (Breiman, 1996) and that the success of boosting can be explained by reduction of variance. In contrast to bagging however, boosting performs well with stumps [(1)], learners which have typically a high bias and a low variance.

Some explanations for the success of boosting were given over the years. Schapire et al. (1998) explained the power of the committee by an increase of the margin. Increasing the margin results in a better separation of the classes and consequently a lower test error. Another explanation for the power of boosting lies in the expression of the final committee (2.1) and was found by Friedman (2008); Friedman et al. (2000) who established a statistical framework for boosting methods. Friedman et al. linked the idea of boosting with the statistical concept of additive models and loss functions. For a comprehensive overview on boosting and its statistical properties the interested reader is referred to (Hastie et al., 2009). The following remarks on boosting and its link to additive modeling are mostly derived from chapter 10.

An additive model has the form

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m\, b(\mathbf{x}; \gamma_m) \tag{2.2}$$

Usually, b are simple functions of the multivariate argument $\mathbf{x}$ characterized by a set of parameters γ_m. These functions are basis functions spanning a function

[(1)] trees with only two terminal nodes

space. In terms of boosting the basis functions are the weak learners and the basis function expansion f is the final committee. Thus, boosting can be

Algorithm 2 Forward Stagewise Additive Modeling (as described in Hastie et al., 2009; chap. 10)

1: initialize $f_0(x) = 0$
2: **for** $m = 1 \rightarrow M$ **do**
3: compute

$$(\beta_m, \gamma_m) = \underset{\beta,\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} l(y_i, f_{m-1}(x_i) + \beta\, b(\mathbf{x_i}; \gamma))$$

4: $f_m \leftarrow f_{m-1}(\mathbf{x}) + \beta_m\, b(\mathbf{x}; \gamma_m)$
5: **end for**

regarded as fitting an additive model minimizing a loss function, more precisely the exponential loss. Figure 2.2 shows an example which demonstrates that boosting optimizes the exponential loss and not the misclassification rate.

Definition 1. Loss function
Consider a response variable Y, a vector of predictors X, and a prediction model $f(X)$ trained on a training set $\mathcal{T}$. A function

$$l : (Y, f(X)) \rightarrow \mathbb{R}$$

measuring the deviance of Y and $f(X)$ is called *loss function*. Typical choices are

$$l(Y, f(X)) = I(Y \neq f(X)) \qquad (0-1 \text{ loss or misclassification}) \tag{2.3}$$

$$l(Y, f(X)) = \exp(-Y\, f(X)) \qquad (\text{exponential loss}) \tag{2.4}$$

$$l(Y, f(X)) = (Y - f(X))^2 \qquad (\text{squared error loss}) \tag{2.5}$$

A more complex loss function is the Huber loss. For small values of $Y - f(X)$ it imitates the squared loss whereas larger differences are penalized linear.

$$L(Y, f(X)) = \begin{cases} (Y - f(X))^2 & |Y - f(X)| \leq \delta \\ 2\delta(|Y - f(X)| - \delta^2) & \text{otherwise} \end{cases} \quad (\text{Huber loss}) \tag{2.6}$$

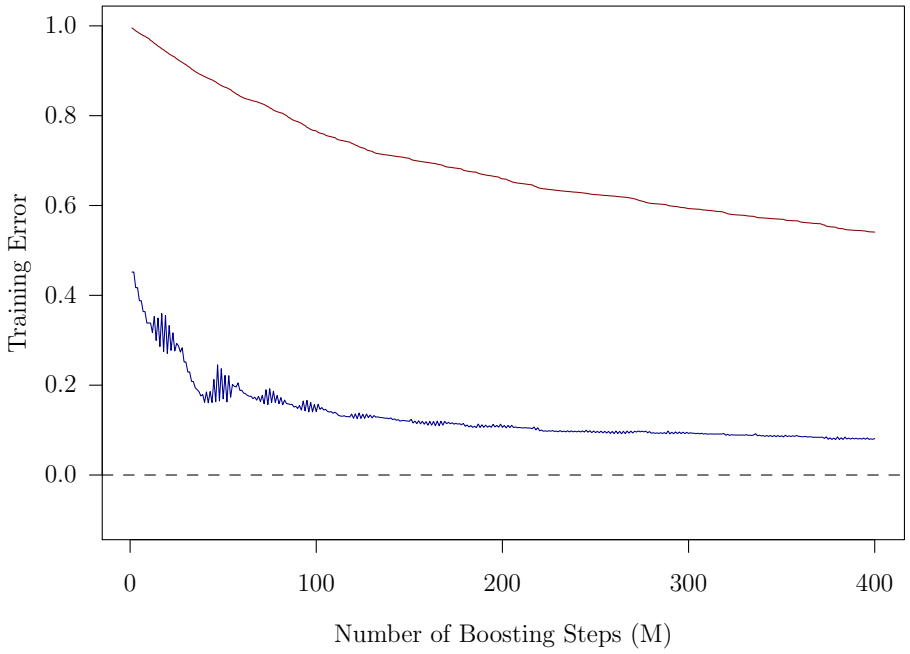


FIGURE 2.2. *The figure shows the training error of boosting as a function of the number of boosting steps. This is an example with synthetic data from Hastie et al. (2009). Ten normal Gaussian predictors* $x_1, \cdots, x_{10}$ *are used. The binary output is calculated as* $y = 2\, I(\sum_j x_j^2 > \chi^2_{10}(0.5)) - 1$. *The training set comprised* 12000 *cases. The blue line indicates the misclassification rate (*$\frac{1}{n} \sum_i^n I(y_i \neq G(\mathbf{x_i}))$*) and the red line the average exponential loss (*$\frac{1}{n} \exp(-y_i\, f(\mathbf{x}_i))$*). The misclassification rate remains nearly constant after* 250 *steps whereas the exponential loss keeps dropping. Clearly, boosting does not optimize the misclassification but rather the exponential loss.*

When fitting the additive model (algorithm 2), the crucial step is to find the pair (β_m, γ_m). Since, for boosting, the basis functions are the weak learners $G_m(x)$, this yields

$$(\beta_m, G_m) = \underset{\beta, G}{\operatorname{argmin}} \sum_{i=1}^{n} \exp\left[-y_i \left(f_{m-1}(x_i) + \beta\, G(x_i)\right)\right] \tag{2.7}$$

and therewith

$$(\beta_m, G_m) = \underset{\beta, G}{\operatorname{argmin}} \sum_{i=1}^{n} w_i^{(m)} e^{-y_i\, \beta\, G(x_i)} \tag{2.8}$$

where

$$w_i^{(m)} = e^{-y_i\, f_{m-1}(x_i)} \tag{2.9}$$

can be regarded as weight independent from β_m and G_m. They depend only on the solution from the prior iteration $m-1$ and will change with every new boosting step. (2.8) can be solved independently for G_m and β_m. For a fixed $\beta_m \neq 0$ [2] (2.8) can be written as

$$G_m = \underset{G}{\operatorname{argmin}} \sum_{i=1}^{n} w_i^{(m)} e^{-\beta\, y_i\, G(x_i)} \tag{2.10}$$

By splitting the sum we get

$$\sum_{y_i = G(x_i)} w_i^{(m)} e^{-\beta\, y_i\, G(x_i)} + \sum_{y_i \neq G(x_i)} w_i^{(m)} e^{-\beta\, y_i\, G(x_i)} \tag{2.11}$$

In the first sum $y_i\, G(x_i)$ is equal 1 and in the second sum it is equal -1. With that in mind (2.11) can be simplified to

$$\sum_{y_i = G(x_i)} w_i^{(m)} e^{-\beta} + \sum_{y_i \neq G(x_i)} w_i^{(m)} e^{\beta} \tag{2.12}$$

[2] $\beta_m = 0$ would be the trivial case that G_m has no contribution to the final model.

The sums can be extended again by introducing the indicator function $I(y_i \neq G(x_i))$

$$e^{-\beta} \sum_{i=1}^{n} w_i^{(m)} - e^{-\beta} \sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G(x_i)) + e^{\beta} \sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G(x_i)) \quad (2.13)$$

which can be written as

$$e^{-\beta} \sum_{i=1}^{n} w_i^{(m)} + (e^{\beta} - e^{-\beta}) \sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G(x_i)) \quad (2.14)$$

Since the first sum and the factor of the second sum are independent of G_m, the classifier has to minimize the second sum, the weighted prediction error rate

$$G_m = \underset{G}{\operatorname{argmin}} \sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G(x_i)) \quad (2.15)$$

Now a solution for β_m can be derived by substitution of G by G_m in (2.8) and setting the partial derivation to zero

$$0 = \frac{\partial}{\partial \beta} \sum_{i=1}^{n} w_i^{(m)} e^{-\beta\, y_i\, G_m(x_i)} \quad (2.16)$$

Solving the partial derivation gives

$$0 = \sum_{i=1}^{n} w_i^{(m)} e^{-\beta\, y_i\, G(x_i)} \left(-y_i\, G_m(x_i)\right) \quad (2.17)$$

As before, the sum can be divided according to right and wrong classified samples, making it possible to solve $y_i\, G_m(x_i)$ for each part

$$0 = \sum_{y_i = G_m(x_i)} -w_i^{(m)} e^{-\beta} + \sum_{y_i \neq G_m(x_i)} w_i^{(m)} e^{\beta} \quad (2.18)$$

Again both sum can be extended to all samples using the indicator function. Furthermore e^{β} and $e^{-\beta}$ can be placed outside of the sums

$$0 = (e^{\beta} + e^{-\beta}) \sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G_m(x_i)) - e^{-\beta} \sum_{i=1}^{n} w_i^{(m)} \tag{2.19}$$

and therewith

$$\frac{e^{\beta} + e^{-\beta}}{e^{-\beta}} = \frac{\sum_{i=1}^{n} w_i^{(m)}}{\sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G_m(x_i))} \tag{2.20}$$

$$e^{2\beta} = \frac{\sum_{i=1}^{n} w_i^{(m)}}{\sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G_m(x_i))} \tag{2.21}$$

$$\beta = \frac{1}{2} \ln \left(\frac{\sum_{i=1}^{n} w_i^{(m)}}{\sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G_m(x_i))} \right) \tag{2.22}$$

Let err_m denote the weighted and normalized error rate minimized by G_m

$$err_m = \frac{\sum_{i=1}^{n} w_i^{(m)} I(y_i \neq G_m(x_i))}{\sum_{i=1}^{n} w_i^{(m)}} \tag{2.23}$$

Together with (2.22) this gives the solution for β_m

$$\beta_m = \frac{1}{2} \ln \left(\frac{1 - err_m}{err_m} \right) \tag{2.24}$$

The update rule for the additive expansion

$$f_m(x) = f_{m-1}(x) + \beta_m G_m(x)$$

in algorithm 2 implies the update of the weights

$$w_i^{(m+1)} = e^{-y_i f_m(x_i)} \tag{2.25}$$

$$= e^{-y_i f_{m-1}(x_i)} e^{-\beta_m y_i G_m(x_i)} \tag{2.26}$$

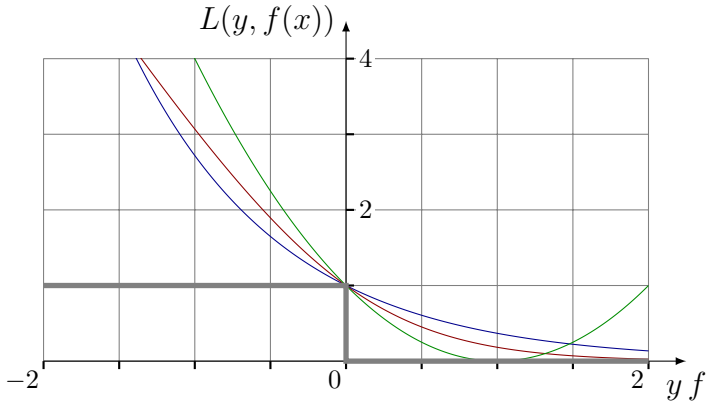
Using the fact that $w_i^{(m)} = e^{-y_i f_{m-1}(x_i)}$ and $y_i G_m(x_i) = -2 I(y_i \neq G_m(x_i)) + 1$ this can be written as

$$w_i^{(m+1)} = w_i^{(m)} e^{c_m I(y_i \neq G_m(x_i))} e^{-\beta_m} \tag{2.27}$$

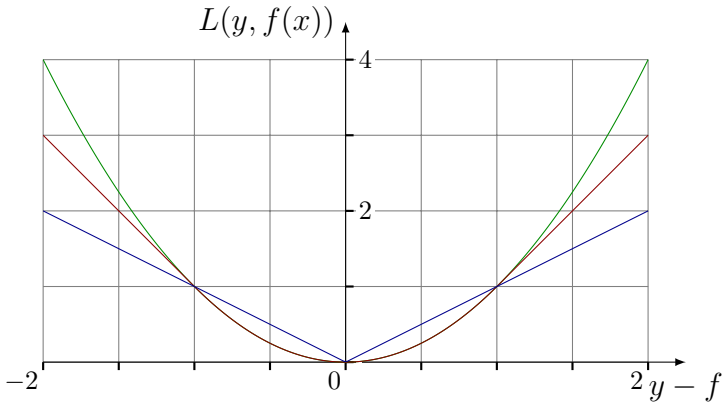
with $c_m = 2\beta_m$. The factor $e^{-\beta_m}$ is independent of i and hence has no effect. With this in mind (2.27) is equivalent to step 6 in AdaBoost.M1 (algorithm 1). Fitting the weak classifier $G_m(x)$ in step 3 can be seen as the search for the optimum of (2.8) (since the trained classifier should minimize the misclassification rate). The final committee in step 8 is in principle sign($f_M(x)$), the sign of the additive expansion[3].

Putting this together Friedman et al. (2000) concluded that AdaBoost.M1 is equivalent to forward stagewise additive modeling minimizing the exponential loss and therewith reasoned the power of this technique. From this point of view boosting is no longer restricted to classification problems but can be used also for regression tasks. Different loss function can replace the exponential loss underlying the original AdaBoost.M1 algorithm.exponential loss underlying the original AdaBoost.M1 algorithm. Figure 2.3 compares different loss functions for classification and regression.

[3]The difference lies in a constant factor since $c_m = 2 * \beta_m$. However, this factor can be placed outside the sum and has no influence to the final result.



(a) Loss functions for classification



(b) Loss functions for regression

FIGURE 2.3. *Figure (a) shows loss functions as functions of the margin $y\,f(x)$. The margin plays a role as error estimate for classification problems with $y \in -1, 1$. The losses are: exponential loss (green) and binomial deviance $log(1 + e^{-2y\,f}$ (blue). Both functions decrease with increasing margin. The third function (green) is the squared loss. Increasing with positive margin (rightly classified) this loss function is less suited for classification tasks. As a reference the misclassification (grey) is given. (b) shows losses as functions of the residual $y - f$, a common error measure in regression tasks. Again, the green curve is the squared loss. The blue curve is the absolute loss $|y - f|$ and the red line is the Huber loss* (2.6)*. Since the squared-error loss emphasises observations with large residuals during the model fit it is less robust and prone to outliers. A more robust choice is the absolute loss or the Huber loss used for M-regression which is resistant against heavy outliers and nearly as efficient for Gaussian errors as least squares (adapted from Hastie et al., 2009).*

2.2 Boosting for Cox Models

2.2.1 Time-to-event Data

In the following section the basic concepts and two fundamental functions used for analysis of survival data will be introduced. For a more detailed overview the interested reader is referred to Tableman and Kim (2004) and Everitt and Hothorn (2006; chap. 9).

Survival data or, more general, time-to-event data (in the following survival time and time-to-event are used interchangeable) usually consist of n observations (e.g. patients) $(t_i, \delta_i, \mathbf{x_i})$ with $1 \leq i \leq n$. t_i is the time the event of interest occurred or the observation was censored[4] and δ_i is the censoring status indicating such a censoring. The third part is an observation specific p-dimensional vector of features or covariates $\mathbf{x_i}$.

The time points t_i can be considered as realizations of a random variable T with a probability density function $f(t)$ and a distribution function

$$F(t) = P(T \leq t) = \int_0^t dx\, f(x) \tag{2.28}$$

When dealing with time-to-event data, two functions are from central importance. The first function is the *survivor function*

$$S(t) = P(T \geq t) = 1 - F(t) = \int_t^\infty dx\, f(x) \tag{2.29}$$

which is defined as the probability that the survival time T is greater or equal a specified time t. That means the survivor function is a time-dependent function explaining how likely it is to be a survivor (event-free) at a given time point. The second function is the *hazard function*

$$h(t) = \lim_{\Delta t \to 0^+} \frac{P(t \leq T \leq t + \Delta t \,|\, T \geq t)}{\Delta t} \tag{2.30}$$

[4] More precisely the data are said to be *right-censored*. For right-censored data the event did not occur until the end of the study or other reasons made it impossible to track the status of the patient. In this case the reason has to be independent from the event of interest.

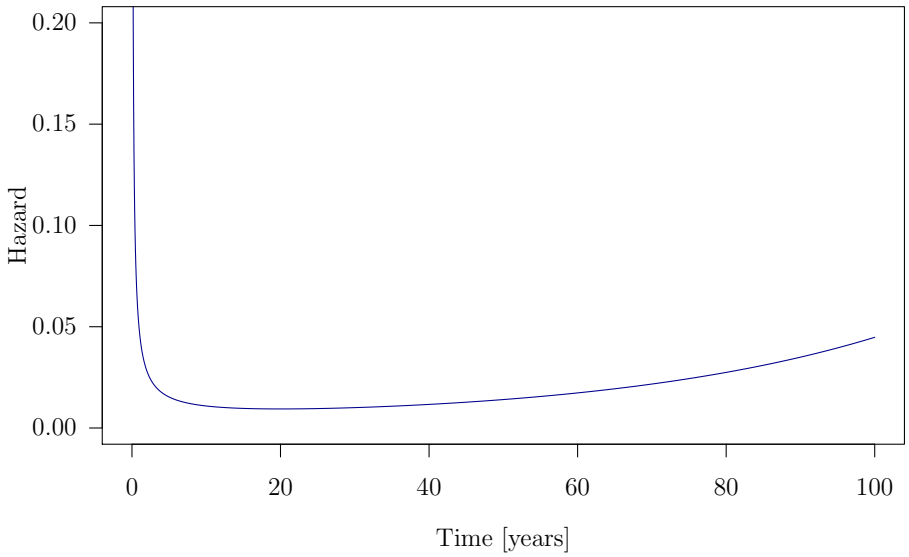


FIGURE 2.4. *The figure shows the "Bath tub" shape of a hazard function. It describes the hazard for death in human beings. It starts high right after birth which is caused by a high infant mortality. In the middle ages the hazard has a low plateau indicating a low death rate. In later years the hazard rises again due to the aging process (adapted from Everitt and Hothorn, 2006)*

defined as the instantaneous rate of failure (having an event) at time $T > t$. Therewith $h(t)\,\Delta t$ is the probability of having the event at time t given the fact that the individual was event free to time t. The condition is essential, e.g. it is unlikely to die at an age of 100 simply for the fact that most people do not reach that age. However, it is much more likely to die at an age of 100 given that the person actually get that old.

The hazard function is often referred to as risk or mortality rate. It is important to note that the hazard is not a probability but a rate which can be seen from (2.30). A conditional probability per unit time is a rate and can have values in the interval $[0, \infty]$. Figure 2.4 shows an example of a hazard function. It is known as "bath tub hazard" of death in human beings. Integrating the hazard function over time gives the cumulative hazard function

$$H(t) = \int_0^t du \quad h(u) \tag{2.31}$$

and by this the connection between hazard and survivor function

$$S(t) = \exp(-H(t)) = \exp(-\int_0^t dx\, h(x)) \tag{2.32}$$

One of the most well known estimates of the survivor function (2.29) is the non-parametric Kaplan-Meier estimate (Kaplan and Meier, 1958)

$$\hat{S}(t) = \prod_{j:t_j \leq t} \left(1 - \frac{d_j}{r_j}\right) \tag{2.33}$$

where d_j is the number of individuals having an event at time t_j and r_j is the number of individuals at risk that means without an event right before t_j. That includes the individuals censored at time point t_j. Figure 2.5 shows the Kaplan-Meier estimate from the *glioma* data set from the coin R package (Hothorn et al., 2011). The data comprises 37 patients suffering from two different types of glioma (Grana et al., 2002), the time of survival and different clinical information. Table 2.1 summarizes the example data set. Based on the estimate of the survivor function the estimate of the cummulative hazard function can be derived as

$$\hat{H} = -\log(\hat{S}(t))$$

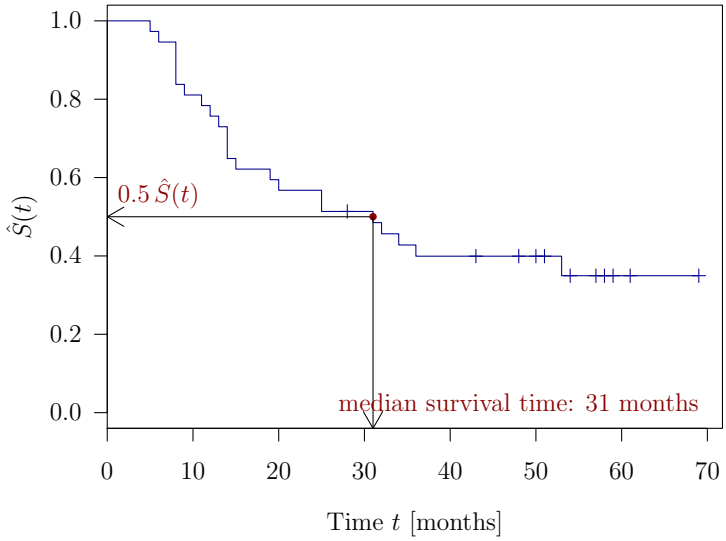
The effect of a covariate on the survivor function can be estimated by building two groups and estimating the survivor function for each of them. The resulting survivor functions can be tested for differences with help of the logrank test (cf. Hosmer et al., 199; chap. 2 for an overview on survivor functions and associated tests).

A more flexible and general approach was given by Cox (1972). The Cox's proportional hazards model or shortly Cox's regression does not model the survivor function directly but the hazard function

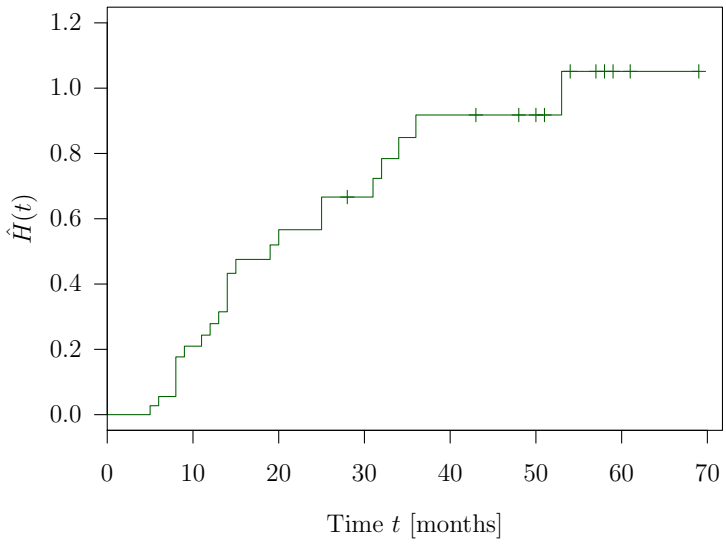
$$h(t|\mathbf{x_i}) = h_0(t)\, e^{\eta_i} \tag{2.34}$$

Age	Sex	Histology	Group	Status	Time
83	Female	GBM	Control	Event	5
61	Male	GBM	Control	Event	6
32	Female	GBM	Control	Event	8
70	Male	GBM	Control	Event	8
57	Female	GBM	Control	Event	8
71	Female	GBM	Control	Event	8
53	Female	Grade3	Control	Event	9
72	Male	GBM	Control	Event	11
46	Male	GBM	Control	Event	12
50	Male	GBM	Control	Event	13
39	Female	GBM	RIT	Event	14
40	Female	GBM	RIT	Event	14
65	Male	GBM	Control	Event	14
44	Male	GBM	Control	Event	15
46	Male	Grade3	Control	Event	19
70	Male	GBM	RIT	Event	20
31	Male	Grade3	RIT	Event	25
42	Female	GBM	Control	Event	25
45	Female	Grade3	RIT	Censored	28
58	Male	GBM	RIT	Event	31
32	Male	Grade3	Control	Event	32
27	Male	Grade3	Control	Event	34
40	Female	GBM	RIT	Censored	36
36	Male	GBM	RIT	Event	36
55	Female	GBM	RIT	Censored	43
19	Female	Grade3	Control	Censored	48
57	Male	Grade3	RIT	Censored	50
33	Female	Grade3	Control	Censored	50
53	Male	Grade3	RIT	Censored	51
41	Female	Grade3	RIT	Event	53
40	Female	Grade3	RIT	Censored	54
36	Male	Grade3	RIT	Censored	57
52	Male	Grade3	RIT	Censored	57
54	Male	Grade3	RIT	Censored	58
47	Female	GBM	RIT	Censored	59
49	Male	Grade3	RIT	Censored	61
48	Male	Grade3	RIT	Censored	69

TABLE 2.1. *The table summarizes the glioma data set from Grana et al. (2002) packed in the coin R package (Hothorn et al., 2011). The data comprises 37 patients with two types of glioma. They have been treated with a standard therapy (Control) and a new radioimmunotherapy (RIT). The event of interest is death, the survival time is given in months.*



(a) Kaplan-Meier estimate of the survivor function



(b) Kaplan-Meier estimate of the cumulative hazard

FIGURE 2.5. *The figure shows the Kaplan-Meier estimates for the glioma data set from Grana et al. (2002) packed in the R package coin (Hothorn et al., 2011). Figure (a) shows the estimate of the survivor function. An easy to see but important indicator is the median survival time, the time where the survivor function reaches a level of* 0.5*, in this case 31 months. Due to too few patients at risk the median survival time is not always observable. Figure (b) shows the resulting estimate of the cumulative hazard function. Note that the cumulative hazard is not a probability and thus not limited to the interval* $[0, 1]$*. In both figures the censoring of patients is indicated by small crosses in the function plot.*

with an unspecified baseline hazard $h_0(t)$ and a linear predictor

$$\eta_i = \mathbf{x_i}^T \boldsymbol{\beta} \tag{2.35}$$

Since the only time-dependent term is the baseline hazard the ratio of the hazards of two patients becomes

$$HR = \frac{h(t|\mathbf{x_1})}{h(t|\mathbf{x_2})} = \frac{\exp(\mathbf{x_1}^T \boldsymbol{\beta})}{\exp(\mathbf{x_2}\, \boldsymbol{\beta})} \tag{2.36}$$

$$= \exp((\mathbf{x_1} - \mathbf{x_2})^T \boldsymbol{\beta}) \tag{2.37}$$

The hazard (HR) is the usual measure of effect of the predictors of interest in survival analysis, comparable e.g. with the odds ratio in logistic regression. An important fact is that the baseline hazard is not included, HR depends solely on the parameter vector $\boldsymbol{\beta}$ and thus is constant over time. This is called the proportional hazard property.

Cox (1972) derived a method to estimate $\boldsymbol{\beta}$ without specifying the baseline hazard. Therefore the Cox model is sometimes referred to as a semi-parametric model. In fact h_0 can be described by a variety of functions which makes the Cox model quite general and powerful. Since the probability density function depends on the baseline hazard so does the likelihood $l(t, \beta)$. It is therefore not possible to perform a regular Maximum Likelihood approach to estimate the parameters. Instead Cox derived a partial likelihood based on conditional probabilities.

Let $t_{(1)}, \ldots, t_{(r)}$ $(r \leq n)$ the increasing times of event without time points where an individual was censored. $\mathcal{R}(t_{(j)})$ is the risk set containing the indices of individuals at risk at time $t_{(j)}$. Furthermore, $\mathbf{x_{(j)}}$ denotes the vector of covariates corresponding to the individual with an event at $t_{(j)}$. Now, conditional probabilities can be defined (cf. Tableman and Kim, 2004 for more details) describing the probability that the individual with $\mathbf{x_{(j)}}$ has an event at time $t_{(j)}$ given that the individual is at risk at this time. This can be written as

$$L_j(\boldsymbol{\beta}) = \frac{h_0(t_{(j)})\, \exp(\mathbf{x_{(j)}}^T \boldsymbol{\beta})}{\sum_{l \in \mathcal{R}(t_{(j)})} h_0(t_{(j)})\, \exp(\mathbf{x_l}^T \boldsymbol{\beta})} \tag{2.38}$$

$$= \frac{\exp(\mathbf{x_{(j)}}^T \boldsymbol{\beta})}{\sum_{l \in \mathcal{R}(t_{(j)})} \exp(\mathbf{x_l}^T \boldsymbol{\beta})} \tag{2.39}$$

Multiply these over the r event times yields Cox's partial likelihood function

$$L(\boldsymbol{\beta}) = \prod_{j=1}^{r} L_j(\boldsymbol{\beta}) = \prod_{j=1}^{r} \frac{\exp(\mathbf{x_{(j)}}^T \boldsymbol{\beta})}{\sum_{l \in \mathcal{R}(t_{(j)})} \exp(\mathbf{x_l}^T \boldsymbol{\beta})} \tag{2.40}$$

By considering the censoring status, all time points can be used in the product

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left(\frac{\exp(\mathbf{x_i}^T \boldsymbol{\beta})}{\sum_{l \in \mathcal{R}(t_i)} \exp(\mathbf{x_l}^T \boldsymbol{\beta})} \right)^{\delta_i} \tag{2.41}$$

Note that (2.41) is not a true likelihood since it do not integrate to 1. However, Cox argued that most of the relevant information about the parameter $\boldsymbol{\beta}$ is covered in the partial likelihood and it is sufficient to maximize this partial likelihood (or more specific the log-partial likelihood) via a Newton-Raphson algorithm. The partial likelihood does not depend on the event times directly but the rank of the event times. It is therefore sometimes referred to as a non-parametric approach. It also important to note that in the multidimensional case ($n \ll p$) the model cannot be fit the classical way[5].

The estimated parameters $\hat{\boldsymbol{\beta}}$ and the associated estimated standard deviations can be used to test the influence of the single predictors on the HR. Also, based on (2.32), the survivor function can now be estimated. Therefore an estimation of the cummulative baseline hazard (and therewith of the baseline hazard) is needed. Several parametric approaches exists if a reasonable assumption about the distribution of h_0 can me made (cf. Tableman and Kim, 2004 for details). An often used non-parametric approach is the Breslow estimator of cummulative baseline hazard (Breslow, 1972) that follows directly from the parameter estimation in the Cox model

$$\hat{H}_0(t) = \sum_{t_{(j)} \leq t} \hat{h_0}(t_{(j)}) = \sum_{t_{(j)} \leq t} \frac{1}{\sum_{l \in \mathcal{R}(t_{(j)})} \exp(\mathbf{x_j}^T \hat{\boldsymbol{\beta}})} \tag{2.42}$$

[5] Like in any regression setting the case where the number of predictors overcomes the number of observations the behavior is degenerated. All β_i would be estimated to $\pm\infty$.

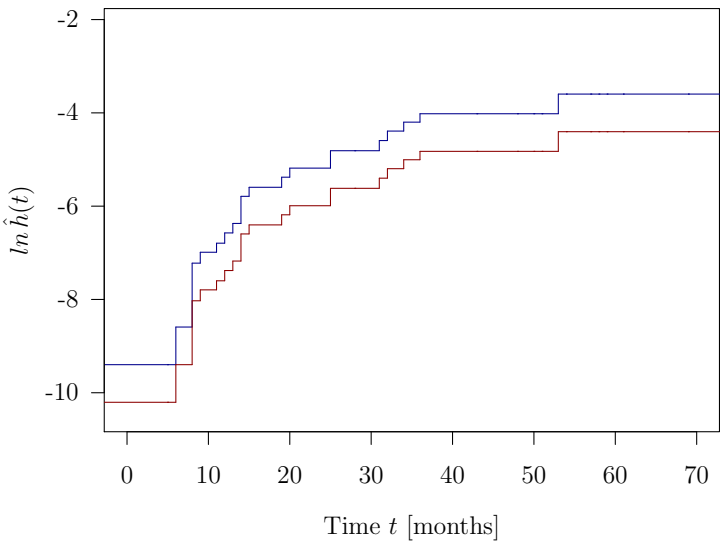


FIGURE 2.6. *The figure shows estimates of the hazard function based on a Cox model for the glioma data set from Grana et al. (2002). The blue curve is the hazard for patients in the control group, the red one the hazard from the group with the new radioimmunotherapy RIT. The baseline hazard is the Breslow estimate. The data are shown on a log_e scale, revealing the proportional hazard property. Is is obvious that the therapy has an huge effect on the hazard and therewith on the survivor function. The patients will gain from this therapy.*

Figure 2.6 shows an example of the hazard estimates based on a Cox model including the Breslow estimate of the baseline hazard. The estimates of the hazards can be used to get an estimate of the survivor function and with this a risk predcition model can be formulated

$$\hat{r}(t|\mathbf{x}) = \hat{S}(t|\mathbf{x}) = \exp(-\hat{H}_0(t)\,\exp(\mathbf{x}^T\,\hat{\boldsymbol{\beta}})) \tag{2.43}$$

2.2.2 Likelihood Boosting and Implicit Feature Selection

2.2.2.1 Introduction to Feature Selection

The main goal of building a prediction model, either to predict a discrete outcome (classification) or a continuous (regression), is to *learn* a prediction rule $y = F(\mathbf{x})$ based on training data $y_i, \mathbf{x_i}$ $(i = 1, \ldots, n)$ where $\mathbf{x_i}$ are p-dimensional feature vectors. In the case of high-dimensional data it usually holds $p \ll n$ and several problems occur, e.g. the curse of dimensionality and the sparseness of the feature space.

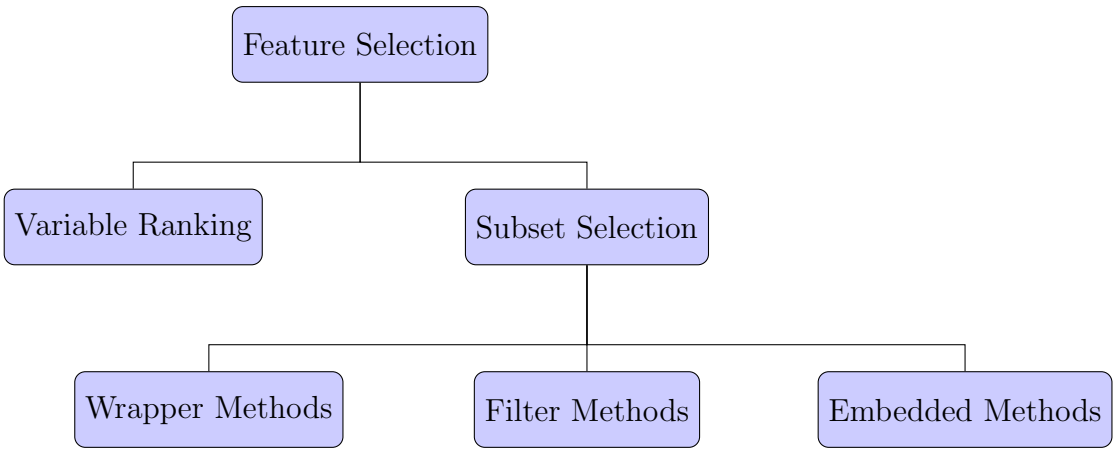


FIGURE 2.7. *The figure shows the different kinds of feature selection methods.*

Usually, it is unclear if all the features (and therewith all dimensions in the p-dimensional space) are needed to infer F. In contrast, uninformative, noisy features can mask the underlying relationship between features and the outcome and lead to worse estimations. For most algorithms the complexity depends on the number of (trainings)-observations n and on the number of features p. Several techniques have been developed to overcome these problems (cf. Alpaydin, 2010; chap. 6 for an overview). They can be roughly divided into two categories: *feature extraction* and *feature selection*.

Especially in the field of signal processing and pattern recognition feature extraction methods a very common. The task is to transform the high-dimensional input data into data of less dimensions that means to create a set of k new features from the original p. The transformation can be linear, e.g in the case of Principal Component Analysis (PCA) (Pearson, 1901), or non-linear.

PCA finds linear combinations of the input features explaining most of variance. Afterwards these linear combinations (principal components) can be used for the learning task. Other similar approach are Multidimensional Scaling (MDS) and Linear discriminant analysis (Fisher, 1936). LDA finds, similar to PCA, a linear projection of the original data but in a supervised fashion that means using the output. In that manner clustering, e.g. the famous k-means algorithm (Lloyd, 1982; MacQueen, 1967), can be used to build combinations of features which are similar to each other.

Non-linear feature extraction methods comprises e.g. Locally linear embedding (LLE) and kernel based methods (cf. Schölkopf and Smola, 2002 for an comprehensive overview on kernel based learning algorithms).

In contrast, methods of the second category, the feature selection methods (figure 2.7), try to find k out of p features improving the prediction model. The resulting models are better interpretable and in the biological case subsequent analysis might be more practicable in sense of effort and costs[6].

Two general approaches can be distinguished (Guyon and Elisseeff, 2003; Kohavi and George H. John, 1997). Variable ranking tries to identify and rank relevant variables. Usually this is done by utilizing a score function related to the outcome, e.g. the correlation (e.g. Golub et al., 1999) or the t-statistic (e.g. Tusher et al., 2001). Variable ranking is a very general approach not limited to building a prediction model. The variables alone are usually of interest (e.g. genes which are differential expressed between two conditions). Furthermore, the most relevant variables are usually suboptimal for building a prediction model (Guyon and Elisseeff, 2003).

The second category are *subset selection* methods. Here not the predictive power of a single feature is of interest but the focus lies on finding am optimal subset of p variables. For such an optimal subset (given a suited optimality criterion) in principal all $2^p - 1$ subsets need to be considered. While this can be done for small p it is impractical for large dimensional data sets[7]. Instead heuristics are used to get a reasonable (but in most cases sub-optimal) subset in polynomial time. Guyon and Elisseeff (2003) divides this class of methods into filter methods, wrapper methods, and embedded methods.

Filter methods are a preprocessing step where the features are filtered based on a certain criterion independent of the subsequent learning algorithm. According to Kohavi and George H. John (1997) the ranking of variables is a filter method where the top ranked variables are used as subset. The number of variables to be taking has to be determined separately. Several methods have been proposed (see Guyon and Elisseeff, 2003 for an overview). In case of

[6] In the biomedical research methods of the second category are preferred since linear or non-linear combinations of the input features are harder to interpret than subsets of the original features, e.g. genes or miRNAs. Dimensional reduction methods are often used to find outliers in the outliers or inspect a general separability of two classes.

[7] In fact, finding the optimal subset is known to be NP-hard (Amaldi and Kann, 1998).

a test statistic, as used by e.g. Tusher et al. (2001), a significance level can be used to estimate the number of informative features.

However, as with variable ranking the most informative features do not necessarily form an optimal feature set. However, since it is basically a ranking of variables, filter methods are fast and as a preprocessing step not tuned for a specific learning algorithm. The reduction of the feature space prior to the actual model fit can be used to overcome the risk of overfitting.

In contrast, wrapper methods (Kohavi and George H. John, 1997) do not asses single features but sets of features. Similar to filters the actual feature selection is a separate step. The learning algorithm is considered a perfect black box. In every iteration a defined set of features or variables is given to the algorithm and the prediction result is assessed. Thus, a wrapper method has to specify two important aspects: (1) How to search the space of possible feature subsets and (2) how to assess the prediction result.

A wide range of search strategies can be used e.g. hill-climbing, best-first, and simulated annealing (cf. Kohavi and George H. John, 1997 for an overview). Like in classical statistic regression models, for greedy strategies two modes of directions are possible: forward selection and backward elimination. Forward selection starts with an empty model, adding the most promising features in every search step, whereas backward elimination progressively eliminates the least promising features from a full model. In high dimensional data where usually only small subsets are of interest forward selection search strategies are computationally less expensive since the learning algorithm operates with much less features compared to backward elimination[(8)]. Both methods produce a nested sequence of subsets. Independent from the direction of search a appropriate measure of the goodness-of-fit is needed in every search step to evaluate the candidate subsets. The search stops if no improvement of the prediction performance can be achieved (Langley, 1994) or pre-defined number of features has been reached.

The wrapper methodology is a rather general concept since the underlying learning algorithm is used as a black box. Thus it can be used for many settings.

[(8)] Dependent on the learning algorithm it might be impossible to fit the model with all features. The simple least-squares estimator for a regression setting for example cannot be computed in the case $p > n$.

On the other side it is often criticized as a "brute force" attempt (Guyon and Elisseeff, 2003) as the space of possible subsets is searched systematically.

A more directed approach are embedded methods. They incorporate the feature selection as part of the training process. Consequently they are more efficient since the re-training and evaluation for every candidate subset can be omitted. Some methods use changes in the objective function together with a greedy search in the feature subset space, e.g. Recursive Feature Elimination (RFE) for SVMs (Guyon et al., 2002). Other methods incorporate a penalty term in the objective function (Bi et al., 2003; Tibshirani, 1996; Weston et al., 2003) to shrink the parameter space and get sparse model fits.

2.2.2.2 GAMBoost and CoxBoost

As shown before (section 2.1) boosting can be seen as a method for function estimation using stagewise, additive modeling with a suited loss function. Dependent on the loss function and the base learners it suited for classification as well as regression tasks. As pointed out by Bühlmann and Hothorn (2007) this makes boosting a very general and powerful method. For example, by replacing the exponential loss underlying AdaBoost with the L_2 loss function (squared error loss) $(y-f)^2/2$ Bühlmann and Yu (2003) derived L_2Boost suited for classification and regression tasks.

Another important class of loss functions is likelihood based, e.g. LogitBoost (Friedman et al., 2000) where the negative log-likelihood is minimized (and therewith the likelihood is maximized). GAMBoost (boosting for general additive models, Tutz and Binder, 2006), another member of this class of boosting algorithms, is shortly explained in the following.

Assuming training data $(y_i, \mathbf{x_i})$ a generalized additive model (see Chambers and Hastie, 1992; chap. 6,7 and Hastie et al., 2009; chap. 9 for an introduction) has the form

$$\mu_i = E(y_i|\mathbf{x_i}) = h(\eta_i) \tag{2.44}$$

and

$$\eta_i = f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) \tag{2.45}$$

h is a specified response function[9]. The functions f_j are unspecified smooth ("nonparametric") functions. In the case where $\eta_i = \mathbf{x_i}^T \boldsymbol{\beta}$ is a simple linear predictor the model becomes a generalized linear model (GLM).

By changing the link function several distributions of the response can be modeled, usually distributions of the exponential family including Gaussian, binomial, and Poisson. With this general linear or general additive models are a fairly general model family useful for many applications including not only regression but also classification tasks[10]. Several algorithms have been proposed to estimate the additive model. Assuming a sufficiently smoothness of the functions f_j e.g. the backfitting algorithm (Hastie and Tibshirani, 1986) can be used.

These algorithms works fine if the set of variables and the associated smoothing parameters are fixed. In case of high-dimensional data ($p > n$ predictors) few most influential variables have to be selected. GAMBoost uses maximization of the log-likelihood to estimate the additive model (an introduction into Maximum Likelihood for model inference can be found in Hastie and Tibshirani, 1986; chap. 8). When the distribution of $y_i|\mathbf{x_i}$ is from the exponential family that means the conditional density of y_i can be written as

$$f(y_i|\mathbf{x_i}) = \exp\left(\frac{y_i\Theta_i - b(\Theta_i)}{\phi} + c(y_i, \phi)\right) \tag{2.46}$$

where Θ_i is the canonical parameter and ϕ a dispersion parameter. Following the boosting principle, GAMBoost fits simple base learners that means simple functions of the variables

$$\eta_i = \eta(\mathbf{x_i}, \gamma) \tag{2.47}$$

where γ is the parameter of the base learner. Now, a log-likelihood can be formulated as a function of the desired parameter γ. Since the likelihood is, under the assumption that the observations y_i are independent of each other,

[9] Some authors, e.g. Chambers and Hastie (1992), use the notation $g(\mu) = \eta$ where $g = h^{-1}$ is called link function.
[10] For e.g. a binary outcome a Bernoulli distribution can be assumed.

simply the product of the densities, the log-likelihood is the sum over the log-densities of y_i

$$l(\gamma) = \sum_{i=1}^{n} l(y_i, \eta_i) \tag{2.48}$$

$$= \sum_{i=1}^{n} \frac{y_i \, \Theta_i - b(\Theta_i)}{\phi} + c(y_i, \phi) \tag{2.49}$$

Note, in this case the canonical parameter Θ_i is simply a function of the base learner η_i and therewith a function of the feature vector $\mathbf{x_i}$ and the parameters of the base learner γ.

Often utilized functions in the field of non-parametric function estimation are smoothing splines. The basic idea is to fit piecewise-polynomial functions to the data. GAMBoost uses a special form of smoothing splines called B-splines[(11)] as base learners. B-splines (basis splines) are a method of constructing a function from simple basis functions which are defined recursively. The linear combination of the basis functions forms the function estimate. The placements of the knots and the degree of the B-Spline basis determines the smoothness and the accuracy of the estimate. Figure 2.8 shows an example of B-spline bases of degree 1, 2, and 3 in the interval $[0, 1]$. Figure 2.9 illustrates a linear combination of cubic B-splines (B-spline basis of degree 3 shown the bottom panel of figure 2.8). The desired parameter γ of the base learner is now simply the weight of the spline basis functions and optional the placement of the knots.

GAMBoost uses *component-wise* smoothing. In every boosting step the base learner is a function of only one variable that means only the contribution of one single feature is considered in each step. As a consequence maximal M (the number of boosting steps) variables can contribute to the final model. Since the number of boosting steps is usually small compared to the number of variables p, GAMBoost performs an implicit feature selection (Tutz and Binder, 2006) and thus implements an embedded feature selection method.

[(11)] In Tutz and Binder (2006) GAMBoost is discussed with smoothing splines as well as with stumps. Since the R implementation of GAMBoost uses splines, these are described here.

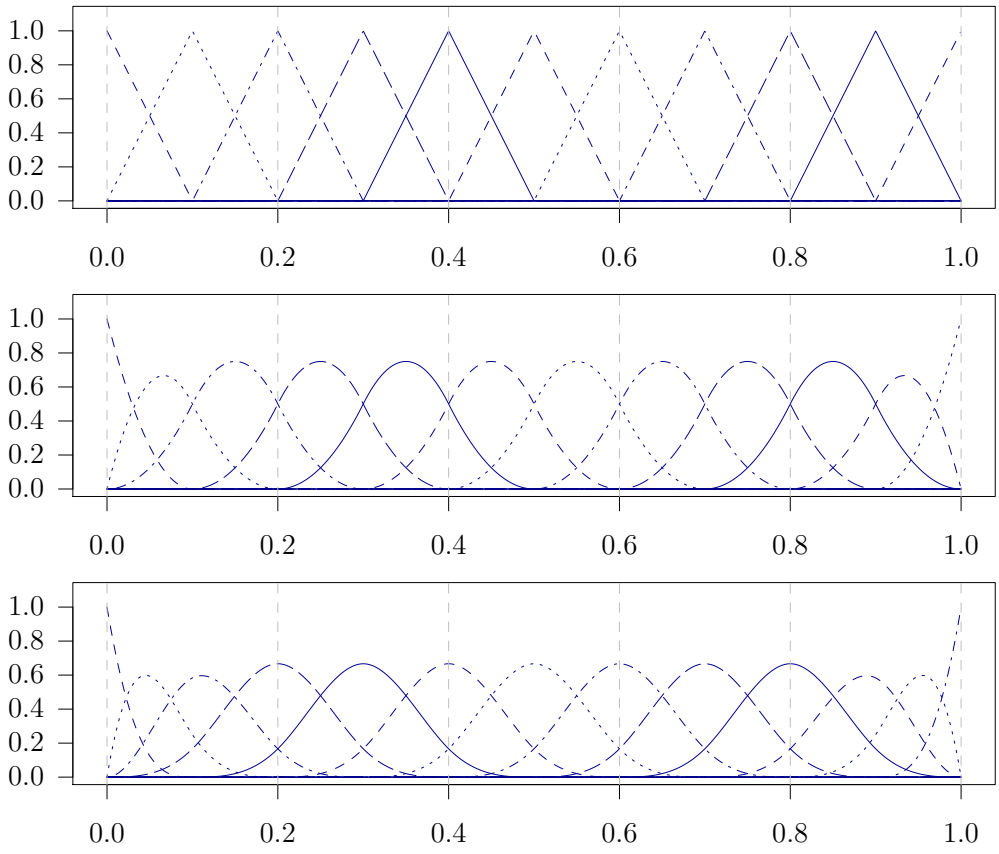


FIGURE 2.8. *B-spline basis functions defined in the interval* $[0, 1]$*. The knots are placed equidistant with a distance of* 0.1 *in the given interval. In the top panel basis functions of degree 1 (constant functions) can be seen. The middle panel shows quadratic splines (degree 2). The bottom panel shows cubic B-splines, the most often used B-spline basis.*

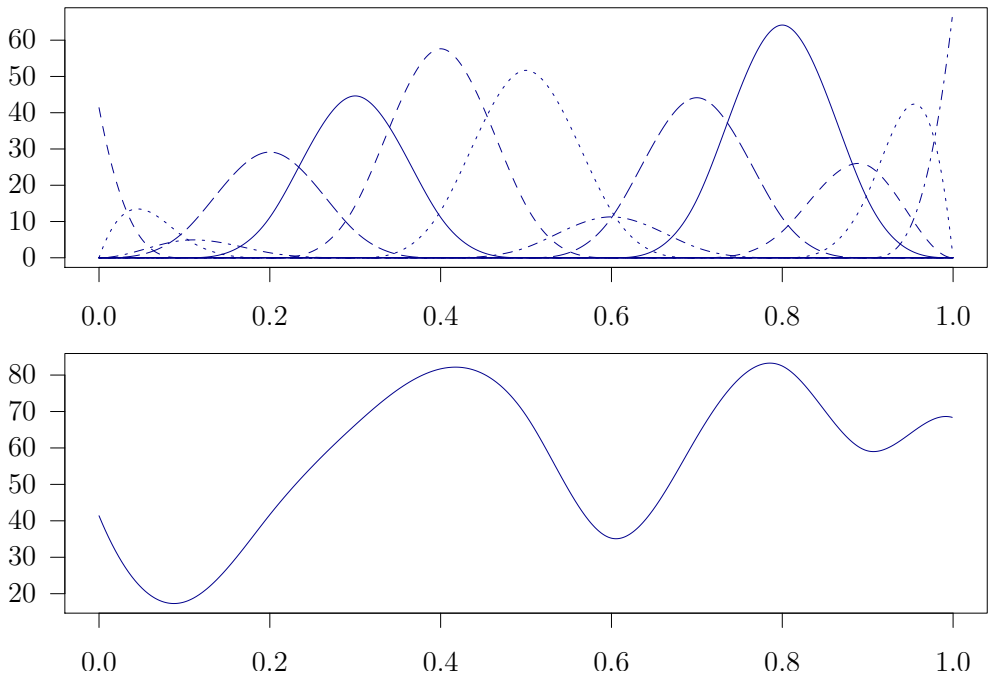


FIGURE 2.9. *B-spline basis expansion of cubic B-splines. The top panel shows different weighted spline basis functions and the lower panel the sum of these basis functions and therewith the linear combination of the B-spline basis.*

To avoid overfitting Tutz and Binder (2006) used penalized B-splines also called P-Splines (Marx and Eilers, 1998; Ruppert, 2002). Thereby many basis functions are used but in a penalized form. As a consequence the log-likelihood becomes a penalized log-likelihood

$$l_p(\gamma) = l(\gamma) - \frac{1}{2}\gamma^T \Delta \gamma \tag{2.50}$$

where Δ is the penalty matrix penalizing differences in the parameters corresponding to basis functions of adjacent knots. The more such differences are penalized the smoother the fit will be and overfitting becomes less likely. The degree of smoothing depends on a penalty parameter λ. Since the algorithm fits the model component-wise, the penalty parameter also determines the size of each boosting step and therewith the contribution of the variable chosen in each step. Indirectly, this parameter controls the number of boosting steps to perform and hence the maximal number of variables included in the model.

The likelihood based principle of GAMBoost can be extended to Cox models (cf. section 2.2.1). In this case the predictor η_i is the linear predictor involving

the variables $\mathbf{x_i}^T\boldsymbol{\beta}$ and instead of the log-likelihood the partial log-likelihood (2.41) is used for maximization. The desired parameter is the coefficient vector $\boldsymbol{\beta}$ and therewith an estimation of the hazard and the survivor function. By using component-wise boosting as deployed by GAMBoost, the resulting fit will be sparse that means most of the entries in the parameter estimation $\hat{\boldsymbol{\beta}}$ will be zero.

CoxBoost (Binder and Schumacher, 2008b) starts with a parameter estimation $\hat{\boldsymbol{\beta}} = \mathbf{0}$. In every boosting step m $(1 \leq m \leq M)$ and for each variable x_{ji} $(1 \leq j \leq p)$ a new linear predictor can be formulated

$$\eta_{ji}^{(m)} = \eta_i^{(m-1)} + x_{ji}\,\gamma_j^{(m)} \tag{2.51}$$

where an estimate for $\eta_i^{(m-1)}$ is given by the linear predictor from the previous boosting step

$$\hat{\eta_i}^{(m-1)} = \mathbf{x_i}^T\,\hat{\boldsymbol{\beta}}^{(m-1)} \tag{2.52}$$

Similar to GAMBoost, the maximization of a log-likelihood function is used to estimate $\eta_i^{(m)}$. Since the final model is a Cox model, instead of a true likelihood the partial likelihood (2.41) described in section 2.2.1 is used as a penalized partial log-likelihood

$$\begin{aligned} l_p\left(\gamma_j^{(m)}\right) = \sum_{i=1}^{n} \delta_i \Bigg[& \left(\hat{\eta_i}^{(m-1)} + x_{ji}\,\gamma_j^{(m)}\right) \\ & - \log\left(\sum_{l \in \mathcal{R}(t_i)} \exp\left(\hat{\eta_l}^{(m-1)} + x_{jl}\,\gamma_j^{(m)}\right)\right)\Bigg] \\ & + \frac{1}{2}\lambda_j^{(m)}\,\left(\gamma_j^{(m)}\right)^2 \end{aligned} \tag{2.53}$$

By using $\hat{\eta_i}^{(m-1)}$ as an offset the information from previous boosting steps is incorporated. As before, the penalty parameter $\lambda_j^{(m)} = \lambda$ determines the size of the boosting steps (and therewith the amount of the contribution of the current base learner and the current variable to the final model) and is typically the same for all boosting steps and variables. It has to be chosen preliminarily but only coarsely such that the resulting number of boosting steps M exceeds

around 50 steps (Binder et al., 2009; Binder and Schumacher, 2009). Otherwise the algorithm is too greedy and the resulting model too sparse.

Again, the Newton-Raphson algorithm is used to find estimates for $\gamma_j^{(m)}$ maximizing the partial log-likelihood. Hereby $U(\gamma) = \partial l(\gamma)/\partial\gamma$ is the score function, the first derivative of the unpenalized partial log-likelihood, and $I(\gamma) = \partial^2 l(\gamma)/\partial^2\gamma$ is the information matrix which is simply the negative Hessian of the unpenalized partial log-likelihood. Furthermore, let $U_j^{(m)} = U(0)$ and $I_j^{(m)} = I(0)$ denote the evaluations of U and I at parameter value $\gamma = 0$. Therewith, only one Newton-Raphson[(12)] is performed to get the estimate

$$\hat{\gamma_j}^{(m)} = \frac{U_j^{(m)}}{I_j^{(m)} + \lambda_j^{(m)}} \tag{2.54}$$

The variables with index j^* that maximizes the score statistic

$$j^* = \underset{j}{\operatorname{argmax}} \frac{\left(U_j^{(m)}\right)^2}{I_j^{(m)} + \lambda_j^{(m)}} \tag{2.55}$$

improves the fit the most in the current boosting step and the corresponding parameter estimate $\hat{\gamma}_j^{(m)}$ is used to update the overall parameter estimate $\hat{\boldsymbol{\beta}}$ as follows

$$\hat{\beta}_j^{(m)} = \begin{cases} \hat{\beta}_j^{(m-1)} + \hat{\gamma}_j^{(m)} & \text{if } j = j^* \\ \hat{\beta}_j^{(m-1)} \end{cases} \tag{2.56}$$

Note, in case the variable was picked for the first time the corresponding entry in $\hat{\boldsymbol{\beta}}$ is now changed from 0 to the current estimate and the variable is included in the final model. That illustrates the fact that after M boosting steps maximal M entries in $\hat{\boldsymbol{\beta}}$ can be unequal 0. Therefore, the number of boosting steps determines the number of variables included in the final model. Algorithm 3 summarizes CoxBoost.

[(12)] Binder and Schumacher (2009) noted that one step is enough since the same variable can be chosen in subsequent boosting steps adjusting the coefficient of this variable.

Algorithm 3 CoxBoost (Binder and Schumacher, 2008b)

1: initialize coefficient $\hat{\boldsymbol{\beta}}_0 = (0, \ldots, 0)$
2: **for** $m = 1 \rightarrow M$ **do**
3: **for** $j = 1 \rightarrow p$ **do**
4: fit candidate model for variable j and determine $\hat{\gamma}_j^{(m)}$ via Newton-Raphson
5: **end for**
6: determine winner model j^* and add $\hat{\gamma}_{j^*}^{(m)}$ to $\hat{\beta}_{j^*}^{(m-1)}$
7: update linear predictor $\hat{\eta}_i^{(m)} = \mathbf{x_i}^T \hat{\boldsymbol{\beta}}^{(m)}$
8: **end for**
9: final output is parameter estimation $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{(M)}$ from the Cox model

2.2.3 Pathboost

When building predictive models in the biomedical field, most often the variables are gene expression data. While in former years gene expression measurements were performed using microarrays nowadays there is a shift to next generation sequencing technologies. Either way, the features available for a predictive model are genes taking values which reflect the expression in the particular samples. Usually gene expression data are measured genome wide yielding several ten thousands to hundred of thousands of features. Boosting as described in the previous section is capable of building a predictive model with various outcomes (depending if the model is an additive model or a Cox model) while performing a feature selection at the same time. That way a panel of genes can be found with high predictive power for the particular problem.

However, for complex problems the performance of such models is usually unsatisfactory caused by the fact that the set of genes found by the algorithm is suboptimal. Like many other methods, boosting assumes independence of the features. Of course this is an assumption that does not hold true for gene expression data. Genes underlie complex regulatory mechanisms and are highly influence by each other. It it known that for many cancer types whole pathways are deregulated. Although sparse models resulting from feature selection are easier to interpret, it is most likely that only one candidate gene of such a pathway is picked for the model. Thus, it is hard to identify such deregulated pathways based on the feature list. However, information about common pathways and direct interactions are available in biological databases nowadays.

Several techniques have been proposed to included these meta-information into the model building process and feature selection Bellazzi and Zupan (2007); Chuang et al. (2007); Johannes et al. (2010); Porzelius et al. (2011a); Rapaport et al. (2007). Thereby, the overall goal is to improve the prediction performance and get gene sets which are robust and better interpretable.

Componentwise likelihood-based boosting is particular suited to include prior knowledge about feature relationships. The key lies in the iterative nature of the method and the flexible penalty structure. Binder and Schumacher (2009) proposed *PathBoost*, an extension to GAMBoost and CoxBoost, which is briefly explained in the following using the example of CoxBoost.

Such prior biological knowledge can be represented as graph G where the knots are the genes and the edges represents gene-gene interactions. These interactions do not need to be actual regulatory interactions observable on the protein level but can also represent e.g. common pathways or other similarities. If a strength can be assigned to such a gene-gene interaction the corresponding adjacency matrix contains not only 0 and 1 but entries $g_{ij} \in [0,1]$[(13)].

The feature selection must now not only consider the feature and its predictive power bit also the connectivity of the feature in this graph. The key to include such knowledge into CoxBoost lies in the penalty term λ in the penalized partial log-likelihood (2.53). Instead of assuming a fixed penalty term for all variables and all boosting steps the penalty will be adopted during the fitting process. Figure 2.10 illustrates the principle of PathBoost by means of a little toy example with 4 genes.

Adapting the penalties during the fitting process requires two update rules. At first, the penalty of the variable picked in the current boosting step $\lambda_{j^*}^{(m)}$ is increased making it less likely that the same variable will be picked again in following boosting steps. For following boosting steps $l > m$ the penalty becomes[(14)]

$$\lambda_{j^*}^{(m_k+1)} = \frac{I_{j^*}^{(m_k+1)}}{1 - \left(1 - \frac{I_{j^*}^{(m_k)}}{I_{j^*}^{(m_k)} + \lambda_{j^*}^{(m_k)}}\right)^{c_f}} - I_{j^*}^{(m_k+1)} \tag{2.57}$$

[(13)] Without a loss of generality it can be assumed that edge weights are scaled to the interval $[0,1]$.

[(14)] In Binder and Schumacher (2009) a linear increase has been described. The here mentioned sigmoid penalty increase is the default in the R implementation of PathBoost and was used in this work.

Here, m_k is the boosting step where the feature was picked the kth time and got a penalty update. The penalty update is only performed for features with at least one connection. The penalty of features without a single connection remains unchanged when they are picked.

The second update rule deals with the penalties of connected features and is more complex. To account for the loss of variability caused by the increase of the penalty of the selected variable the penalty of connected features j^+ is reduced in following boosting steps

$$\lambda_{j^+}^{(m+1)} = \frac{(1-\pi_{j^+}^{(m)})\, I_{j^+}^{(m+1)}\, g_{j^*j^+}}{(1-c_f)\frac{(1-\pi_{j^*}^{(m)})\, I_{j^*}^{(m+1)}\, g_{j^*j^+}}{I_{j^*}^{(m+1)}+\lambda_{j^*}^{m}} + \frac{(1-\pi_{j^+}^{(m)})\, I_{j^+}^{(m+1)}}{I_{j^+}^{(m+1)}+\lambda_{j^+}^{m}}} - I_{j^+}^{(m+1)}$$

$\pi^{(m)}$ is the approximated fraction of the Maximum Likelihood estimate (obtained via non-boosting estimation) that has been realized for the feature in the mth boosting step (cf. Binder and Schumacher, 2009 for details). The degree of the penalty decrease is influenced by the measure of uncertainty $0 < g_{j^*j^+} \leq 1$ for the edge between feature j^* and j^+ in the graph G (the graph representing the biological knowledge).

The decrease of the penalty of connected features increases the probability of picking these features in future steps. Thus, it is therefore more likely to pick features connected to features already included in the model.

The step-size modification factor c_f takes values between 0 and 1 and controls the influence of connection graph on the feature selection. For $c_f = 1$ no connection information would be considered. This would result in a standard CoxBoost fit. Small values of c_f increase the influence of the prior knowledge.

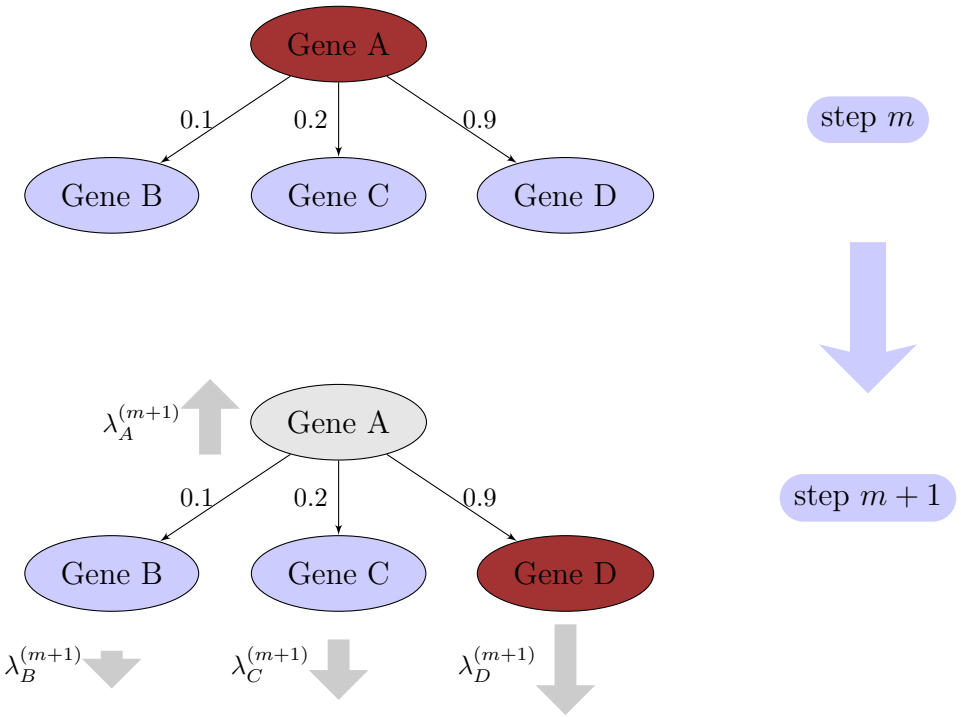


FIGURE 2.10. *This example illustrates the basic principle of the PathBoost extension based on a network with 4 genes. In step m gene A is chosen by the boosting algorithm (either GAMBoost or CoxBoost). As a result the penalty of gene A $\lambda_A^{(m+1)}$ is increases in the next step. On the other hand the penalty of the three adjacent genes B, C, and D is decreased according to the weight of the edge from gene A. In a biological network this might be the strength of the interaction or the number of common pathways (scaled to the interval $[0,1]$). Note that the edges do not necessarily need a weight, the simple case where the adjacency matrix contains only 0 and 1 is also allowed. In this example the penalty of gene D $\lambda_D^{(m+1)}$ is decreased the most and thus it is picked in step $m+1$.*

2.3 Other Methods Suited for Time-to-event Data

Here two other methods suited for time-to-event data are shortly introduced. These two methods are often used competitors of boosting approaches when embedded feature selection is needed. They both are suited for time-to-event data and were used for comparison in this work.

2.3.1 Regularized Regression Methods

Lasso (Tibshirani, 1996) was proposed as a shrinkage regression models (Hastie et al., 2009; chap. 3) implementing an embedded feature selection. The regression coefficients are penalized with anL_1 penalty term

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left(l(\boldsymbol{\beta}) - \alpha ||\boldsymbol{\beta}||_1 \right) \tag{2.58}$$

with a likelihood function $l(\boldsymbol{\beta})$ suited for the outcome. While Ridge regression uses an L_2 (quadratic) penalty term the use of the absolute penalty forces most of the entries in $\hat{\boldsymbol{\beta}}$ to be exactly 0 and therewith performs an embedded features selection. Ridge regression performs a parameter shrinking leaving most of them > 0. If a feature selection is needed a cutoff for the parameters needs to be defined. On the other hand, if many variables with small effects can be assumed Ridge regression might be a better choice. As a trade-off the method yields large models that are harder to interpret.

Originally, Tibshirani (1996) proposed quadratic programming to solve (2.58) for linear regression models. Tibshirani (1997) extended the idea of Lasso to Cox proportional hazard models still based on quadratic programming. Since the solution for Cox models is much more computationally intensive, Goeman (2010) proposed a solution of the Lasso estimation based on gradient ascent optimization. The associated R implementation (Goeman, 2011) was used for comparison.

2.3.2 Random survival forests

The second method is based on decision trees. Here, the sample space is divided into smaller subspaces based on single variables. The dividing process

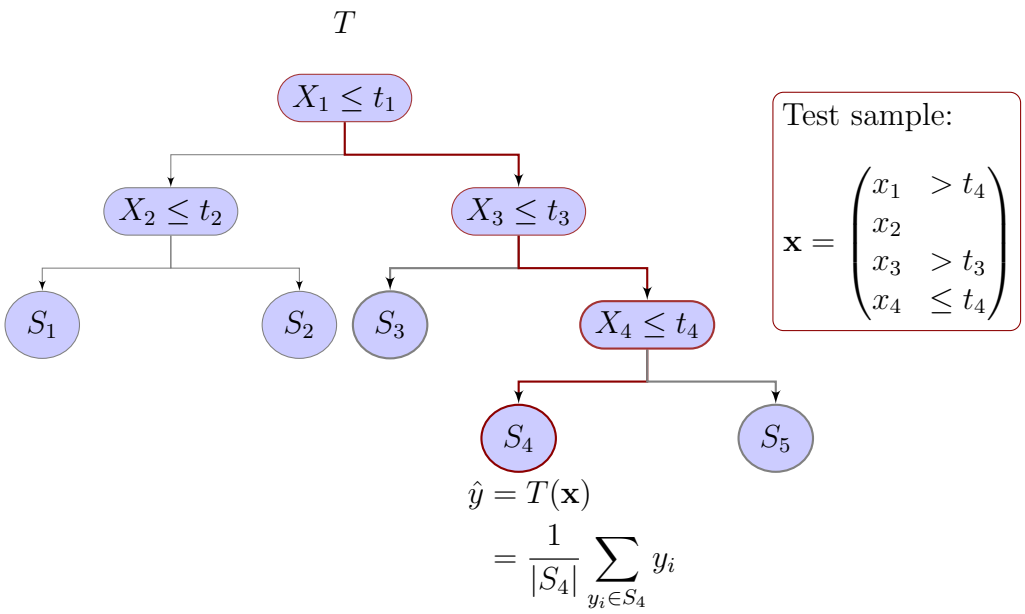


FIGURE 2.11. *Example for a decision tree (adopted from Hastie et al., 2009). The regression tree T divides the space of input samples into several subspaces $S_1, \ldots, S_5$ based on the variables $X_1, \ldots, X_4$ and assigned split points $t_1, \ldots, t_4$ in a hierarchical manner. A new sample with feature vector $\mathbf{x} = (x_1, \ldots, x_4)^T$ can now be assigned to one of the subspaces. The prediction of the tree for this sample is then simple the average of the training samples in the given subspace, in this example S_4. For a classification tree the prediction $\hat{C}(\mathbf{x})$ would be simply the majority vote of the training samples in S_4. Note, not all entries in $\mathbf{x}$ influence the decision. In this example the value of x_2 is irrelevant for the prediction.*

is hierarchical and thus can be illustrated as tree. A formal definition can be found in Alpaydin (2010):

Definition 2. Decision trees
A decision tree is a hierarchical model for supervised learning where the local region is identified in a sequence of recursive splits in a smaller number of steps. The tree is composed of internal decision nodes and terminal leaf nodes.

Each internal decision node implements a test function based on one variable (univariate tree) or several variables (multivariate tree). Such a tree can then be used for prediction. The test sample is assigned to a terminal leaf node (and therewith a subset of the training samples) based on the test functions of the internal decision nodes. The prediction is simply a majority vote over those training samples (classification tree) or the average (regression tree). Figure 2.11 illustrates the basic principle of a decision tree.

The structure of the tree is not fixed a priori but has to be learned together with the decision functions of the internal nodes. Several approaches have been proposed to learn such a tree based on training data, e.g. the CART (Classification and Regression Trees) algorithm (Breiman et al., 1984) and C4.5 (J. R. Quinlan, 1993).

After learning the structure and internal test functions a tree is a simple and easy to interpret prediction model learned from the data. The simplicity comes with a price. Decision trees usually have a high variance based on an inherent instability. Slight changes in the data could cause a complete different series of splits and an error in the top of the hierarchy is propagated through the whole structure.

A solution for this problem was found by Breiman (2001). Instead of learning one single tree, B trees are trained and their predictions are averaged. Similar to Bagging (bootstrap aggregation, Breiman, 1996) the single trees are trained on bootstrap samples of the original training, introducing a randomization to the data, thus the name of the method: *Random forests*. Additionally, during the tree growing process before choosing a split point, $m \leq p$ predictor variables are chosen as candidates for the split. The resulting trees are (for large B) uncorrelated and reduce the variance of the overall prediction model.

The parameters B and m have to be determined a priori. After fitting the trees the prediction of a training sample with feature vector $\mathbf{x}$ is given by

$$\hat{y} = \hat{f}_{rf}^{B}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} T(\mathbf{x}, \gamma_b) \tag{2.59}$$

for a regression problem and by

$$\hat{g} = \hat{C}_{rf}^{B} = \text{majority vote}\{\hat{C}_b(\mathbf{x})\}_1^B \tag{2.60}$$

for a classification problem. Thereby, $\hat{C}_b(\mathbf{x})$ is the class prediction of the b-th decision tree.

Random forests perform remarkably well for most situations with little tuning efforts (see Hastie et al., 2009; chap. 15 for a comprehensive overview and comparisons to boosting). Additionally it performs an embedded feature

selection and can deal with variables on different scales, making it a good choice for high-dimensional heterogenous data sets.

Ishwaran et al. (2008) proposed an extension of Random forests suited for right censored survival data called *Random survival forests* (RSF). Following the principles of Random forests a collection of binary decisions trees is built from bootstrap samples. For the internal nodes of each tree a random set of m variables is chosen. The variable with corresponding split point that maximises the survival difference between the two resulting daughter nodes is used for the split. A terminal node is created when no more split can be performed e.g. a specified number of unique events is reached. The authors also provide an R implementation of their method (Ishwaran and Kogalur, 2007) which was used in this work.

2.4 Model Assessment and Selection

Model assessment and, if several models are available, the choice which model is best suited for the given data are one of the fundamental problems in statistical learning. Several measures can be considered when judging the quality of a certain model or learning algorithm.

For a prediction model as introduced in the previous sections the most important measure is the *generalization performance* as a measure for its prediction capabilities on yet unknown data. The assessments of this performance is fundamental since it not only guides the choice of the learning algorithm but allows the evaluation of the final model and therewith the prediction results.

Per definition the generalization performance is unknown during the learning process. The next section introduces some terms and concepts leading to an estimate of this performance called .632 bootstrap estimate. Several other estimates exists and the interested reader is referred to Hastie et al. (2009; chap. 7) for more details and comparison.

2.4.1 Introduction to Test- and Training Error

Let $\mathcal{T}$ denote the training set used for fitting the model that means the set of samples where the variables $\mathbf{x}_i$ and the outcome y_i is known. The *training error* is given by

$$\overline{err} = \sum_{i}^{n} l(y_i, \hat{f}(\mathbf{x})) \tag{2.61}$$

with a given loss function l. Since the prediction error is calculated at the same data as the model was fitted $\overline{err}$ usually overestimates the performance of the model and the underlying learning algorithm. Thus, it is a poor estimate for the generalization performance. A more realistic estimate is given by the so called *test error* or *generalization error*, the prediction error on an independent test sample

$$Err_{\mathcal{T}} = E\left[l(y_j, \hat{f}(\mathbf{x}_j) \,|\, \mathcal{T}\right] \qquad (y_i, \mathbf{x}_j) \notin \mathcal{T} \tag{2.62}$$

which needs a test set on the side. Figure 2.12 shows a comparison of the training- and test error as function of the model complexity.

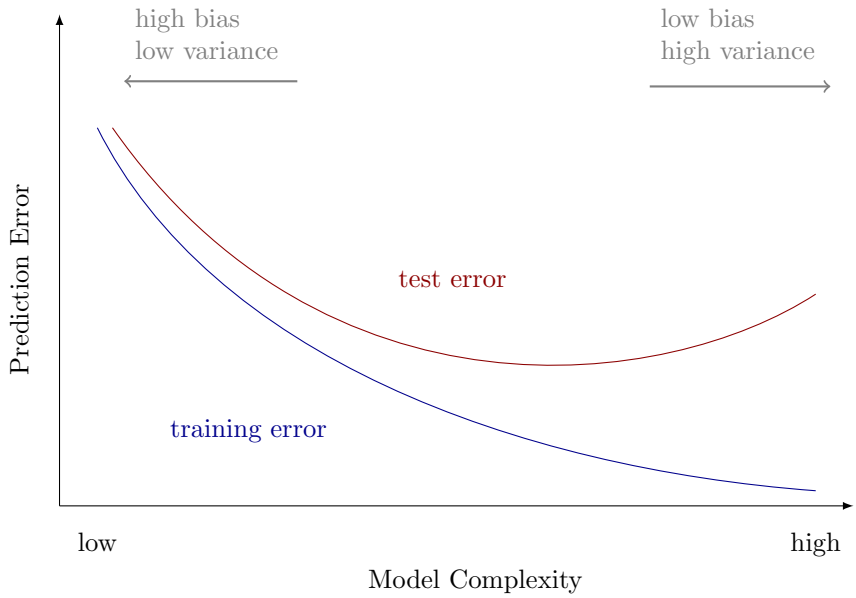


FIGURE 2.12. *The figure illustrates the training error (blue line) and the test error (red lines) as a function of the mode complexity. The training error (prediction error on the training set) decreases continuously with increasing model complexity and can even drop to zero if the model gets complex enough. The test error (prediction error on the test set) on the other side increases after a first drop. At this point the model starts to overfit and learns noise instead th functional relationship between the outcome y and the variables* $\mathbf{x}$ *(adapted from Hastie et al., 2009).*

Clearly, the training error $\overline{err}$ is biased downward compared to the test error. It can even drop down to 0 if the model gets complex enough. Such a model is usually too adapted to the training set $\mathcal{T}$ and is therewith *overfitted*. Therefore the test error $Err_{\mathcal{T}}$ for such a model is high. Such a model does not only reflect the underlying functional relationship between the outcome Y and the variables X but includes additional noise. Hence, the generalization performance is poor.

Equation (2.62) shows that the test error is still dependent on a fixed training set $\mathcal{T}$. The *expected test error* can now be defined as

$$\begin{aligned} Err &= E\left[l(y, \hat{f}\mathbf{x})\right] \\ &= E\left[Err_{\mathcal{T}}\right] \end{aligned} \tag{2.63}$$

Note, the expected test error integrates over all possible training sets and is therefore independent. It is the desired error to judge the generalization performance of a particular model and hence several strategies have been proposed to estimate Err. With such estimates the two problems mentioned at the beginning of this section can be addressed

1. **Model Selection**
   If a learning algorithm is parameterized with a tuning parameter (e.g. the number of boosting steps M) the optimal model can be determined based on the lowest Err.

2. **Model Assessment**
   If a final model has been fitted the performance of this model is given by an estimate of Err.

2.4.2 K-fold Cross-Validation

The first task, model selection, was performed using a theoretical concept called K-fold cross-validation (Allen, 1974; Kohavi, 1995; M. Stone, 1974). Thereby, the available data are split into K equal sized subsets. Different models (with different model parameters) are trained on the remaining subsets and tested in the chosen subset. This is done for every of the K subsets. After all subsets

have been used as test set every sample in the data set was used for prediction once. The resulting cross-validation estimation of the Err is given by

$$\widehat{Err}_{CV} = \sum_{i=1}^{n} l(y_i, \hat{f}^{-\kappa(i)}(\mathbf{x}_i) \tag{2.64}$$

where $\kappa : \{1 \ldots, n\} \mapsto \{1, \ldots, K\}$ is an indexing function mapping a sample to the subset it belongs to. Hence $\hat{f}^{-\kappa(i)}(\mathbf{x}_i)$ is the function, fitted with the subset with $(y_i, \mathbf{x}_i$ removed. The model with a parameter set minimizing (2.64) is chosen. Typical choices for K are 5 or 10. In the extreme case $K = n$ a special form, the so called leave-one-out cross-validation is used.

Cross-validation can be also used for the task of model assessment. In this case the available data have to be split in a nested fashion. The so called outer cross-validation is used for assessment of the best model. The best model can be found by splitting the training subsets again and performing the inner cross-validation.

2.4.3 Bootstrap and the .632 Error Estimator

For smaller K the training sets become small compared to the whole data set. As a consequence cross-validation overestimates the generalization error and underestimates the performance of the model. For large K the variance of the estimation becomes higher since the training sets become more and more similar to each other[15]. Considerable sizes of K are 5 and 10 (Breiman and Spector, 1992; Kohavi, 1995). The leave-one out cross-validation is considered near unbiased and is therefore often used despite the problem of high variance. However for large sample sets the computational overhead of leave-one-out compared to the 5- or 10-fold procedure is considerable.

Because of the high variance of the cross-validation estimator other methods for estimating the prediction performance have been proposed. Bootstrap in general is a method to assess the accuracy of an estimator. The basic idea is to draw random samples with replacement from the original data, each of the size of the original set. A given estimator is computed for all *bootstrap samples*.

[15] In the extreme case $K = n$, the leave-one-out cross-validation, the several training sets differ only by one sample.

Besides the estimator itself the variance of the estimation can be determined assuming enough bootstrap samples were drawn.

Estimating the expected prediction error Err is carried out the same way. However to determine a realistic estimate of Err the observations in one bootstrap sample cannot be used for both, fitting the model and predicting the outcome. Also prediction of the original training data would lead to an underestimation of the generalization error since a bootstrap sample overlaps with the training data. Similar to a cross-validation procedure a split in training and test data is needed.

Since a bootstrap sample is drawn with replacement from the original data set, some observations will picked more than once and hence other observations are omitted. More precisely the probability to pick an observation i in bootstrap sample b is

$$\begin{aligned} P(i \in b) &= 1 - \left(1 - \frac{1}{n}\right)^n \\ &\approx 1 - e^{-1} \\ &= 0.632 \end{aligned} \tag{2.65}$$

As a consequence approximately 63.2 % of the original data are in one bootstrap sample. By using these observations for fitting the model, the remaining data, the *out-of-bag* data can be used for testing. The bootstrap estimation of the test error, the *leave-one-out bootstrap* estimate (Efron, 1979), is then given by[16]

$$\widehat{Err}_{boot} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{|C^{-b}|}\sum_{i \in C^{-b}} l(y_i, \hat{f}^{*b}(\mathbf{x}_i) \tag{2.66}$$

Here, $\hat{f}^{*b}(\mathbf{x}_i)$ denotes the prediction at $\mathbf{x}_i$ of the model fitted using the b-th bootstrap sample. The set C^{-b} contains the set of indices i of observations belonging to the b-th test set (out-of-bag data of the b-th bootstrap sample). Since only 63 % of the available data are used for fitting the single models (2.66) suffers from the same bias as the cross-validation estimate $\widehat{Err}_{CV}$ for smaller K. It overestimates the prediction error.

[16] Efron and Tibshirani (1997) pointed out that the leave-one-out bootstrap estimator can be seen as a smoothed version of the leave-one-out cross-validation estimator. The smoothing results in reduced variance of the estimation.

A solution for this problem was found by Efron (1983) and is based on (2.65). The so called .632 *bootstrap estimator* alleviates the training-size bias by averaging over the test- and training error

$$\widehat{Err}_{.632} = .368 \cdot \overline{err} + .632 \cdot \widehat{Err}_{boot} \tag{2.67}$$

where $\overline{err}$ is training error on the original data. The interested reader is referred to Efron and Gong (1983) for a detailed derivation of (2.67). In addtion the single bootstrap test errors instead of the averaged one can be used in (2.67). This allows the estimation of the variance of $\widehat{Err}_{.632}$ and thus gives a hint how stable the prediction performance of a certain learning algorithm is.

In settings where a high overfitting is possible (e.g. settings where $\overline{err} = 0$) the .632 estimator is biased downward. Efron and Tibshirani (1997) proposed a correction by taking into account the amount of overfitting. The resulting .632+ estimator corrects the downward bias. The amount of overfitting is given by the no-information error rate. This rate can be estimated by for example shuffling the outcome y_i. Based on the no-information error rate a relative overfitting rate $\hat{R}$ can be defined. It ranges from 0 indicating no overfitting ($\overline{err} = \widehat{Err}_{boot}$) to 1 where the overfitting reaches the no-information value (for details cf. Efron and Tibshirani, 1997). The .632+ estimator is given by

$$\widehat{Err}_{.632+} = (1 - \hat{w}) \cdot \overline{err} + \hat{w} \cdot \widehat{Err}_{boot} \tag{2.68}$$

with weights

$$\hat{w} = \frac{0.632}{1 - 0.368\,\hat{R}} \tag{2.69}$$

$\widehat{Err}_{.632+}$ ranges from $\widehat{Err}_{.632}$ in case no overfitting occurs to the leave-one out bootstrap error $\overline{err} = \widehat{Err}_{boot}$ when there is heavy overfitting. It can be seen as compromise between the training error and the bootstrap estimator depending on the degree of overfitting. Note that the calculation of $\widehat{Err}_{.632+}$ is computationally more expensive since the no-information rate has to be estimated.

2.4.4 Prediction Error for Time-to-event Data

The estimates of the generalization- or test error $\widehat{Err}$ introduced in the last section need a loss function $l(y_i, \hat{f}(\mathbf{x}_i))$ to measure the deviance between the outcome y_i and the model prediction based on the variables $\mathbf{x}_i$. In a classification or simple regression setting one could simply choose the misclassification rate, the exponential loss, or the Huber loss (cf. section 2.1). For time-to-event data the situation is more complex. After fitting a Cox model the risk prediction (2.43) is a function of time describing the predicted probability of still being event free at time t given a set of variables. Let $Y_i(t)$ denote true event state of individual i at time t

$$Y_i(t) = \begin{cases} 1 & T_i > t \\ 0 & \text{otherwise} \end{cases} \tag{2.70}$$

Here T_i denotes the true (and possibly unknown) event time of individual i. The given time t_i is the minimum of T_i and the censoring time C_i.

To assess the quality of these predictions the Brier score

$$BS(t) = E\left[\frac{1}{n}\sum_{i=1}^{n}\left(Y_i(t) - \hat{r}(t|\mathbf{x_i})\right)^2\right] \tag{2.71}$$

can be used (Graf et al., 1999), describing the average discrepancy between the event states and the model predictions. Due to censoring, inverse probability of censoring weights (Gerds and Schumacher, 2006; Graf et al., 1999) have to be used to obtain consistent estimates of (2.71). By tracking this empirical version of the Brier over time, prediction error curve estimates are obtained:

$$PEC(t) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i(t) - \hat{r}(t|\mathbf{x_i})\right)^2 W(t,i) \tag{2.72}$$

with weights

$$W(t, \hat{G}, i) = \frac{I(t_i \leq t, \delta_i = 1)}{\hat{G}(t_i)} + \frac{I(t_i > t)}{\hat{G}(t)} \tag{2.73}$$

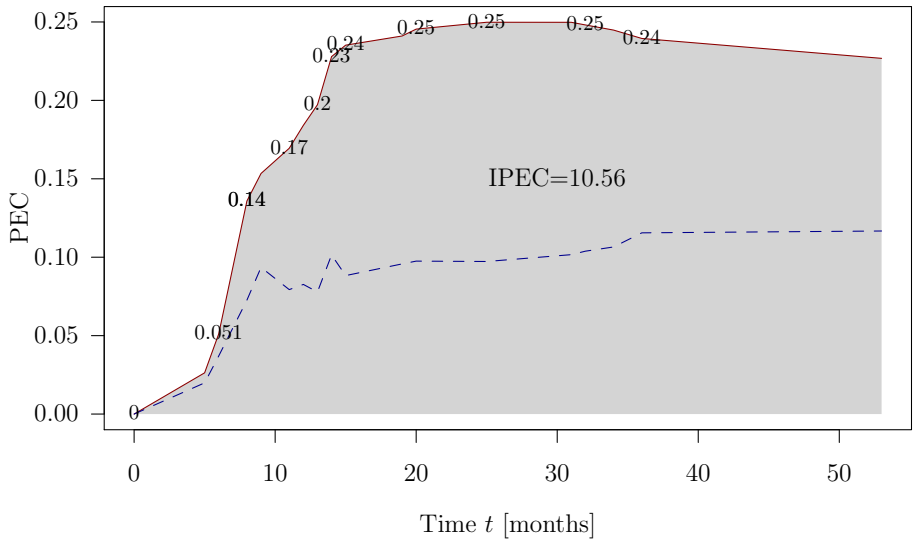


FIGURE 2.13. *Example of the prediction error curve (PEC) and the integrated prediction error curve (IPEC) of the glioma dataset. The solid red curve shows the PEC based on a Kaplan-Meier risk prediction estimate. The area under the curve is the corresponding IPEC. The numbers on the curve give the error estimated at the event times of the patients (for a better readability only every second event time point was used). Note that the KM estimate of the Survivor function do no take into account any variables. As a reference the PEC from a Cox model including variables Group, Age, and Histology is given (blue dashed line). The benefit of the additional variables in the Cox model compared with the non-parametric estimation of the Kaplan Meier is obvious.*

where $\hat{G}(t)$ denotes a consistent estimate of the conditional probability of being censored at time t. In this case the Kaplan-Meier estimate can be used (Graf et al., 1999). By integration over time the integrated prediction error curve (IPEC) is obtained. Figure 2.13 shows an example of the PEC based on the glioma data set (cf. table 2.1).

If the risk prediction model in (2.72) is trained using all data, (2.72) denotes the training error or apparent error $\overline{err}(t, \hat{r})$. Now, a bootstrap estimate of the test error can be formulated

$$\widehat{Err}_{boot}(t, \hat{r}) = \frac{1}{B} \sum_{b=1}^{B} \frac{1}{C^{-b}} \sum_{i \in C^{-b}} \left(Y_i(t) - \hat{r}_b(t|\mathbf{x_i}) \right)^2 W(t, i) \tag{2.74}$$

where $\hat{r}_b$ denotes the risk prediction of the model trained on the b-th bootstrap sample. The definition of the .632 estimator is now straight forward. The .632+ estimator is given by

$$\widehat{Err}_{.632+} = (1 - w(t)) \cdot \overline{err}(t, \hat{r}) \; + \; w(t) \cdot \widehat{Err}_{boot}(t, \hat{r}) \tag{2.75}$$

where weights $w(t)$ are adapted for right-censored time-to-event data according to Gerds and Schumacher (2007).

2.5 MicroRNA Target Predictions

miRNAs are key regulators of gene expression (cf. section 1.1.3.2). Hence, the knowledge about potential targets is the key for the understanding and analysis of miRNA data. Since the experimental validation of a miRNA-mRNA interaction is time-consuming, in-silico predictions of miRNA targets are an important source of knowledge.

A variety of target prediction algorithms and databases exist (cf. Panagiotis et al., 2009 for an overview). Two approaches were considered in this work. The first is the MicroCosm target prediction database[(17)] (Griffiths-Jones et al., 2006, 2008). The MicroCosm target prediction pipeline is based on the miRanda algorithm (Enright et al., 2003; John et al., 2004). miRanda is a three-phase method. At first, dynamic programming alignment is used to find matches between miRNAs and the 3' UTRs of potential targets. A weighted scheme is used to reward matches at the 5' end of the miRNA. Alignments with more than one mismatch in the seed region of the miRNA are discarded. The result is a score for a sequence match between a miRNA and a potential target gene. Afterwards the free energy ΔG for each miRNA-mRNA match found in the first phase is computed using the Vienna algorithm (Wuchty et al., 1999). The third phase is mandatory and gives information how conserved a target site is across different species.

Additionally a p-value is calculated based on an extreme value distribution as described in Rehmsmeier et al. (2004). The p-value is solely based on the miRanda scores without taking into account thermal stability or cross-species

[(17)] formerly miRBase Targets, version 5

conservation. It it a measure for the significance of a certain miRNA-mRNA pair. Only matches with a p-value lower then 0.05 are reported in the target prediction database. Although the thermal stability and conservation of target sites are not considered, miRNA-mRNA pairs with a very low p-value tend to have conserved target sites and a low free energy.

For this work the target predictions were downloaded as miRNA-mRNA pairs together with the assigned p-value (transcripts were given as Ensembl[18] transcript identifiers).

A second target prediction database has been used for comparison. The TargetScan predictions (version 5.5, Friedman et al., 2009; Lewis et al., 2005). TargetScan does not use thermodynamic stability but rely solely on matches between miRNA seeds and highly conserved regions of UTRs (3' UTR of potential target mRNAs). Since only k-mers (6,7, or 8) of the seed region of a miRNA are considered the predictions are valid for whole miRNA families. The algorithm uses alignments of 3' UTRs of different species and searches for well conserved matches to the seed region of miRNAs. As mentioned in section 1.1.3.2 miRNA target sites are found to be highly conserved across multiple species which is another hint for the importance of miRNA mediated gene expression regulation.

Since mRNA sequences can be conserved for many reasons beyond miRNA targeting the conservation of a match to a miRNA seed is not sufficient. Additionally, a background conservation has to be estimated. For example, a well conserved targets site within a rapidly evolving UTR is far more likely conserved due to miRNA targeting than a site in a highly conserved UTR and therewith a more promising candidate.

Similar to MicroCosm, TargetScan provides a p-value as a measure of certainty for miRNA-mRNA match. The TargetScan database contains P_{CT} which is approximately equal to $(\frac{S}{B} - 1)/\frac{S}{B}$. $\frac{S}{B}$ is an estimated signal-to-background ratio calculated using controls of equal size as the miRNA target sites (cf. Friedman et al., 2009 for details). Thus, P_{CT} is the Bayesian estimate of the probability that a specific target site is conserved due to miRNA targeting and not by chance. $1 - P_{CT}$ is an estimate of the false discovery rate (FDR) and

[18] The Ensembl project (www.ensembl.org, Flicek et al., 2012) provides genomic information with a focus on the human genome

can be used to assess the biological importance of a particular miRNA-mRNA pair.

TargetScan predictions were downloaded as pairs of miRNA families and mRNAs together with the P_{CT} values (transcripts were given as RefSeq[19] transcript identifiers). The miRNAs were matched to their families such that at the end every miRNA in one family was assigned to the same targets.

2.6 Data Set

A prostate cancer data set from Taylor et al. (2010) was used in this study. Raw expression data from Affymetrix Human Exon 1.0 ST arrays were obtained from the NCBI GEO data repository[20] (GEO accession number GSE21034) comprising 131 samples of tumor patients. Furthermore, miRNA expression data from the Agilent microRNA V2 were downloaded (GEO accession number GSE21036) including 113 samples of tumor patients.

2.6.1 Data Preprocessing

Preprocessing and especially normalization is a crucial part in the microarray analysis (cf. section 1.1.4 for an introduction to microarrays). A typical preprocessing consists of 3 steps

1. background correction
2. summarization of probe intensities
3. normalization

A laser is used to create the image of the array. The first step, the background correction, is used to correct for noise caused by reflections of the array (cf. Ritchie et al., 2007 for an overview).

Normalization is used to remove any systematic effects arising from the microarray technology rather than from the biological experiment. In a first

[19] The Reference Sequence database (RefSeq) is a collection of genomic, transcript, and protein records (Pruitt et al., 2012).
[20] The Gene Expression Omnibus (GEO, Barrett et al., 2011) is a public repository for high-throughput microarray data.

step, within-array normalization removes local effects from the probe intensities that are based on different hybridization efficiency across the array. Especially for spotted two-color arrays, this was an important step in the preprocessing (cf. Smyth and Speed, 2003; Yang et al., 2002 for an overview). On the other side, modern one-color arrays and hybridization protocols have reached a quality level that obviates the need for within-array normalization.

The next step is the summarization of the probe intensities. Modern gene expression microarrays, like the Affymetrix Human Exon 1.0 ST used in this data set, contain several probes for one gene. After background correction a signal intensity is associated to every spot and therewith every probe on the microarray. The aim of the summarization step is to calculate one gene expression value based on the single probe intensities.

Between-array normalization is intended to achieve consistency and therewith comparability between the arrays of one experiment. It eliminates variation of non-biological origin between the arrays e.g. differences in the RNA extraction efficiency.

The gene expression data for the prostate cancer data set used in this work were derived from the raw data files using Robust Multichip Average (RMA, Irizarry, 2003) implemented in the Affymetrix Power Tools. RMA realizes a background correction via a linear model and a robust probe summarization for Affymetrix microarrays. Afterwards the data were normalized using quantile normalization as proposed by Bolstad et al. (2003). Quantile normalization is an often used normalization method that is suited not only for gene expression data. The method adapts the distributions of the expression values of each array by equalising their quantiles, hence the name quantile normalization.

Raw data files from miRNA expression data were analyzed using the R-package limma (Smyth, 2005). Each miRNA was represented by 16 probes (replicates) on the array. The replicates were summarized using the sample-wise median. Again, quantile normalization was used to remove inter-array variation.

At the end only tumor samples with gene expression as well as miRNA expression data were used yielding a data matrix with 98 tumor samples, 17881 transcripts (mRNAs), and 723 miRNAs.

2.6.2 Biochemical Relapse Status

Clinical parameters of the patients samples were downloaded from the supplemental material of Taylor et al. (2010). The time to the biochemical relapse and the censoring status for 98 cancer patients were available. Of these 98 patients 18 suffered a relapse and 80 were censored.

Chapter 3

Results and Discussion

3.1 Graph-Based Fusion of miRNA and mRNA Expression Data

Due to their role as posttranscriptional regulators of around 30 % of the human genome and their involvement in crucial cellular processes such as cell proliferation, differentiation and apoptosis, miRNAs were subject of numerous studies in the past years. Large genome wide screening studies as well as functional studies revealed an involvement of miRNAs in the development and progression of cancer in general (Garzon et al., 2006; Groce, 2009; Lu et al., 2005) and particularly in prostate cancer (Brase et al., 2011; Coppola et al., 2010).

Since miRNAs are shorter than mRNAs they are more stable and in general more resistant against degradation processes than the longer mRNAs. Consequently, miRNA expression is measurable even in serum (Brase et al., 2010) and paraffin-embedded tissues where mRNA expression is hardly detectable. Therefore, miRNAs are proper candidates for biomarkers and indeed several studies were conducted to identify miRNAs with diagnostic and prognostic potential (Brase et al., 2010).

Genome wide measurements of mRNA expression has been a common method to identify patterns and potential biomarkers in biomedical research, especially cancer research. In fact, panels of mRNA markers gained from genome wide studies (Paik et al., 2004; van 't Veer et al., 2002; Wang et al.,

2005) are now used in clinical routine to aid clinician's treatment decisions in breast cancer. However, for prostate cancer the prognostic potential from mRNA markers remains unsatisfactory (Tosoian and Loeb, 2010).

The regulatory nature of the miRNAs together with nowadays abilities of genome wide miRNA expression studies makes the integration of mRNA and miRNA expression data a logical step towards the understanding of posttranscriptional regulations. Indeed, several studies have combined gene and miRNA expression data (Cho et al., 2011; Nymark et al., 2011) or gene expression data with miRNA target predictions (Cheng and Li, 2008) to infer new miRNA regulation activities. In addition, several tools have been developed to integrate such data (Huang et al., 2011; Sales et al., 2010). In most cases, correlations between mRNA and miRNA expression profiles gained from matched samples and target prediction scores are the central element in the analysis.

Furthermore, a combined prediction model with mRNA and miRNA expression data, a *fusion* of these data sets, could improve prediction of clinical endpoints and finally lead to candidate biomarker panels consisting of both: miRNAs as the regulators and genes as the effectors. In most cases only mRNAs or miRNAs are used to build a predictive model, only a few approaches have been proposed to integrate mRNA and miRNA data to discover novel regulatory relations or to build combined prediction models (Buffa et al., 2011). A central problem in these high-dimensional data is the tendency to overfit. When integrating several *omics* data sets the number of features increases what makes the feature selection even more important.

Here, a method capable to fuse mRNA and miRNA expression data in a model to predict a clinical endpoint is introduced (Gade et al., 2011). Given genome wide mRNA and miRNA expression data are available from the same patients the method estimates the regulatory relationships of miRNAs and genes. These estimations can be represented as a graph. Both datasets together with the graph are than used in the prediction model. Likelihood boosting (Binder and Schumacher, 2008b; Tutz and Binder, 2006, cf. section 2.2) was used as a method for fitting prediction models because of its performance and its ability to implicitly select features in the training process. The graph holding the regulation estimates is thereby used to guide the feature selection

leading to better predictions and more stability in resulting feature sets. The workflow of the method is shown in figure 3.1.

As a first step the regulatory relations between miRNAs and mRNAs are estimated. Two sources of informations are considered for this estimation. The first are the expression profiles of the n patients. Based on what is known so far, binding of a miRNA to the target mRNA leads in most cases to the degradation of the target mRNA, which is measurable by gene expression arrays. As a consequence the expression profiles of the miRNAs and their target mRNAs are correlated. Here, the Pearson correlation coefficient $\rho(i,j)$ was calculated for every mRNA i ($1 \leq i \leq p_1$) and miRNA j ($1 \leq j \leq p_2$). The correlation coefficient can be tested for a significant shift from zero leading to a p-value for every mRNA-miRNA pair

$$\begin{aligned} p_{i,j}^{cor} &= P(H_0 : \rho(i,j) = 0) \\ &\qquad \forall\, i \in [1,p_1], j \in [1,p_2] \end{aligned} \tag{3.1}$$

In a genome wide setting the number of tests is enormous ($p_1 \times p_2$) and a high false discovery rate can be expected. Thus, the resulting p-values have to be corrected e.g. with the method from Benjamini and Hochberg (1995). In the following $p_{i,j}^{cor}$ refers to the corrected values.

A strong correlation of the expression profiles of a mRNA-miRNA pair does not necessarily imply a direct regulation but can be caused by secondary interactions(1). A direct regulation requires a sequence match of the seed region of the miRNA and the 3' UTR of the target mRNA. A logical step is to include knowledge about sequence similarities between miRNAs and mRNAs. More advanced are target predictions based on not only similarity between the seed region of the miRNA and the 3' UTR of the mRNA but also thermal stability of the resulting mRNA-miRNA complex and the evolutionary conservation of the mRNA binding site.

The target predictions from MicroCosm (Enright et al., 2003, cf. section 2.5) provides a score reflecting the sequence similarity. Additionally, a theoretical distribution under the null hypothesis that no binding occurs is derived for the scores. At the end a p-value for a possible mRNA-miRNA complex is provided. These p-values $p_{i,j}^{pred}$ are the second source of information used in the method.

(1) Secondary interactions in this case mean indirect regulatory relationships.

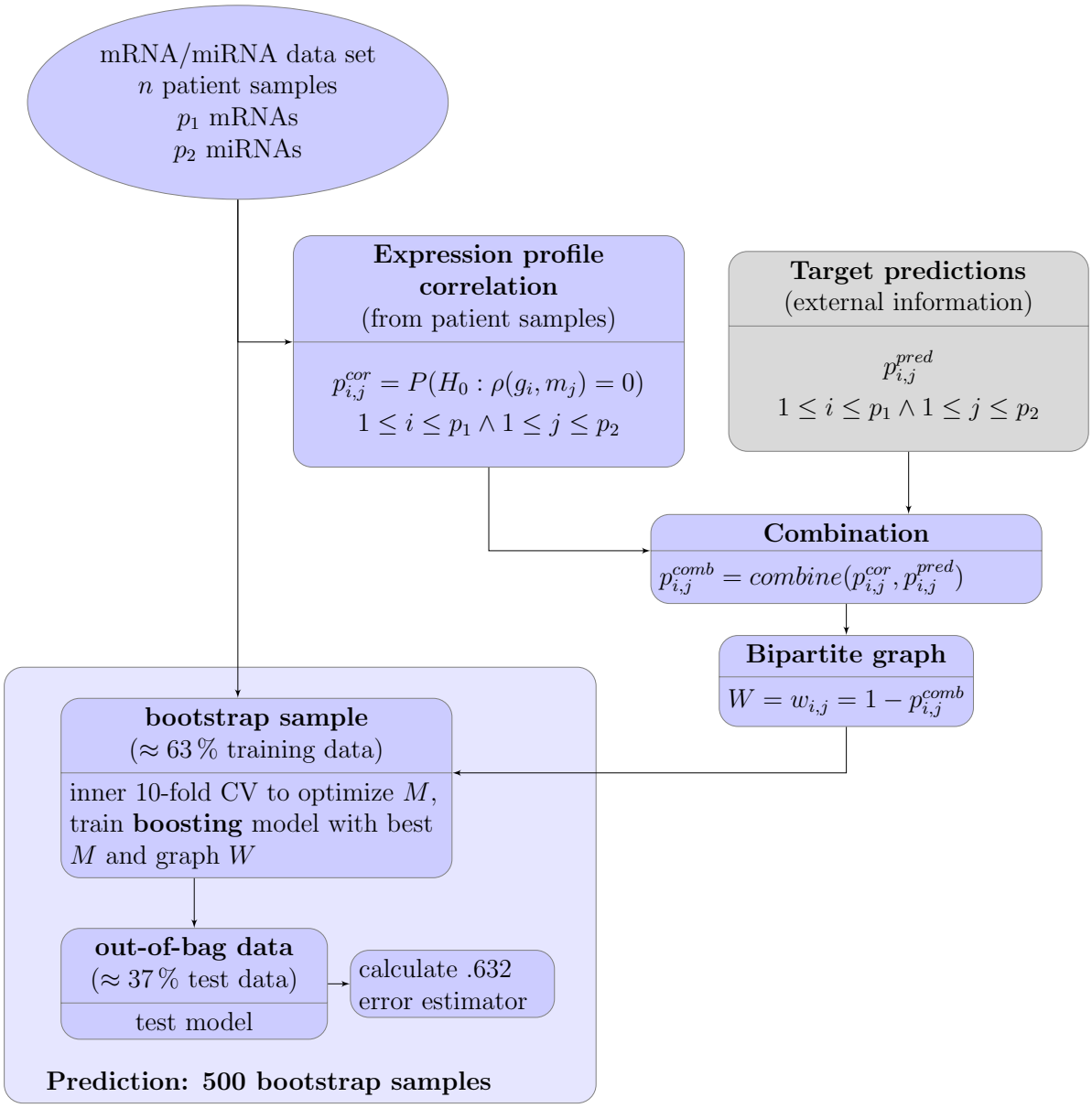


FIGURE 3.1. *The workflow of the proposed method to fuse miRNA and mRNA expression data from the same patients in one prediction model (Gade et al., 2011).*

They reflect the probability that a miRNA j is actually capable of binding mRNA i and strengthen the importance of the connection of a mRNA i and a certain miRNA j in the case where i is a predicted target of j. Since the MicroCosm target database holds only mRNA-miRNA pairs with a p-value below 0.05 the p-values of pairs not present in MicroCosm were set to 1.

Finally, two p-values are obtained for a possible mRNA-miRNA pair. Having p-values is favourable since they are independent of the underlying target prediction score and the number of samples is already taken into account when estimating the correlation between the mRNA and the miRNA. Another advantage is that with the two p-values a combined overall hypothesis can be formulated.

In the statistical field of meta-analysis several methods have been formulated allowing the integration of p-values (Davidov, 2011; Loughin, 2004; Zaykin et al., 2002). The method used here was proposed by Stouffer et al. (1949) and uses z-scores of the single p-values to get a combined one

$$p_{i,j}^{comb} = 1 - \Phi\left(\frac{1}{\sqrt{2}}\left(\Phi^{-1}(1 - p_{i,j}^{cor}) + \Phi^{-1}(1 - p_{i,j}^{pred})\right)\right) \tag{3.2}$$

where $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\,\pi}} e^{\frac{z^2}{2}} dz$ is the probability distribution function of the standard normal distribution. This combination is a central part of the method leading to well distributed combined p-values that reflect the possibility of a binding between miRNA and mRNA (p^{pred}) and the effect of a possible binding to the mRNA level in the cell (p^{cor}).

The combined p-values can easily be transformed into weights

$$w_{i,j} = 1 - p_{i,j}^{comb} \tag{3.3}$$

The resulting matrix of weights $W = w_{i,j}$ can be viewed as the $p_1 \times p_2$ adjacency matrix of a bipartite graph W containing the estimations of the regulatory relationships between mRNAs and miRNAs.

The graph W is interpreted as a directed graph with edges from mRNAs to miRNAs. Together with likelihood boosting the graph is used to guide the feature selection. Thereby weight is transfered from the mRNAs to the miRNAs. This is done similar to the idea of Pathboost (Binder and Schumacher, 2009, cf.

section 2.2.3). But instead of graphs describing biological pathway knowledge the mRNA-miRNA graph W with the regulatory estimations is used. Every time an mRNA i is picked the penalties λ of miRNAs connected to i are lowered according to the weight of the connection (cf. section 2.2.3 for details). As a consequence it is more likely to choose a miRNA j highly correlated and being a predicted regulator of i in one of the next boosting steps. miRNAs with a connection with high weight to i are likely to be a direct regulator of i and can be assumed to be important for the outcome as well.

To get an impression how well the final model can predict the outcome the error has to be estimated. Here, the .632 error estimator (cf. section 2.4.3) is used with 500 bootstrap samples. For every bootstrap sample the number of boosting steps M is optimized via a 10-fold cross-validation (cf. section 2.4.2). After fitting the model including mRNA and miRNA expression data and the graph W the resulting model is tested on the out-of-bag data. Together with the training error the .632 error estimator of the test error can be computed.

3.2 Evaluation of the Method

The new method was evaluated with different objectives in mind using the prostate cancer dataset from Taylor et al. (2010) (cf. section 2.6). As the clinical endpoint of interest the time to biochemical relapse (BCR) was chosen (cf. section 2.6.2). Since these are time-to-event data, CoxBoost was used for fitting the model and the PEC and the IPEC (cf. section 2.4.4) were used as error measurements.

The first question to answer was if the bipartite graph W together with the mRNA and miRNA expression data can improve the prediction error compared to models fitted with only the single data sets (cf. section 3.2.1). As mentioned before, every model was fitted and evaluated with 500 bootstrap samples resulting in 500 IPEC for every model. To be able to compare the different model prediction errors the same bootstrap samples were used to fit each of the models. The model were fitted using the CoxBoost R package (Binder, 2010). The parallel evaluation of the models and the calculation of the IPEC was performed using the R package peperr (Porzelius and Binder, 2010).

The second question concerns the feature selection. By transferring weight from the mRNAs to the miRNAs by using the graph W, the miRNA are favored during the feature selection process. The question was if this is observable in the feature lists of the models. Several authors pointed out that additional knowledge can improve the stability of the selected features and therewith improve the interpretability of the results (Johannes et al., 2010). Thus, it had to be clarified if the graph also improves the stability of the feature selection (section 3.2.2).

The estimation of the graph W is done using all available samples. Though no information about the outcome is used, there might be a risk of overfitting. As pointed out by Ambroise and McLachlan (2002) all modeling steps should be included in the error estimation procedure (bootstrap in this case). Excluded from this general rule are unsupervised screening steps, e.g. variance based filtering of features (Hastie et al., 2009). Therefore, section 3.2.3 examines the influence of the use of the whole dataset when estimating the bipartite graph W.

The influence of a different target prediction algorithm is elucidated in section 3.2.4. Finally, section 3.2.5 includes a comparison with two state-of-the-art methods suited for time-to-event data.

3.2.1 Evaluation of the Prediction Error

To test whether the graph W improves the prediction accuracy by increasing the probability of selecting miRNAs with connections to already chosen mRNAs, CoxBoost was trained on both data sets, not given the graph information, and on the single data sets. To assure a comparability of the prediction models a common penalty of 1296 was determined such that the number of boosting steps exceeds 50 in every case (table 3.1 lists the optimal number of boosting steps for every model).

The 500 .632 estimators for the PEC and IPEC were calculated based on pre-calculated bootstrap samples. The first question to answer was if the graph improves the prediction accuracy. Figure 3.2 shows the PEC (averaged over the 500 bootstrap samples) of the CoxBoost model trained with and without

	Optimal Number of Boosting Steps M
only miRNA	98
only mRNA	100
both no graph	99
both with graph	99

TABLE 3.1. *The table shows the optimal number of boosting steps M for every CoxBoost model. The optimal number of steps was determined using a simple cross-validation procedure on the single and combined data sets. The number of steps that minimizes the average log-partial likelihood is considered optimal (Binder, 2010).*

the graph when given mRNA and miRNA expression data[(2)]. The graph with regulatory relations clearly improved the model in terms of prediction error. To take into account the variance of the prediction errors the 500 IPEC (.632 estimation) resulting from the 500 bootstrap samples were compared. In addition to the models with and without the graph, two CoxBoost models were trained using only the single data sets (mRNA and miRNA expression data alone). The results are shown in figure 3.2. The medians of the resulting 500 IPEC and their interquartile ranges (IQRs) are given in table 3.2. To test whether the differences of the IPECs are significant, a one-sided Wilcoxon test was carried out between the single models without a graph and the model incorporating the bipartite graph. For every three risk prediction models without graph information the difference was significant assuming a significance level of 0.05.

Several authors pointed out that in high overfitting settings the .632 estimator is biased downward and thus the prediction accuracy is overestimated (Efron and Tibshirani, 1997). The .632+ error estimator corrects for this for the cost of a higher computational complexity (cf. section 2.4.3 for a definition of the .632+ estimator). Although, in the comparative setting used in this work, bias is probably of minor importance, additional tests have been carried out comparing the IPEC .632+ estimations of the single models. Table 3.2 summarizes the results.

Both expression data sets together with the graph W improved the prediction error significantly compared to the model without the graph. There was no

[(2)] The data where scaled to ensure a mean of 0 and a standard deviation of 1 for all mRNAs and miRNAs

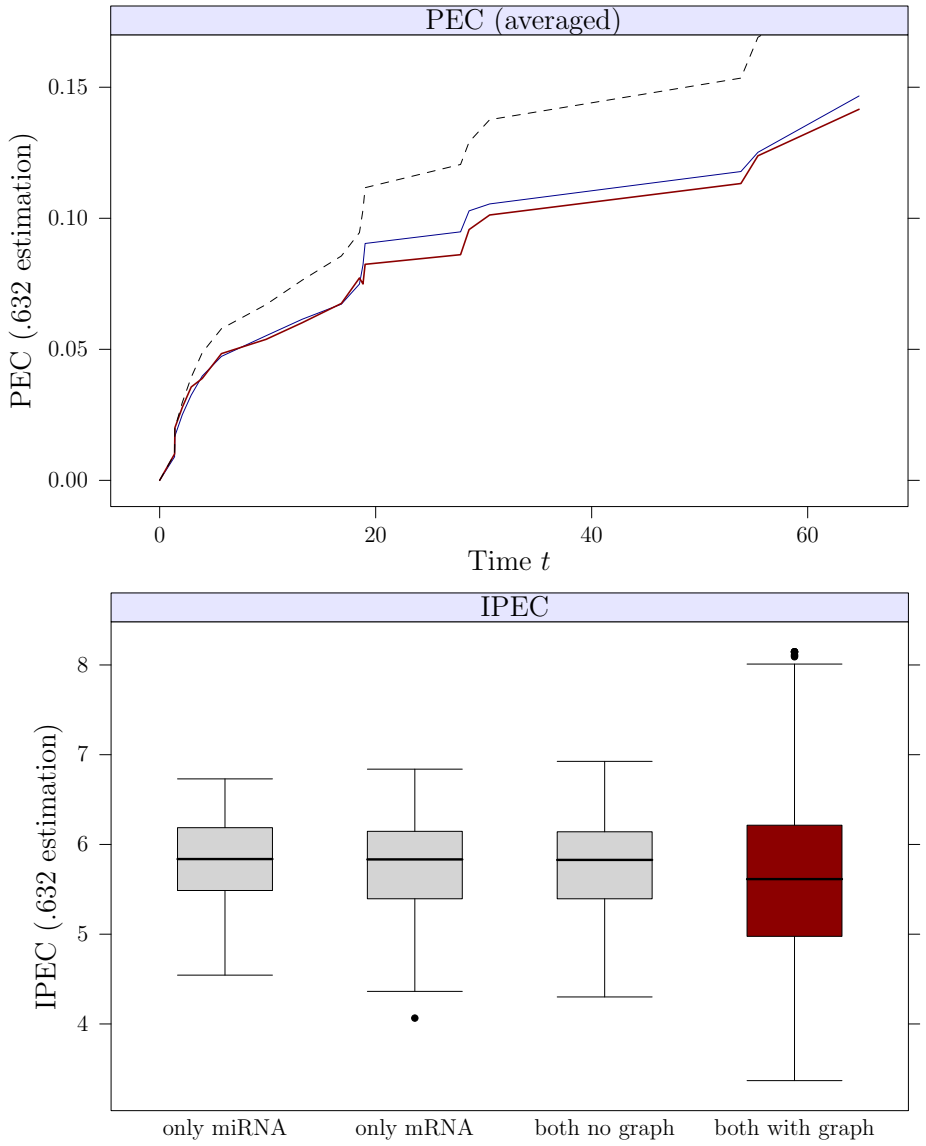


FIGURE 3.2. *The upper figure shows the prediction models trained with (red line) and without (gray line) the bipartite graph describing the relations between the features. The incorporation of the graph resulted in a reduction of the prediction error. The .632 estimation of the prediction error was used in this plot averaged over the 500 bootstrap samples. As a reference (dashed line) the prediction error of the Kaplan-Meier estimator (cf. section 2.2.1) is shown. The lower figure shows the 500 IPEC (.632 estimation) for the models trained only on the miRNA expression data, only on the mRNA expression data, on both data sets but without the graph W, and on both data sets using the graph W. The boxes are the standard boxplots in R. The box represents the interquartile range (IQR) of the data with the median indicated by a bold line. The whiskers extends to the most extreme point that is more than* 1.5 *times the IQR away from the box (*$1.5\,IQR \pm 0.75/0.25 quartile$*). Points above or below the whiskers are considered as outliers (points in the boxplot).*

	IPEC (median)	IQR	p-value
.632 estimator			
only miRNA	5.90	0.88	< 0.001
only mRNA	5.82	0.87	< 0.001
both no graph	5.79	0.86	< 0.001
both with graph	5.46	1.20	-
.632+ estimator			
only miRNA	5.84	0.70	< 0.001
only mRNA	5.83	0.75	< 0.001
both no graph	5.83	0.75	< 0.001
both with graph	5.61	1.20	-

TABLE 3.2. *The table shows the* .632 *and the* .632+ *IPEC estimations (median and IQR) of 500 bootstrap runs. Lower IPEC scores indicate better prediction accuracy. The p-value is the result of a one-sided Wilcoxon test (unpaired) comparing the single data set prediction models and the prediction model without graph with the combination incorporating the bipartite graph.*

clear trend regarding mRNA and miRNA data alone, though the miRNA seemed to perform slightly worse. This might be due to the lower number of features. Interestingly, the combination of both data sets without the graph W yielded almost the same error as the gene expression data alone. Without the graph information the additional information from the miRNAs seemed to be worthless. This underpins the theory that feature selection is the crucial step in these high-dimensional settings and guiding the feature selection via prior knowledge is a successful strategy. The comparison of the .632 and the .632+ estimators yielded similar results which leads to the conclusion that the inclusion of prior knowledge used here is not a strong overfitting setting.

Though the improvement using both data sets and the graph is significant it is rather small compared to the overall error. This might be due the complexity of the problem or due to uncertainty in the graph W describing the relations between mRNA and miRNA. Another reason might be the relative less number of events (18/98 events) aggravating an accurate estimation of the prediction error. Finally, the diverse nature of prostate cancer makes it probably difficult to find a reliable prediction model for a large group of patients. Bair and Tibshirani (2004) proposed the idea to divide the patients into subgroups (for cancer types like prostate cancer where no such subtypes are known a priori)

bases on gene expression data and clinical variables. The stratification of the patients according to such subtypes might afterwards simplify the model fitting process and feature selection. The definition of prostate cancer subtypes is a major topic in cancer research. Thus, this remains to be elucidated.

3.2.2 Evaluation of Stability and Interpretability of Selected Features

The features chosen during fitting of a prediction model are most often as interesting as the prediction power of the resulting model. Gene signatures based on gene expression microarrays have been proposed for diagnostic and prognostic issues. For breast cancer for example gene signatures are used in clinical praxis (van 't Veer et al., 2002; Wang et al., 2005). However, the overlap of different signatures for one cancer type is usually poor.

In settings where many genes (or in general features) are correlated to the clinical outcome it is hard to define the "best" gene set predicting the outcome (Ein-Dor et al., 2005; Venet et al., 2011). Therefore, stability of feature selection is nowadays as important as the prediction accuracy in order to get consistent feature sets suited as possible signature.

In order to investigate the stability of the feature selection and the influence of the graph the 500 bootstrap samples were used. The single bootstrap samples differed on average in 56 patients (figure 3.3) and thus could be used to simulate different patient cohorts. The number of bootstrap samples in that a certain feature was picked by the feature selection was used as an indicator for the stability of the feature selection.

When performing 500 bootstrap samples, 500 is the maximal number how often a certain feature (gene or miRNA) can be selected. Table 3.3 compares these feature counts for CoxBoost trained on mRNA and miRNA expression data with and without the graph W. It shows the top 30 features based on the feature counts for the two models. The graph information remarkably improved the stability of the feature selection process. The top features were picked almost twice as often with inclusion of the graph than the top features picked in the model without the graph.

No graph		With graph	
Feature	**Counts**	**Feature**	**Counts**
ESM1	161	hsa-miR-513a-3p	329
hsa-miR-412	151	hsa-miR-513a-5p	316
INHBA	130	hsa-miR-128	249
COMP	126	hsa-miR-1226*	233
ZFHX4	114	hsa-miR-1231	209
SLC6A14	103	hsa-miR-1224-5p	206
hsa-miR-484	92	hsa-miR-220a	199
PI15	83	hsa-miR-1233	198
hsa-miR-556-3p	79	hsa-miR-208a	169
hsa-miR-409-3p	74	hsa-miR-199b-3p	168
ABCC11	70	hsa-miR-513b	157
hsa-miR-431*	65	hsa-miR-527	154
hsa-miR-342-3p	52	*COMP*	150
HOXB6	49	hsa-miR-1225-3p	146
PRM2	48	hsa-miR-1234	144
CEBPD	47	*INHBA*	141
PARS2	44	hsa-miR-1226	140
3603927	42	hsa-miR-1237	139
KRTAP26-1	42	hsa-miR-1225-5p	136
hsa-miR-451	42	hsa-miR-1238	127
ZNF334	39	hsa-miR-513c	126
GRIK1	39	hsa-miR-1229	119
hsa-miR-147b	35	hsa-miR-1228*	117
ITGBL1	34	hsa-miR-1227	102
ITGA11	33	*ESM1*	100
3680663	33	*ZFHX4*	98
TMC5	32	hsa-miR-1224-3p	79
hsa-miR-103	32	hsa-miR-597	68
3400384	30	hsa-miR-409-3p	55
hsa-miR-409-5p	29	hsa-miR-625	55

TABLE 3.3. *The table lists the top 30 features from CoxBoost with and without graph information. mRNA names are given by their official HGNC (Seal et al., 2011) symbols (capital letters) , or in case where no HGNC symbol was available the Affymetrix IDs (7 digit number) are given. miRNA names are given by their official miRBase IDs (starting with hsa-miR). The Counts columns indicate how many times the feature was chosen. Consequently, the maximal count would be 500.*

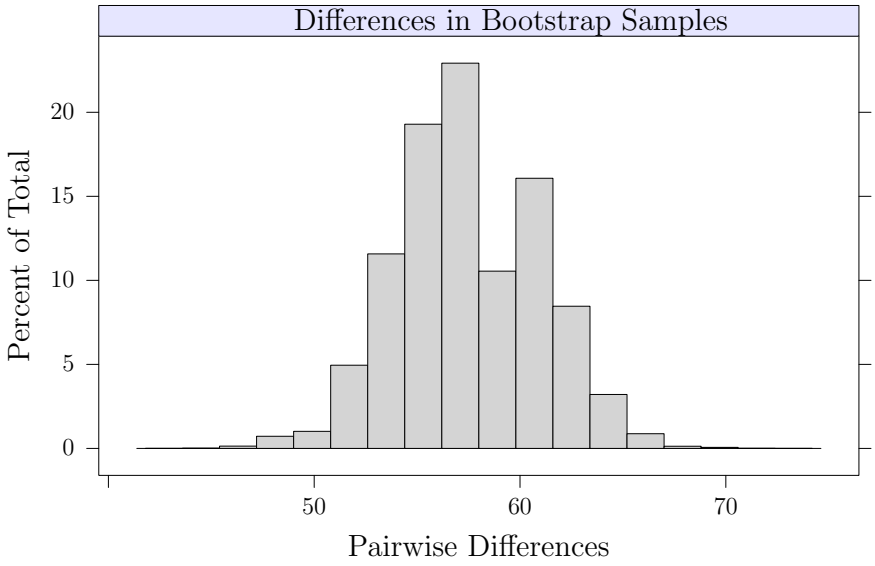


FIGURE 3.3. *The figure shows the pairwise differences (in number of patients) of the single bootstrap samples.*

Another difference is the proportion of genes and miRNAs picked by the models. The ratio among the top 30 features between mRNAs and miRNAs was 2/3 without the graph, which was already higher than expected (there were almost 25 time as many mRNAs than miRNAs in the top list). The graph transfered weight from the mRNAs to the miRNAs. It is thus not surprising that the ratio drops to 2/15. However, this clearly showed the influence of the graph W on the feature selection.

Obviously, the graph lead to a more stable feature selection and a favouring of miRNAs in the model. At the same time the accuracy of the predictions was improved leading to the assumption that miRNA expression data carried the main part of information needed to predict the time to the biochemical relapse. However, it is important to note that miRNA expression data alone failed to predict the relapse as accurate as the combined data with the graph. This may be caused by the fact that one miRNA can have several targets and dysregulation of a miRNA can affect multiple molecular pathways with no direct connection to the outcome. Therefore, the genes as effectors seem to be a mandatory source of information.

Among the top 30 features picked using the graph there are some miRNAs found to play a role in prostate cancer, e.g. hsa-miR-513 (Porkka et al., 2007)

and hsa-miR-128 (Khan et al., 2010). However, most of the miRNAs have not been associated with prostate cancer before. The genes among the top 100 features of CoxBoost with and without graph were investigated for enriched GO terms (The Gene Ontology Consortium et al., 2000) using the R package topGO (Alexa and Rahnenfuhrer, 2010). In both cases GO terms functional related to cancer were found. However, no clear pattern could be revealed in one or the other case. It is therefore important to note that it is not straightforward to derive functional implications for single biomarkers from a panel found by a prediction model.

To summarize, it can be concluded that the graph W improves the stability of the feature selection process and, as expected, favors miRNAs in the selection process. If this would lead to an improvement in sense of predictions has to be shown in the future when more datasets with such a setting will be available.

3.2.3 Assessing the Influence of Correlations

To be able to estimate the prediction error of a certain model, all modeling steps should be included in the error estimation procedure (Ambroise and McLachlan, 2002). Otherwise the estimated error might be to optimistic, that means it is biased downwards. A typical example is known as *selection bias* occurring if features are selected using the outcome over all training samples (Furlanello et al., 2003).

In the workflow described in figure 3.1 the whole data set is used, prior to the bootstrapping procedure, to estimate the regulation graph W. This involves the danger of overfitting and a biased error estimation.

On the other hand, the graph estimation includes the correlations between mRNAs and miRNAs without any knowledge about the outcome. To check if this alone lead to overfitting one test run was conducted were the estimation of the regulation graph W was moved to the bootstrap procedure. That means the correlations were re-calculated for each bootstrap sample using solely the patient samples included in that particular bootstrap sample.

Again, the .632 estimator of the IPEC was used to compare CoxBoost with and without the graph. The median IPEC of CoxBoost with the graph increased from 5.46 to 5.64 with an IQR of 0.99. In comparison with the IPECs

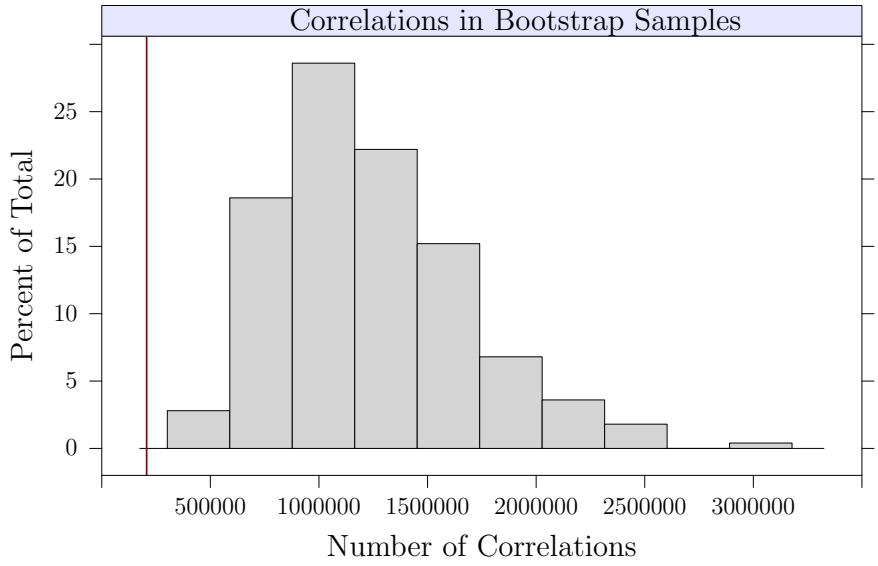


FIGURE 3.4. *The figure shows the number of significant correlations (assuming a significance level of* 0.05*) in the 500 bootstrap samples (the correlations between mRNAs and miRNAs were calculated using only the patients samples included in the particular boostrap sample). The red line indicates the number of correlations found between mRNAs and miRNAs when the whole data set (all 98 patient samples) was used.*

of CoxBoost without graph the prediction error was still significantly smaller assuming a significance level of 0.05 (p-value from one-sided Wilcoxon test: 0.006).

Although the prediction error increased when not using all samples for estimating the regulation graph W the result remains the same. It is, however, still unclear if the higher IPEC is due to overfitting. The number of correlations between mRNAs and miRNAs found in the bootstrap samples is obviously larger than the number of correlations resulting if all patient samples were used (figure 3.4). For bootstrapping the patients samples were drawn with replacement (cf. section 2.4.3). As a consequence some patient samples are contained several times in a bootstrap sample. This can cause artificial correlations between the features and the outcome (Binder and Schumacher, 2008a). The same effect probably caused the higher number of correlations in the bootstrap graph estimations leading to many false positive connections in the graphs. These connections could be another reason for the higher prediction error estimates.

From these results it can be concluded that no clear overfitting occurs and hence all samples can be used to estimate the graph W.

3.2.4 Assessing the Influence of Different Target Prediction Databases

In the models described above the MicroCosm target prediction database has been used. Many other approaches exist with partially large differences (Panagiotis et al., 2009). Thus, the question arises how the choice of the target prediction algorithm influences the performance of our method.

To our knowledge, TargetScan (Friedman et al., 2009) is the only target prediction source besides MicroCosm that delivers not only a score but also a p-value for a miRNA-mRNA pair. The TargetScan flatfiles (version 5.2) contain a score P_{CT} which can according to Friedman et al. (2009) be used to asses the biological relevance of predicted miRNA-mRNA interactions. $1 - P_{CT}$ is an estimate of the FDR.

This FDR was used as prediction p-values $p_{i,j}^{pred}$ to build the graph W. CoxBoost using this graph yielded a median IPEC (.632 estimation) of 6.60 with an IQR of 0.95. Using MicroCosm the median IPEC was 5.46 with an IQR of 1.20.

Obviously, the use of TargetScan resulted in a higher prediction error. This result can possibly be explained by the lower coverage of TargetScan. From the 723 miRNAs in the data set only 170 could be found in TargetScan having a P_{CT} value. In comparison, the MicroCosm predictions contained 698 out of the 723 miRNAs with p-values. This indicates that the predictions play an important role for the regulation graph W and that the correlations alone do not cover the regulations sufficiently.

Of course many other target predictions are available nowadays, e.g. PicTar (Krek et al., 2005). However, in order to use other sources, the scores from these predictions have to be combined with either the correlation p-values $p_{i,j}^{cor}$ or the correlation coefficients $\rho_{i,j}$ directly. But this eliminates the handy interpretation of the combined p-values $p_{i,j}^{comb}$ and makes it necessary to find another combination function (3.2).

Testing all available algorithms for miRNA target prediction is beyond the scope of this work. Yet, the choice of the target prediction database seems to be an important factor. Combinations of different target prediction algorithms to improve the coverage might be a possible solution in future research.

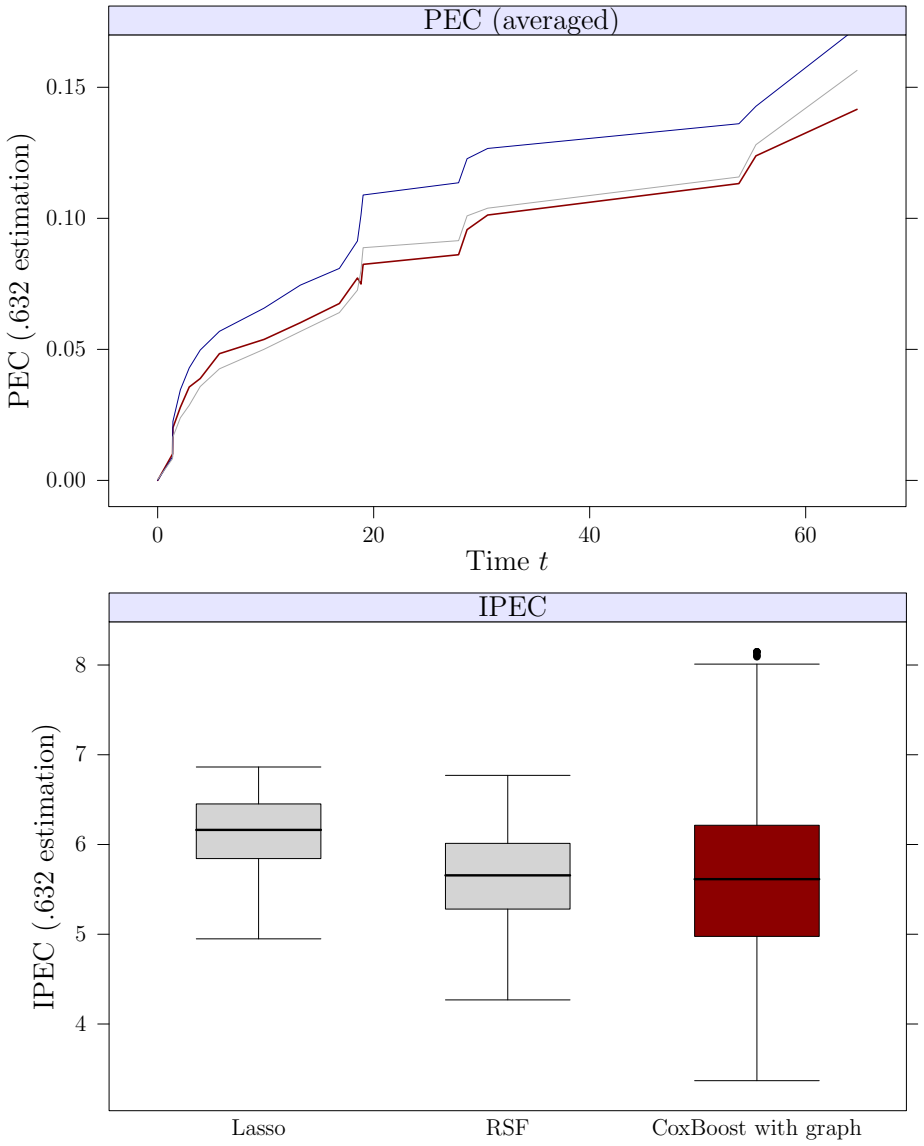


FIGURE 3.5. *The upper figure shows the PEC (.632 estimations, averaged over the 500 bootstrap samples) for CoxBoost with the graph W (red line), Lasso (blue line), and RSF (gray line). The lower figure shows the IPEC for all three methods.*

3.2.5 Comparison to Other Prediction Methods

The assessments shown so far compared different models that were fitted with CoxBoost. In addition two other methods suited for high-dimensional time-to-event data were used for comparison.

Lasso (Tibshirani, 1996, 1997, see section 2.3.1) belongs to the regularization or shrinkage methods. It is a regression method for linear and general linear

models where the coefficient vector $\boldsymbol{\beta}$ is penalized via an L_1 norm. The adaption for the Cox proportional hazard model used here was proposed by Goeman (2010). The associated R package penalized (Goeman, 2011) was used for fitting the Lasso estimator. To guarantee comparability, the same mRNA and miRNA expression data and the identical 500 bootstrap samples were used for the evaluation. Similar to CoxBoost the resulting fit is an estimate of the survivor function and a risk prediction. As before, the .632 estimation of the IPEC was used as an error measurement.

The second method is a adaption of Random Forest for time-to-event data. Random Survival Forest (RSF, Ishwaran et al., 2008, see section 2.3.2) was trained given the mRNA and miRNA expression using the 500 bootstrap samples. The R-package randomSurvivalForest (Ishwaran and Kogalur, 2007) was used for model fitting.

The complexity parameter for RSF is the number of variables m to choose from at each node. This parameter had to be determined a priori. Following a suggestion from Porzelius et al. (2011b) the .632+ estimator of the IPEC (should be minimal) was used to determine the optimal m given three choices $\frac{1}{2}\sqrt{p}$, $\sqrt{p}$, and $2\sqrt{p}$. In this case $p = p_1 + p_2$ is the total number of features that means the number of mRNAs and miRNAs. These three choices for m follow a suggestion from Breiman (2002). Furthermore the "logrank" splitting rule has been used and, in order to gain speed, for each variable $nsplit = 2$ randomly chosen splitting points were considered (cf. Ishwaran and Kogalur, 2007, 2010 for details). The model was trained with the default of $ntree = 1000$ trees.

Figure 3.5 shows the PEC averaged over the 500 bootstrap samples. Obviously, CoxBoost with graph as well as RSF performed better than Lasso. RSF was slightly worse than CoxBoost with the graph, though the difference is marginal.

To assess a statistical significance the 500 IPEC (.632 estimations) from each method (figure 3.5) were compared using a one-sided Wilcoxon test. The results can bee seen in table 3.4. Although CoxBoost with the graph W performed only slightly better on this data sets, the performance gain is significant assuming a significance level of 0.05. RSF seems to be able to detect even non-linear relations between the features and the outcome. This might be the reason why

	IPEC (median)	IQR	p-value
Lasso	6.10	1.12	< 0.001
RSF	5.66	0.78	< 0.001
CoxBoost only miRNA	5.90	0.88	< 0.001
CoxBoost only mRNA	5.82	0.87	< 0.001
CoxBoost with graph	5.46	1.20	-

TABLE 3.4. *The table shows the comparison of Lasso and RSF with CoxBoost with the bipartite graph regarding the prediction error. The median and IQR from 500 IPECs were calculated. The p-value is based on a one-sided Wilcoxon test comparing the 500 IPECs of Lasso and RSF with the IPECs of CoxBoost. As a reference the values of CoxBoost using only the single data sets are shown as well.*

it performed so remarkably well without information about the relationships among the features. Surprisingly Lasso performed even worse than CoxBoost on the single data sets.

Besides the prediction error there was a remarkable difference in the runtime of the three risk prediction models. Training and prediction for 500 bootstrap samples took 40:17 hours for RSF, 2:25 hours for Lasso, and 1:16 hours for CoxBoost with graph on a 20 core (2.7 GHz) machine with 64 GB memory.

To summarize, Lasso and RSF performed worse (in case of RSF only slightly worse) than CoxBoost with the graph W while taking more computation time.

Chapter 4

Conclusions

Nowadays, the prostate specific antigen, or short PSA, is the standard diagnostic marker to detect prostate cancer. Hence, it is possible to detect prostate cancer in an early stage and treat a cancer that is still ranked on top of cancer caused death in the western hemisphere. However, the specificity of PSA is still under debate. In many cases patients are over-treated causing a heavy burden of side effects. To avoid unnecessary patient treatment, prognostic biomarkers are needed that can complement PSA. But until now, no satisfying prognostic marker has been established.

For years large scale gene expression measurements have been used to search for new promising biomarkers, especially in a prognostic setting for various cancer types. Machine learning methods have been applied in the field of bioinformatics to guide this search, and help to condense thousands of features into a signature with prognostic value. In the last decade a new class of non-coding RNA molecules were found. MicroRNAs (miRNAs) were found to be major regulators of gene expression. Similar to mRNAs their expression can be measured on a global scale and it seems logical, as a next step, to search for combined signatures of genes and miRNAs. Yet, until now, there is still a lack of methods for building such a combined prediction model - the *fusion* - from both kinds of data.

The regulatory relationships between miRNAs and genes violate the general assumption of independence among the features in such a model. Even for high-dimensional gene expression data alone the flaws of methods relying on such assumptions became apparent. Unsatisfactory prediction accuracy and

poor overlap among signatures were the reason to develop new methods that incorporate an estimation of the regulatory relationships among the genes as *prior* knowledge. Many databases, based on different approaches and data sources, are available providing such knowledge.

In a similar manner I developed a workflow to combine miRNA and gene expression data from the same patients to build a predictive model. Boosting, as the underlying model fitting method, is quite flexible, can be used for different types of endpoints and, most importantly, it allows the integration of *prior* biological knowledge. We showed that the regulatory relationships between miRNAs and genes can be effectively estimated as *regulatory graph* from the expression data and target prediction databases.

From these two sources a combined prediction model could be fitted. We showed on a large prostate cancer data set that our workflow yielded a model with better prediction accuracy compared to using only the expression data. Furthermore, the stability of the feature selection could be improved significantly. A comparison with other methods suited for time-to-event data showed that the improvement in prediction accuracy by incorporating the regulatory graph is not a bias caused by the boosting approach.

Without a doubt, the prediction results can be substantially improved using better target prediction databases including more accurate and more complete knowledge. In-silico miRNA target prediction is a central topic in the miRNA related research. We are convinced that the accuracy of these predictions will increase dramatically in the next years and hope that more methods will use these resources to build combined models from miRNA and gene expression data.

References

Alexa, A. and Rahnenfuhrer, J. (**2010**). *topGO: Enrichment analysis for Gene Ontology*. R package version 2.6.0.

Allen, D. M. (**1974**). "The Relationship between Variable Selection and Data Agumentation and a Method for Prediction." *Technometrics*, 16(1): 125–127.

Alpaydin, E. (**2010**). *Introduction to machine learning*. The MIT Press, Camebridge, Massachusette, 2nd edition.

Amaldi, E. and Kann, V. (**1998**). "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems." *Theoretical Computer Science*, 209: 237–260.

Ambroise, C. and McLachlan, G. J. (**2002**). "Selection bias in gene extraction on the basis of microarray gene-expression data." *Proceedings of the National Academy of Sciences of the United States of America*, 99(10): 6562–6566.

Ambros, V., Bartel, B., Bartel, D. P., Burge, C. B., Carrington, J. C., Chen, X., Dreyfuss, G., Eddy, S. R., Griffiths-jones, S., Marshall, M., Matzke, M., Ruvkun, G., and Tuschl, T. (**2003**). "A uniform system for microRNA annotation." *RNA*, 9: 277–279.

Bair, E. and Tibshirani, R. (**2004**). "Semi-supervised methods to predict patient survival from gene expression data." *PLoS Biology*, 2(4): E108.

Balk, S. P., Ko, Y.-J., and Bubley, G. J. (**2003**). "Biology of Prostate-Specific Antigen." *Journal of Clinical Oncology*, 21(2): 383–391.

Bard, J. B. L. and Rhee, S. Y. (**2004**). "Ontologies in biology: design, applications and future challenges." *Nature Reviews Genetics*, 5(3): 213–222.

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. a., Phillippy, K. H., Sherman, P. M., Muertter, R. N., Holko, M., Ayanbule, O., Yefanov, A., and Soboleva, A. (**2011**). "NCBI GEO: archive for functional genomics data sets–10 years on." *Nucleic Acids Research*, 39(Database issue): D1005–D1010.

Bartel, D. P. (**2009**). "MicroRNAs: target recognition and regulatory functions." *Cell*, 136(2): 215–233.

Bartel, D. P. and Chen, C.-Z. (**2004**). "Micromanagers of gene expression: the potentially widespread influence of metazoan microRNAs." *Nature Reviews Genetics*, 5(5): 396–400.

Beissbarth, T. (**2006**). "Interpreting experimental results using gene ontologies." *Methods in Enzymology*, 411: 340–352.

Beissbarth, T. and Speed, T. P. (**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." *Journal of Biomedical Informatics*, 40(6): 787–802.

Bellman, R. E. (**1961**). *Adaptive control processes: a guided tour*. Princeton University Press.

Benjamini, Y. and Hochberg, Y. (**1995**). "Controlling the False Discovery Rate : A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society, Series B*, 57(1): 289–300.

Bi, J., Bennett, K. P., Embrechts, M., Breneman, C. M., and Song, M. (**2003**). "Dimensionality Reduction via Sparse Support Vector Machines." *Journal of Machine Learning Research*, 3: 1229–1243.

Bill-Axelson, A., Holmberg, L., Filén, F., Ruutu, M., Garmo, H., Busch, C., Nordling, S., Häggman, M., Andersson, S.-O., Bratell,

S., Spångberg, A., Palmgren, J., Adami, H.-O., and Johansson, J.-E. (**2008**). "Radical Prostatectomy Versus Watchful Waiting in Localized Prostate Cancer: the Scandinavian Prostate Cancer Group-4 Randomized Trial." *Journal of the National Cancer Institute*, 100(16): 1144–1154.

Binder, H. (**2010**). *CoxBoost: Cox models by likelihood based boosting for a single survival endpoint or competing risks*. R package version 1.2-2.

Binder, H., Allignol, A., Schumacher, M., and Beyersmann, J. (**2009**). "Boosting for high-dimensional time-to-event data with competing risks." *Bioinformatics*, 25(7): 890–896.

Binder, H. and Schumacher, M. (**2008**a). "Adapting Prediction Error Estimates for Biased Complexity Selection in High-Dimensional Bootstrap Samples." *Statistical Applications in Genetics and Molecular Biology*, 7(1): 27.

Binder, H. and Schumacher, M. (**2008**b). "Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models." *BMC Bioinformatics*, 9: 14.

Binder, H. and Schumacher, M. (**2009**). "Incorporating pathway information into boosting estimation of high-dimensional risk prediction models." *BMC Bioinformatics*, 10(18): 11.

Biomarkers Definitions Working Group (**2001**). "Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework." *Clinical Pharmacology & Therapeutics*, 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 "Proceedings of the fifth annual workshop on Computational learning theory," pages 144–152. ACM, Pittsburgh, Pennsylvania, United States.

Brase, J. C., Johannes, M., Schlomm, T., Fälth, M., Haese, A., Steuber, T., Beissbarth, T., Kuner, R., and Sültmann, H. (**2011**). "Circulating miRNAs are correlated with tumor progression in prostate cancer." *International Journal of Cancer*, 128(3): 608–616.

Brase, J. C., Wuttig, D., Kuner, R., and Sültmann, H. (**2010**). "Serum microRNAs as non-invasive biomarkers for cancer." *Molecular Cancer*, 9(1): 306.

Breiman, L. (**1996**). "Bagging Predictors." *Machine Learning*, 24: 123–140.

Breiman, L. (**2001**). "Random Forests." *Machine Learning*, 45: 5–32.

Breiman, L. (**2002**). "Manual on Setting Up, Using, And Understanding Random Forests V3.1." URL `http://oz.berkeley.edu/users/breiman/Using_random_forests_V3.1.pdf`.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (**1984**). *Classification and regression trees*. Wadsworth International Group, Belmont, CA.

Breiman, L. and Spector, P. (**1992**). "Submodel Selection and Evaluation in Regression. The X-Random Case." *International Statistical Review*, 60(3): 291–319.

Breslow, N. (**1972**). "Contribution to the discussion of paper by D.R. Cox." *Journal of the Royal Statistical Society , Series B*, 34(2): 216–217.

Buffa, F. M., Camps, C., Winchester, L., Snell, C. E., Gee, H. E., Sheldon, H., Taylor, M., Harris, A. L., and Ragoussis, J. (**2011**). "microRNA-Associated Progression Pathways and Potential Therapeutic Targets Identified by Integrated mRNA and microRNA Expression Profiling in Breast Cancer." *Cancer Research*, 71: 5635–5645.

Bühlmann, P. and Hothorn, T. (**2007**). "Boosting Algorithms: Regularization, Prediction and Model Fitting." *Statistical Science*, 22(4): 477–505.

Bühlmann, P. and Yu, B. (**2003**). "Boosting With the L2 Loss: Regression and Classification." *Journal of the American Statistical Association*, 98: 324–339.

Cary, M. P., Bader, G. D., and Sander, C. (**2005**). "Pathway information for systems biology." *FEBS Letters*, 579(8): 1815–1820.

Chajès, V., Joulin, V., and Clavel-Chapelon, F. (**2011**). "The fatty acid desaturation index of blood lipids, as a biomarker of hepatic stearoyl-CoA desaturase expression, is a predictive factor of breast cancer risk." *Current Opinion in Lipidology*, 22(1): 6–10.

Chambers, J. M. and Hastie, T., editors (**1992**). *Statistical Models in S*. Wadsworth & Brooks/Cole, 1st edition.

Cheng, C. and Li, L. M. (**2008**). "Inferring microRNA activities by combining gene expression with microRNA target prediction." *PloS one*, 3(4): 9.

Cho, J.-H., Gelinas, R., Wang, K., Etheridge, A., Piper, M. G., Batte, K., Dakhallah, D., Price, J., Bornman, D., Zhang, S., Marsh, C., and Galas, D. (**2011**). "Systems biology of interstitial lung diseases: integration of mRNA and microRNA expression changes." *BMC Medical Genomics*, 4(1): 8.

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." *Comparative and Functional Genomics*, 5(2): 163–168.

Choi, N., Zhang, B., Zhang, L., Ittmann, M., and Xin, L. (**2012**). "Adult Murine Prostate Basal and Luminal Cells Are Self-Sustained Lineages that Can Both Serve as Targets for Prostate Cancer Initiation." *Cancer Cell*, 21(2): 253–265.

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

Chung, C. C. and Chanock, S. J. (**2011**). "Current status of genome-wide association studies in cancer." *Human Genetics*, 130(1): 59–78.

Coppola, V., De Maria, R., and Bonci, D. (**2010**). "MicroRNAs and prostate cancer." *Endocrine-related cancer*, 17: F1–F17.

Cottrell, M., Hammer, B., Hasenfuss, A., and Villmann, T. (**2006**). "Batch and median neural gas." *Neural Networks*, 19(6-7): 762–771.

Cover, T. M. and Hart, P. E. (**1967**). "Nearest Neighbor Pattern Classification." *IEEE Transactions on Information Theory*, 13(1): 21–27.

Cox, D. R. (**1972**). "Regression Models and Life-Tables." *Journal of the Royal Statistical Society , Series B*, 34(2): 187–220.

Davidov, O. (**2011**). "Combining p-values using order-based methods." *Computational Statistics & Data Analysis*, 55(7): 2433–2444.

Demir, E., Cary, M. P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J., Schacherer, F., Martinez-Flores, I., Hu, Z., Jimenez-Jacinto, V., Joshi-Tope, G., Kandasamy, K., Lopez-Fuentes, A. C., Mi, H., Pichler, E., Rodchenkov, I., Splendiani, A., Tkachev, S., Zucker, J., Gopinath, G., Rajasimha, H., Ramakrishnan, R., Shah, I., Syed, M., Anwar, N., Babur, O., Blinov, M., Brauner, E., Corwin, D., Donaldson, S., Gibbons, F., Goldberg, R., Hornbeck, P., Luna, A., Murray-Rust, P., Neumann, E., Ruebenacker, O., Reubenacker, O., Samwald, M., van Iersel, M., Wimalaratne, S., Allen, K., Braun, B., Whirl-Carrillo, M., Cheung, K.-H., Dahlquist, K., Finney, A., Gillespie, M., Glass, E., Gong, L., Haw, R., Honig, M., Hubaut, O., Kane, D., Krupa, S., Kutmon, M., Leonard, J., Marks, D., Merberg, D., Petri, V., Pico, A., Ravenscroft, D., Ren, L., Shah, N., Sunshine, M., Tang, R., Whaley, R., Letovksy, S., Buetow, K. H., Rzhetsky, A., Schachter, V., Sobral, B. S., Dogrusoz, U., McWeeney, S., Aladjem, M., Birney, E., Collado-Vides, J., Goto, S., Hucka, M., Le Novère, N., Maltsev, N., Pandey, A., Thomas, P., Wingender, E., Karp, P. D., Sander, C., and Bader, G. D. (**2010**). "The BioPAX community standard for pathway data sharing." *Nature Biotechnology*, 28(9): 935–942.

Dettling, M. and Buhlmann, P. (**2003**). "Boosting for tumor classification with gene expression data." *Bioinformatics*, 19(9): 1061–1069.

Dudoit, S., Fridlyand, J., and Speed, T. P. (**2012**). "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data." *Journal of the American Statistical Association*, 97(457): 77–87.

Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P., and Trent, J. M. (**1999**). "Expression profiling using cDNA microarrays." *Nature Genetics*, 21(1 Suppl): 10–14.

Efron, B. (**1979**). "Bootstrap Methods: Another Look at the Jackknife." *The Annals of Statistics*, 7(1): 1–26.

Efron, B. (**1983**). "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation." *Journal of the American Statistical Association*, 78(382): 316–331.

Efron, B. and Gong, G. (**1983**). "A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation." *The American Statistician*, 37(1): 36–48.

Efron, B. and Tibshirani, R. (**1997**). "Improvements on Cross-Validation : The . 632+ Bootstrap Method." *Journal of the American Statistical Association*, 92(438): 548– 560.

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.

Enright, A. J., John, B., Gaul, U., Tuschl, T., Sander, C., and Marks, D. S. (**2003**). "MicroRNA targets in Drosophila." *Genome Biology*, 5: 14.

Everitt, B. S. and Hothorn, T. (**2006**). *A Handbook of Statistical Analyses Using R*. Chapman & Hall/CRC.

Felici, A., Pino, M. S., and Carlini, P. (**2012**). "A Changing Landscape in Castration-Resistant Prostate Cancer Treatment." *Frontiers in Endocrinology*, 3: 85:8.

Filipowicz, W., Bhattacharyya, S. N., and Sonenberg, N. (**2008**). "Mechanisms of post-transcriptional regulation by microRNAs: are the answers in sight?" *Nature Reviews Genetics*, 9: 102–114.

Fisher, R. A. (**1936**). "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics*, 7: 179–188.

Fix, E. and Hodges, J. L. (**1951**). "Discriminatory analysis - nonparametric discrimination: Consistency properties." Technical Report 21-49-004,4, U.S. Air Force, School of Aviation Medicine, Randolph Field, TX.

Flicek, P., Amode, M. R., Barrell, D., Beal, K., Brent, S., Carvalho-Silva, D., Clapham, P., Coates, G., Fairley, S., Fitzgerald, S., Gil, L., Gordon, L., Hendrix, M., Hourlier, T., Johnson, N., Kähäri, A. K., Keefe, D., Keenan, S., Kinsella, R., Komorowska, M., Koscielny, G., Kulesha, E., Larsson, P., Longden, I., McLaren, W., Muffato, M., Overduin, B., Pignatelli, M., Pritchard, B., Riat, H. S., Ritchie, G. R. S., Ruffier, M., Schuster, M., Sobral, D., Tang, Y. A., Taylor, K., Trevanion, S., Vandrovcova, J., White, S., Wilson, M., Wilder, S. P., Aken, B. L., Birney, E., Cunningham, F., Dunham, I., Durbin, R., Fernández-Suarez, X. M., Harrow, J., Herrero, J., Hubbard, T. J. P., Parker, A., Proctor, G., Spudich, G., Vogel, J., Yates, A., Zadissa, A., and Searle, S. M. J. (**2012**). "Ensembl 2012." *Nucleic Acids Research*, 40(Database issue): D84–90.

Freund, Y. (**1995**). "Boosting a weak learning algorithm by majority." *Information and Computation*, 121: 256–285.

Freund, Y. and Schapire, R. E. (**1996**). "Experiments with a New Boosting Algorithm." In "Proceedings of the Thirteenth International Conference on Machine Learning," pages 148–156.

Friedman, J. (**2008**). "Response to Mease and Wyner , Evidence Contrary to the Statistical View of Boosting." *Journal of Machine Learning Research*, 9: 175–180.

Friedman, J., Hastie, T., and Tibshirani, R. (**2000**). "Additive Logistic Regression: A Statistical View of Boosting." *The Annals of Statistics*, 28(2): 337–407.

Friedman, J. H. (**1997**). "On Bias , Variance , 0 / 1 - Loss , and the Curse-of-Dimensionality." *Data Mining and Knowledge Discovery*, 1: 55–77.

Friedman, R. C., Farh, K. K.-H., Burge, C. B., and Bartel, D. P. (**2009**). "Most mammalian mRNAs are conserved targets of microRNAs." *Genome Research*, 19(1): 92–105.

Furlanello, C., Serafini, M., Merler, S., and Jurman, G. (**2003**). "Entropy-based gene ranking without selection bias for the predictive classification of microarray data." *BMC Bioinformatics*, 4: 20.

Gade, S., Porzelius, C., Maria, F., Brase, J. C., Wuttig, D., Kuner, R., Binder, H., Holger, S., and Beissbarth, T. (**2011**). "Graph based fusion of miRNA and mRNA expression data improves clinical outcome prediction in prostate cancer." *BMC Bioinformatics*, 12: 488.

Garzon, R., Fabbri, M., Cimmino, A., Calin, G. A., and Croce, C. M. (**2006**). "MicroRNA expression and function in cancer." *Trends in molecular medicine*, 12(12): 580–7.

Gerds, T. a. and Schumacher, M. (**2006**). "Consistent Estimation of the Expected Brier Score in General Survival Models with Right-Censored Event Times." *Biometrical Journal*, 48(6): 1029–1040.

Gerds, T. a. and Schumacher, M. (**2007**). "Efron-type measures of prediction error for survival analysis." *Biometrics*, 63(4): 1283–1287.

Germann, S., Gratadou, L., Dutertre, M., and Auboeuf, D. (**2012**). "Splicing Programs and Cancer." *Journal of Nucleic Acids*, 2012: 9.

Gerstein, M. B., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O., Emanuelsson, O., Zhang, Z. D., Weissman, S., and Snyder, M. (**2007**). "What is a gene, post-ENCODE? History and updated definition." *Genome Research*, 17(6): 669–681.

Giraldez, A. J., Mishima, Y., Rihel, J., Grocock, R. J., Van Dongen, S., Inoue, K., Enright, A. J., and Schier, A. F. (**2006**). "Zebrafish MiR-430 promotes deadenylation and clearance of maternal mRNAs." *Science*, 312: 75–79.

Goeman, J. J. (**2010**). "L1 penalized estimation in the Cox proportional hazards model." *Biometrical Journal*, 52(1): 70–84.

Goeman, J. J. (**2011**). *Penalized R package*. R package version 0.9-35.

Goldstein, A. S., Huang, J., Guo, C., Garraway, I. P., and Witte, O. N. (**2010**). "Identification of a cell of origin for human prostate cancer." *Science*, 329(5991): 568–571.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (**1999**). "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring." *Science*, 286(5439): 531–537.

Graf, E., Schmoor, C., Sauerbrei, W., and Schumacher, M. (**1999**). "Assessment and comparison of prognostic classification schemes for survival data." *Statistics in Medicine*, 18(17-18): 2529–2545.

Grana, C., Chinol, M., Robertson, C., Mazzetta, C., Bartolomei, M., Cicco, C. D., Fiorenza, M., Gatti, M., Caliceti, P., and Paganelli, G. (**2002**). "Pretargeted adjuvant radioimmunotherapy with Yttrium-90-biotin in malignant glioma patients : A pilot study." *British Journal of Cancer*, 86: 207 – 212.

Griffiths-Jones, S., Grocock, R. J., van Dongen, S., Bateman, A., and Enright, A. J. (**2006**). "miRBase: microRNA sequences, targets and gene nomenclature." *Nucleic Acids Research*, 34(Database issue): D140–144.

Griffiths-Jones, S., Saini, H. K., van Dongen, S., and Enright, A. J. (**2008**). "miRBase: tools for microRNA genomics." *Nucleic Acids Research*, 36(Database issue): D154–D158.

Groce, C. M. (**2009**). "Causes and consequences of microRNA dysregulation in cancer." *Nature Reviews Genetics*, 10: 704–714.

Gunderson, K. L., Kruglyak, S., Graige, M. S., Garcia, F., Kermani, B. G., Zhao, C., Che, D., Dickinson, T., Wickham, E., Bierle, J., Doucet, D., Milewski, M., Yang, R., Siegmund, C., Haas, J.,

Zhou, L., Oliphant, A., Fan, J.-b., Barnard, S., and Chee, M. S. (**2004**). "Decoding Randomly Ordered DNA Arrays." *Genome Researcht*, 14(5): 870–877.

Guyon, I. and Elisseeff, A. (**2003**). "An Introduction to Variable and Feature Selection." *Journal of Machine Learning Research*, 3(7-8): 1157–1182.

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.

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

Hanahan, D. and Weinberg, R. A. (**2011**). "Hallmarks of Cancer: The Next Generatio." *Cell*, 144(5): 646–674.

Hastie, T., Tibshirani, R., and Friedman, J. (**2009**). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition.

Hastie, T. and Tibshirani, R. J. (**1986**). "Generalized Additive Models." *Statistical Science*, 1(3): 297–318.

Hill, S. M., Neve, R. M., Bayani, N., Kuo, W.-L., Ziyad, S., Spellman, P. T., Gray, J. W., and Mukherjee, S. (**2012**). "Integrating biological knowledge into variable selection: an empirical Bayes approach with an application in cancer biology." *BMC Bioinformatics*, 13(1): 94.

Hosmer, D. W., Lemeshow, S., and May, S. (**199**). *Applied Survival Analysis: Regression Modeling of Time to Event Data*. John Wiley & Sons.

Hothorn, T., Hornik, K., van de Wiel, M. A., and Zeileis, A. (**2011**). *coin: Conditional Inference Procedures in a Permutation Test Framework*. R package version 1.0-20.

Huang, G. T., Athanassiou, C., and Benos, P. V. (**2011**). "mirConnX: condition-specific mRNA-microRNA network integrator." *Nucleic Acids Research*, pages 1–8.

Irizarry, R. a. (**2003**). "Summaries of Affymetrix GeneChip probe level data." *Nucleic Acids Research*, 31(4): 8.

Ishwaran, H. and Kogalur, U. B. (**2007**). "Random Survival Forests for R." *R News*, 7(2): 25–31.

Ishwaran, H. and Kogalur, U. B. (**2010**). *Random Survival Forests*. R package version 3.6.3.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (**2008**). "Random survival forests." *The Annals of Applied Statistics*, 2(3): 841–860.

J. R. Quinlan (**1993**). *C4.5: programs for machine learning*. The Morgan Kaufmann series in machine learning. Morgan Kaufmann Publishers.

Jemal, a., Bray, F., Center, M. M., Ferlay, J., Ward, E., and Forman, D. (**2011**). "Global cancer statistics." *CA: A Cancer Journal for Clinicians*, 61(2): 69–90.

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.

John, B., Enright, A. J., Aravin, A., Tuschl, T., Sander, C., and Marks, D. S. (**2004**). "Human MicroRNA Targets." *PLoS Biology*, 2(11): 1862–1879.

Kamburov, A., Pentchev, K., Galicka, H., Wierling, C., Lehrach, H., and Herwig, R. (**2011**). "ConsensusPathDB: toward a more complete picture of cell biology." *Nucleic Acids Research*, 39(Database issue): D712–D717.

Kamburov, A., Wierling, C., Lehrach, H., and Herwig, R. (**2009**). "ConsensusPathDB–a database for integrating human functional interaction networks." *Nucleic Acids Research*, 37(Database issue): D623–D628.

Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M. (**2004**). "The KEGG resource for deciphering the genome." *Nucleic Acids Research*, 32(Database issue): D277–D280.

Kaplan, E. L. and Meier, P. (**1958**). "Nonparametric Estimation from Incomplete Observations." *Journal of the American Statistical Association*, 53(282): 457– 481.

Keshava Prasad, T. S., 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., Harrys Kishore, C. J., 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 Research*, 37(Database issue): D767–D772.

Khan, A. P., Poisson, L. M., Bhat, V. B., Fermin, D., Zhao, R., Kalyana-Sundaram, S., Michailidis, G., Nesvizhskii, A. I., Omenn, G. S., Chinnaiyan, A. M., and Sreekumar, A. (**2010**). "Quantitative proteomic profiling of prostate cancer reveals a role for miR-128 in prostate cancer." *Molecular & Cellular Proteomicst*, 9(2): 298–312.

Kim, V. N., Han, J., and Siomi, M. C. (**2009**). "Biogenesis of small RNAs in animals." *Nature Reviews Molecular Cell Biology*, 10(2): 126–139.

Kohavi, R. (**1995**). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *Proceedings of the International Joint Conference on Artificial Intelligence*.

Kohavi, R. and George H. John (**1997**). "Wrappers for feature subset selection." *Artificial Intelligence*, 97: 273–324.

Kohonen, T. (**1982**). "Self-organized formation of topologically correct feature maps." *Biological Cybernetics*, 43(1): 59–59.

Krek, A., Grün, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., da Piedade, I., Gunsalus, K. C., Stoffel, M., and Rajewsky, N. (**2005**). "Combinatorial microRNA target predictions." *Nature Genetics*, 37(5): 495–500.

Langley, P. (**1994**). "Selection of Relevant Features in Machine Learning." In "Proceedings of the AAAI Fall Symposium on Relevance," pages 140–144. AAAI Press, New Orleans, LA.

Lee, E., Chuang, H.-Y., Kim, J.-W., Ideker, T., and Lee, D. (**2008**). "Inferring pathway activity toward precise disease classification." *PLoS Computational Biology*, 4(11): e1000 217.

Lewis, B. P., Burge, C. B., and Bartel, D. P. (**2005**). "Conserved Seed Pairing , Often Flanked by Adenosines , Indicates that Thousands of Human Genes are MicroRNA Targets." *Cell*, 120(1): 15–20.

Li, C. and Li, H. (**2008**). "Network-constrained regularization and variable selection for analysis of genomic data." *Bioinformatics*, 24(9): 1175–1182.

Licata, L., Briganti, L., Peluso, D., Perfetto, L., Iannuccelli, M., Galeota, E., Sacco, F., Palma, A., Nardozza, A. P., Santonico, E., Castagnoli, L., and Cesareni, G. (**2012**). "MINT, the molecular interaction database: 2012 update." *Nucleic Acids Research*, 40(Database issue): D857–D861.

Lipshutz, R. J., Fodor, S. P. A., Gingeras, T. R., and Lockhart, D. J. (**1999**). "High density synthetic oligonucleotide arrays." *Nature Genetics*, 21(1 Suppl): 20–24.

Lloyd, S. P. (**1982**). "Least Squares Quantization in PCM." *IEEE Transactions on Information Theory*, 28: 129–137.

Loughin, T. (**2004**). "A systematic comparison of methods for combining p-values from independent tests." *Computational Statistics & Data Analysis*, 47(3): 467–485.

Lu, J., Getz, G., Miska, E. a., Alvarez-Saavedra, E., Lamb, J., Peck, D., Sweet-Cordero, A., Ebert, B. L., Mak, R. H., Ferrando, A. a.,

Downing, J. R., Jacks, T., Horvitz, H. R., and Golub, T. R. (**2005**). "MicroRNA expression profiles classify human cancers." *Nature*, 435: 834–838.

M. Stone (**1974**). "Cross-Validatory Choice and Assessment of Statistical Predictions." *Journal of the Royal Statistical Society, Series B*, 36(2): 111–147.

MacQueen, J. (**1967**). "Some Methods For Classification And Analysis Of Multivariate Observations." In "Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability," pages 281–297.

Martinetz, T. M., Berkovich, S. G., and Schulten, K. J. (**1993**). ""Neural Gas" Network for Vector Quantization and its Application to Time-Series Prediction." *IEEE Transactions on Neural Networks*, 4(4): 558–569.

Marx, B. D. and Eilers, P. H. C. (**1998**). "Direct generalized additive modeling with penalized likelihood." *Computational Statistics & Data Analysis*, 28: 193–209.

Matys, V., Kel-Margoulis, O. V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A. E., and Wingender, E. (**2006**). "TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes." *Nucleic Acids Research*, 34(Database issue): D108–D110.

Michiels, S., Kramar, A., and Koscielny, S. (**2011**). "Multidimensionality of microarrays: statistical challenges and (im)possible solutions." *Molecular Oncology*, 5(2): 190–196.

Mikeska, T., Bock, C., Do, H., and Dobrovic, A. (**2012**). "DNA methylation biomarkers in cancer: progress towards clinical implementation." *Expert Review of Molecular Diagnostics*, 12(5): 473–487.

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.

Nymark, P., Guled, M., Borze, I., Faisal, A., Lahti, L., Salmenkivi, K., Kettunen, E., Anttila, S., and Knuutila, S. (**2011**). "Integrative Analysis of microRNA , mRNA and aCGH Data Reveals Asbestos- and Histology-Related Changes in Lung Cancer." *Genes, Chromosomes & Cancer*, 50: 585–597.

Paik, S., Shak, S., Tang, G., Kim, C., Baker, J., Cronin, M., Baehner, F. L., Walker, M. G., Watson, D., Park, T., Hiller, W., Fisher, E. R., Wickerham, D. L., Bryant, J., and Wolmark, N. (**2004**). "A multigene assay to predict recurrence of tamoxifen-treated, node-negative breast cancer." *The New England Journal of Medicine*, 351(27): 2817–2826.

Pan, W., Xie, B., and Shen, X. (**2010**). "Incorporating Predictor Network in Penalized Regression with Application to Microarray Data." *Biometrics*, 66(2): 474–484.

Panagiotis, A., Manolis, M., L, P. G., Reczko, M., and G, H. A. (**2009**). "Lost in translation: an assessment and perspective for computational microRNA target identification." *Bioinformatics*.

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

Penault-Llorca, F., Bilous, M., Dowsett, M., Hanna, W., Osamura, R. Y., Rüschoff, J., and van de Vijver, M. (**2009**). "Emerging technologies for assessing HER2 amplification." *American Journal of Clinical Pathology*, 132(4): 539–548.

Porkka, K. P., Pfeiffer, M. J., Waltering, K. K., Vessella, R. L., Tammela, T. L. J., and Visakorpi, T. (**2007**). "MicroRNA expression profiling in prostate cancer." *Cancer Research*, 67(13): 6130–6135.

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 Research*, 38(suppl 1): D105–D110.

Porzelius, C. and Binder, H. (**2010**). *peperr: Parallelised Estimation of Prediction Error*. R package version 1.1-5.

Porzelius, C., Johannes, M., Binder, H., and Beissbarth, T. (**2011**a). "Leveraging external knowledge on molecular interactions in classification methods for risk prediction of patients." *Biometrical Journal*, 53(2): 190–201.

Porzelius, C., Schumacher, M., and Binder, H. (**2011**b). "The benefit of data-based model complexity selection via prediction error curves in time-to-event data." *Computational Statistics*, 26(2): 293–302.

Pruitt, K. D., Tatusova, T., Brown, G. R., and Maglott, D. R. (**2012**). "NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy." *Nucleic Acids Research*, 40(Database issue): D130–135.

Rajan, P., Elliott, D. J., Robson, C. N., and Leung, H. Y. (**2009**). "Alternative splicing and biological heterogeneity in prostate cancer." *Nature Reviews Urology*, 6(8): 454–460.

Rapaport, F., Zinovyev, A., Dutreix, M., Barillot, E., and Vert, J.-P. (**2007**). "Classification of microarray data using gene networks." *BMC Bioinformatics*, 8: 35.

Rehmsmeier, M., Steffen, P., Höchsmann, M., and Giegerich, R. (**2004**). "Fast and effective prediction of microRNA / target duplexes." *RNA*, 10(10): 1507–1517.

Ritchie, M. E., Silver, J., Oshlack, A., Holmes, M., Diyagama, D., Holloway, A., and Smyth, G. K. (**2007**). "A comparison of background correction methods for two-colour microarrays." *Bioinformatics*, 23(20): 2700–2707.

Ruppert, D. (**2002**). "Selecting the Number of Knots for Penalized Splines." *Journal of Computational and Graphical Statistics*, 11(4): 735–757.

Sales, G., Coppe, A., Bisognin, A., Biasiolo, M., Bortoluzzi, S., and Romualdi, C. (**2010**). "MAGIA, a web-based tool for miRNA and Genes Integrated Analysis." *Nucleic Acids Research*, 38: 352–359.

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., and Buetow, K. H. (**2009**). "PID: the Pathway Interaction Database." *Nucleic Acids Research*, 37(Database issue): D674–D679.

Schapire, R. E. (**1990**). "The Strength of Weak Learnability." *Machine Learning*, 5: 197–227.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (**1998**). "Boosting the Margin : A New Explanation for the Effectiveness of Voting Methods." *The Annals of Statistics*, 26: 1651–1686.

Schena, M., editor (**1999**). *DNA microarrays: A practical approach*. Oxford University Press, Oxford.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (**1995**). "Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray." *Science*, 270(5235): 467–470.

Schölkopf, B. and Smola, A. J. (**2002**). *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, Camebridge, Massachusette.

Seal, R. L., Gordon, S. M., Lush, M. J., Wright, M. W., and Bruford, E. a. (**2011**). "genenames.org: the HGNC resources in 2011." *Nucleic Acids Research*, 39(Database issue): D514–9.

Smyth, G. K. (**2005**). "Limma: linear models for microarray data." In Gentleman, R., 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.

Smyth, G. K. and Speed, T. (**2003**). "Normalization of cDNA microarray data." *Methods*, 31(4): 265–273.

Sotiriou, C. and Piccart, M. J. (**2007**). "Taking gene-expression profiling to the clinic: when will molecular signatures become relevant to patient care?" *Nature Reviews Cancer*, 7(7): 545–553.

Stekel, D. (**2003**). *Microarray Bioinformatics*. Cambridge University Press.

Stouffer, S., Suchman, E., DeVinney, L., Star, S., and Williams, R. J. (**1949**). *The American Soldier, Vol. 1: Adjustment during Army Life*. Princeton University Press, Princeton.

Strachan, T. and Read, A. P. (**2005**). *Molekulare Humangenetik*. Elsevier, München, 3rd edition.

Tableman, M. and Kim, J. S. (**2004**). *Survival Analysis Using S: Analysis of Time-To-Event Data*. Chapman & Hall/CRC, Boca Raton.

Tannock, I. F., de Wit, R., Berry, W. R., Horti, J., Pluzanska, A., Chi, K. N., Oudard, S., Théodore, C., James, N. D., Turesson, I., Rosenthal, M. A., and Eisenberger, M. A. (**2004**). "Docetaxel plus prednisone or mitoxantrone plus prednisone for advanced prostate cancer." *The New England Journal of Medicine*, 351(15): 1502–1512.

Taylor, B. S., Schultz, N., Hieronymus, H., Gopalan, A., Xiao, Y., Carver, B. S., Arora, V. K., Kaushik, P., Cerami, E., Reva, B., Antipin, Y., Mitsiades, N., Landers, T., Dolgalev, I., Major, J. E., Wilson, M., Socci, N. D., Lash, A. E., Heguy, A., Eastham, J. a., Scher, H. I., Reuter, V. E., Scardino, P. T., Sander, C., Sawyers, C. L., and Gerald, W. L. (**2010**). "Integrative Genomic Profiling of Human Prostate Cancer." *Cancer Cell*, 18: 1–12.

The Gene Ontology Consortium, 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." *Nature Genetics*, 25(1): 25–29.

Tibshirani, R. (**1996**). "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society, Series B*, 58(1): 267–288.

Tibshirani, R. (**1997**). "The lasso method for variable selection in the Cox model." *Statistics in Medicine*, 16: 385–395.

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 of the United States of America*, 99(10): 6567–6572.

Tosoian, J. and Loeb, S. (**2010**). "PSA and beyond: the past, present, and future of investigative biomarkers for prostate cancer." *TheScientificWorldJournal*, 10: 1919–31.

Tusher, V. G., Tibshirani, R., and Chu, G. (**2001**). "Significance analysis of microarrays applied to the ionizing radiation response." *Proceedings of the National Academy of Sciences of the United States of America*, 98(9): 5116–5121.

Tutz, G. and Binder, H. (**2006**). "Generalized additive modeling with implicit variable selection by likelihood-based boosting." *Biometrics*, 62(4): 961–971.

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: 530–536.

Vannucci, M. and Stingo, F. C. (**2010**). "Bayesian Models for Variable Selection that Incorporate Biological Information." In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, "Bayesian Statistics 9 - Proceedings of the Ninth Valencia International Meeting," pages 659–679. Oxford University Press, Oxford.

Vapnik, V. (**1999**). *The nature of statistical learning theory*. Springer, New York, 2nd ed. edition.

Venables, J. P. (**2004**). "Aberrant and alternative splicing in cancer." *Cancer Research*, 64(21): 7647–54.

Venet, D., Dumont, J. E., and Detours, V. (**2011**). "Most Random Gene Expression Signatures Are Significantly Associated with Breast Cancer Outcome." *PLoS Computational Biology*, 7(10): e1002 240.

Voet, D. and Voet, J. G. (**2004**). *Biochemistry*. Jon Wiley & Sons, 3rd edition.

Walt, D. R. (**2000**). "Bead-based Fiber-Optic Arrays." *Science*, 287(5452): 451–452.

Wang, X., Kruithof-de Julio, M., Economides, K. D., Walker, D., Yu, H., Halili, M. V., Hu, Y.-P., Price, S. M., Abate-Shen, C., and Shen, M. M. (**2009**). "A luminal epithelial stem cell that is a cell of origin for prostate cancer." *Nature*, 461(7263): 495–500.

Wang, Y., Klijn, J. G. M., Zhang, Y., Sieuwerts, A. M., Look, M. P., Yang, F., Talantov, D., Timmermans, M., Meijer-van Gelder, M. E., 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: 671–679.

Wei, Z. and Li, H. (**2007**). "Nonparametric pathway-based regression models for analysis of genomic data." *Biostatistics*, 8(2): 265–284.

Weinberg, R. A. (**2007**). *The biology of cancer*. Garland Science, Taylor & Francis Group, LLC, New York.

Weston, J., Elisseeff, A., Schölkopf, B., and Tipping, M. (**2003**). "Use of the Zero-Norm with Linear Models and Kernel Methods." *Journal of Machine Learning Research*, 3: 1439–1461.

Wit, E. and McClure, J. D. (**2004**). *Statistics for microarrays: design, analysis, and inference*. John Wiley & Sons.

Wu, L., Fan, J., and Belasco, J. G. (**2006**). "MicroRNAs direct rapid deadenylation of mRNA." *Proceedings of the National Academy of Sciences of the United States of America*, 103(11): 4034–4039.

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.

Wuchty, S., Fontana, W., Hofacker, I. L., and Schuster, P. (**1999**). "Complete Suboptimal Folding of RNA and the Stability of Secondary Structures." *Biopolymers*, 49: 145–165.

Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J., and Speed, T. P. (**2002**). "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation." *Nucleic Acids Research*, 30(4): e15.

Zaykin, D. V., Zhivotovsky, L. A., Westfall, P. H., and Weir, B. S. (**2002**). "Truncated product method for combining P-values." *Genetic Epidemiology*, 22(2): 170–185.

Zhu, Y., Shen, X., and Pan, W. (**2009**). "Network-based support vector machine for classification of microarray samples." *BMC Bioinformatics*, 10(Suppl 1): S21.

Zou, H. and Hastie, T. (**2005**). "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society, Series B*, 67(2): 301–320.

Stephan Gade

Personal Information

Date of Birth	04 March 1981
Place of Birth	Jena, Germany
Nationality	German
Address	Im Eschbachtal 13 61352 Bad Homburg
eMail	stephan.gade@gmail.com

Education

10/2000 – 07/2008	**Diplom Informatiker**, Eberhard Karls University Tübingen Major Field of Study: Bioinformatics Diploma thesis: "Visualization and clustering of the expression content of multiple groups using barycentric coordinates"
12/2008 – 12/2012	**PhD in Bioinformatics**, University of Göttingen / German Cancer Research Center Heidelberg Major Field of Study: Bioinformatics/Biostatistics PhD thesis: "Graph based fusion of high-dimensional gene- and microRNA expression data"

Professional Experience

12/2008 – 03/2012	**Bioinformatician/Biostatistician**, German Cancer Research Center Heidelberg Topics: Data Integration, Statistical Analysis of High-Throughput Data, Data Visualization, Machine Learning
since 04/2012	**Bioinformatician**, GBG Forschungs GmbH Topics: Statistical Analysis of High-Throughput Data especially Sequencing Data

Publications

Brase, J. C., Johannes, M., Mannsperger, H., Fälth, M., Metzger, J., Kacprzyk, L. a., Andrasiuk, T., Gade, S., Meister, M., Sirma, H., Sauter, G., Simon, R., Schlomm, T., Beissbarth, T., Korf, U., Kuner, R., and Sültmann, H. (2011). TMPRSS2-ERG-specific transcriptional modulation is associated with prostate cancer biomarkers and TGF-beta signaling. *BMC Cancer*, 11(1):507.

Brase, J. C., Mannsperger, H., Fröhlich, H., Gade, S., Schmidt, C., Wiemann, S., Beissbarth, T., Schlomm, T., Sültmann, H., and Korf, U. (2010). Increasing the sensitivity of reverse phase protein arrays by antibody-mediated signal amplification. *Proteome Science*, 8:10.

Gade, S., Porzelius, C., Maria, F., Brase, J. C., Wuttig, D., Kuner, R., Binder, H., Holger, S., and Beissbarth, T. (2011). Graph based fusion of miRNA and mRNA expression data improves clinical outcome prediction in prostate cancer. *BMC Bioinformatics*, 12:488.

Gogolin, S., Batra, R., Harder, N., Ehemann, V., Paffhausen, T., Diessl, N., Sagulenko, V., Benner, A., Gade, S., Nolte, I., Rohr, K., König, R., and Westermann, F. (2012). MYCN-mediated overexpression of mitotic spindle regulatory genes and loss of p53-p21 function jointly support the survival of tetraploid neuroblastoma cells. *Cancer Letters*, (December):11.

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.

Kuner, R., Fälth, M., Pressinotti, N. C., Brase, J. C., Puig, S. B., Metzger, J., Gade, S., Schäfer, G., Bartsch, G., Steiner, E., Klocker, H., and Sültmann, H. (2012). The maternal embryonic leucine zipper kinase (MELK) is upregulated in high-grade prostate cancer. *Journal of Molecular Medicine*, pages 1–12.

Mannsperger, H., Gade, S., Henjes, F., Beissbarth, T., and Korf, U. (2010). RPPanalyzer: Analysis of reverse phase protein array data. *Bioinformatics*, 26(17):2202–2203.

Zacher, B., Abnaof, K., Gade, S., Younesi, E., Tresch, A., and Fröhlich, H.

(2012). Joint Bayesian inference of condition-specific miRNA and transcription factor activities from combined gene and microRNA expression data. *Bioinformatics*, 28(13):1714–1720.