INAUGURAL - DISSERTATION

zur Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Diplom-Informatiker der Medizin
Falk Schubert
aus Mittweida

Tag der mündlichen Prüfung: 21. Januar 2010

Machine learning of genomic profiles

Dekan:	Professor Dr. R. Rannacher, Universität Heidelberg
Erster Gutachter:	Professor Dr. Roland Eils, Universität Heidelberg
Zweiter Gutachter:	Professor Dr. Fred Hamprecht, Universität Heidelberg

Abstract

Machine learning is an area of computer science concerned with the study of algorithms that reveal patterns and rules from data sets. Genomic profiles describe alterations of a genome, like copy number variations. Cancer often originates from a combination of genomic alterations.

In this thesis, I consider machine learning and its application to genomic profiles. The main aspects of this work can be summarised as follows:

First, I discuss several machine learning methods, with particular regard to genomic profiles, and then develop a special loss function for survival data.

Next, I introduce a framework to find aberration patterns associated with a particular tumour type or disease state. This workflow starts with preprocessing, feature selection and discretisation of genomic profiles, includes strategies to deal with missing values and provides a multi-resolutional analysis. Then, training and analysis of a classifier is performed.

Additionally, I introduce an explanation component that emphasizes important features of the classification process and estimates the certainty of classification results. Such an explanation method could provide the basis for the integration of a classification algorithm, such as a support vector machine, in a clinical decision support system.

The methods proposed in the thesis were applied to various data sets, focussing on important biological questions, such as early metastasis and micrometastasis, and lead to the detection of new tumour markers.

The results of these investigations indicate that machine learning methods can enhance our understanding of genomic aberrations and may help to improve the delivery of therapies to cancer patients.

Zusammenfassung

Gegenstand dieser Arbeit ist das maschinelle Lernen und seine Anwendung auf genomische Profile.

Maschinelles Lernen ist ein Teilbereich der Informatik, der sich mit der Analyse und dem Design von Algorithmen beschäftigt, die Regeln und Muster aus Datensätzen ableiten. Genomische Profile beschreiben Veränderungen der DNA, z.B. der Anzahl ihrer Kopien. Tumorerkrankungen werden oftmals von diesen genomischen Veränderungen hervorgerufen.

Es werden verschiedene Verfahren des maschinellen Lernens auf ihre Anwendbarkeit in Bezug auf genomische Profile untersucht. Des Weiteren wird eine Verlustfunktion für Überlebenszeitdaten entworfen.

Anschließend wird ein analytischer Bezugsrahmen entwickelt, um Aberrationsmuster zu finden, die mit einer speziellen Tumorerkrankung assoziiert sind. Der Bezugsrahmen umfaßt die Vorverarbeitung, Merkmalsselektion und Diskretisierung von genomischen Profilen sowie Strategien zum Umgang mit fehlenden Werten und eine mehrdimensionale Analyse. Abschließend folgen das Training und die Analyse des Klassifikators.

In dieser Arbeit wird weiterhin eine Erklärungskomponente vorgestellt, die wichtige Merkmale für die Klassifikation eines Falles identifiziert und ein Maß für die Richtigkeit einer Klassifikation liefert. Solch eine Erklärungskomponente kann die Basis für die Integration eines Klassifikators , z.B. einer Support-Vektor-Maschine, in ein entscheidungsunterstützendes System sein.

Die im Rahmen dieser Arbeit entwickelten Methoden wurden erfolgreich zur Beantwortung von biologischen Fragestellungen wie der frühen Metastasierung oder der Mikrometastasierung angewandt und führten zur Entdeckung bisher unbekannter Tumormarker.

Zusammenfassend zeigen die Ergebnisse der vorliegenden Arbeit, dass Verfahren des maschinellen Lernens zum Erkenntnisgewinn in Bezug auf genomische Veränderungen beitragen und Möglichkeiten zu einer weiteren Verbesserung der Therapie für Tumorpatienten aufzeigen.

Acknowledgements

Foremost, I would like to thank Professor Dr. Roland Eils for having provided me with the opportunity to work in the stimulating environment of the Department of Theoretical Bioinformatics at the Deutsches Krebsforschungszentrum (German Cancer Research Center) and for supervising my doctoral work at the Ruprecht-Karls-Universität Heidelberg (University of Heidelberg), Naturwissenschaftlich-Mathematische Gesamtfakultät (Combined Faculty of the Natural Sciences and Mathematics).

Grateful thanks are also due to Professor Dr. Fred Hamprecht for being the second supervisor of my thesis and for stimulating lectures.

Thanks also to all my other colleagues in the Theoretical Bioinformatics group, with whom I had the pleasure to work. I am much obliged to Dr. Benedikt Brors, Dr. Rainer König, Patrick Warnat, Dr. Marc Zapatka, and Jasmin Müller for useful discussions, constructive criticism and proofreading.

Furthermore, special thanks go to Dr. Jan Wiemer for many inspiring discussions, and to Karlheinz Groß and Rolf Kabbe for providing an excellent infrastructure.

I want to thank my collaborators and co-authors Dr. Björn Fritz, PD Dr. Stefan Joos, Professor Dr. Christoph Klein, Professor Dr. Peter Lichter, Dr. Gunhild Mechtersheimer, Jasmin Müller, Professor Dr. Klaus Panthel, Dr. Bernhard Radlwimmer, Daniel Stange, Bernhard Tausch and Dr. Ute Wölfle.

Thanks to Dr. Matthias Ebert for providing me with the LaTeX style sheets and to Stefan Skonetzki for valuable comments.

Moreover, I gratefully thank Maria Haughey for proof-reading a draft of this dissertation in a cosy cafe. Any remaining mistakes are entirely my own.

I express my appreciation for the support I received from my parents, Ian Wood and Thomas Zimmerling.

Finally, the peace at St. Marienthal Convent was invaluable to me as I was writing parts of this thesis.

Cambridge/Heidelberg, April 2008

Contents

Chapter 1

Introduction

1.1 Overview

Cancer can be caused by, and often correlates with, a combination of genomic alterations. A better understanding of these genomic alterations will lead to improved diagnostic schemes and finally to additional therapeutic interference points [WLS+01, SKS+02, Saw03].

In the last two decades, a manifold of experimental techniques has emerged to investigate genomic alterations in cancer patients at different levels of resolution. Currently, techniques such as comparative genomic hybridisation (CGH), loss of heterozygosity (LOH) and especially matrix-CGH, are revolutionising cancer research.

CGH, matrix-CGH and LOH can provide a comprehensive, genome-wide analysis of genomic alterations and describe the genomic profile of each tumour. The dimensionality and complexity of genomic profiles demand the application of computer-based methods. Important methods for analysing and modelling the underlying structure of genomic aberrations originated from the field of machine learning.

Questions that can be answered by machine learning are:

- What are the typical aberrations of a specific tumour type?
- Are different tumour types distinguishable?
- Which aberrations distinguish one tumour type from another?

- Are genomic aberrations associated with certain clinical variables, such as tumour grading, lymph node status, age, gender and tumour size?
- Can the (relapse, metastasis or survival) risk of an individual patient be estimated based upon genomic profiles, and how?
- Can genomic profiles be used to tailor the appropriate therapy for a specific patient in cancer treatment, and how?

1.2 How machines learn

Machine learning is an interdisciplinary field including, but not limited to: artificial intelligence (AI), statistics and information theory.

The term "machine", as utilised in AI, refers to a program running on a multi-purpose computer, rather than to a mechanical device.

Historically, one main motivation of AI has been to develop a tool that can solve all problems posed - in essence, a general problem solver [NS61]. However, this ambitious goal has not been fulfilled and the direction of research has increasingly shifted towards using AI for the solution of well-defined problems in separable domains [McC04, RN95].

One well-defined problem is the automatic assignment of pattern (e.g., pictures) to classes (e.g. category apples versus category peaches). This has been a goal of AI from the beginning. For example, in 1955, Selfridge and Dinneen [Sel55, Din55] presented a model of a system which should learn to recognise visual patterns like the letters "A" and "O". Selfridge proposed to "feed the machine A's and O's, telling the machine each time which letter it is." Furthermore, he suggested searching assignment rules from randomly chosen combinations of basis operations (basis functions) such that they discriminate A's and O's.

Machine learning reveals mathematical assignment rules from a set of classified example objects (e.g., pictures of apples and pictures of peaches). Finally, a new object can be automatically assigned to one class or the other (e.g., apples or peaches).

Learning based on examples relies on the induction principle of philosophy. Induction means that general rules (laws) about future observations are inferred from a limited number of observations in the past. But "what is the justification for the belief that the future will resemble the past?" ([Pop72]).

If we observed only green apples in the past, we would conclude that all apples are green. In fact, a machine learning algorithm would generalise from a training set of green apples and red peaches that all green objects are apples and all red objects are peaches. However, if we would apply the learned classifier to red apples in the future, they may be detected as peaches. The underlying problem of stationarity will be covered in section 2.2.

Machine learning involves searching through a space of available assignment rules (hypotheses). The goal is to find the simplest hypothesis that fits the available data and prior knowledge. Problems directly derived from this definition are:

- What is the space of available hypotheses (function space)?
- What is a useful search strategy?
- How can the quality of fit between data and hypotheses be estimated?
- How can the simplicity/complexity of a hypothesis be assessed?

These problems will be discussed further in chapter 2.

In addition to traditional issues of machine learning, I consider here also the task of explaining the results derived from machine learning to a user who might not be familiar with this field. Imagine a classifier calculated that an object is an apple. So we may ask the questions:

- Why did the classifier decide that this object is an apple?
  What are the most important features for classifying this object as an apple?
- What is the probability that this object is an apple?
- Is the classifier competent to make this decision? Has it ever been trained with apples? Is the induction from example objects possible?

The answers to these questions can lead to an integration of models discovered by machine learning into decision support systems and will be discussed in section 4.4.1.

All machine learning algorithms applied in this thesis can be efficiently simulated by a Turing machine and multi-purpose digital computers. Interestingly, we can also consider learning algorithms derived from a class of recurrent artificial networks (ARRN) with real weights (and infinite precision) that cannot be computed by Turing machines, but may mimic classification problems of natural phenomena [Sie99]. However, this is not only beyond the "Turing limit", but also beyond the scope of this thesis.

1.3 Outline

The remainder of this thesis is organised into five major parts.

Chapter 2 reviews some basics of machine learning and statistical learning theory.

Chapter 3 provides some fundamental insights into the biology of cancer, genomic aberrations, genomic profiles and the underlying experiments. The focus of the genomic profiles under consideration clearly lies in the field of cancer research.

Chapter 4 derives a workflow for machine learning of genomic profiles. This workflow includes data preprocessing, feature selection, classifier design and evaluation. An explanation component, which emphasises important features of the classification process and estimates the certainty of classification results, is introduced.

Chapter 5 presents various results where this workflow has been successfully applied.

Chapter 6 discusses and concludes the results. Finally, future directions of research are outlined.

This thesis includes material published in [FSW+02, WSG+03, VKM+04, SMF+03, WBZ+05, STJE05, SMH+05, SRS+06, HGS+08].

Chapter 2

Machine learning

2.1 Introduction to machine learning

This chapter gives an overview of known machine learning methods as far as they are of importance in the domain of genomic profiles. My own contribution consists of a formulation of an SVM for survival data (section 2.4.3).

Machine learning is an interdisciplinary field based on artificial intelligence, (Bayesian) statistics, control theory, information theory, complexity theory, psychology and philosophy [Mit97].

The focus of machine learning under consideration is classification. The task of classification can be defined as finding a rule which assigns classes to objects due to their observed properties. These rules are mathematical functions which are learnt from objects with observed properties and known class assignments. Finally, the decision function should assign the correct class to unseen objects.

The objects of classification problems in the domain of genomic profiles are often cases (e.g., patients, distinct tumour samples from patients, animals, cell culture samples). Let us define a genomic profile for the moment as a real-valued vector, each element representing a so-called copy number ratio (see chapter 3 for a detailed discussion). Properties of objects (e.g., colour) are called features and their observed values (e.g., red) feature values. The decision function is sometimes also called a learning machine or a learner.

The formal setting of this learning problem can be formalised as follows [Vap95, Mik02, Rät01, Sch97, MMR+01].

2.2 Formal setting

Given is a training set $\mathbb{S} = \{(\boldsymbol{x_i}, y_i) \in \mathbb{X} \times \mathbb{Y} | i = 1, ..., M\}$ of M cases. Each $\boldsymbol{x_i}$ is an input or feature vector. $\mathbb{X}$ is the input space, often $\mathbb{R}^N$. Each dimension of $\mathbb{X}$ represents a feature A_i, $i = 1, ..., N$. The nature of the target variable $\mathbb{Y}$ depends on the problem to be solved. A regression problem is characterised by $\mathbb{Y} = \mathbb{R}$ whereas a classification problem with K classes is defined by $\mathbb{Y} = \{1, ..., K\}$. For the special case of $K = 2$ the class labels are defined as $\mathbb{Y} = \{-1, +1\}$.

In the domain of this thesis, the set $\mathbb{X}$ describes genomic profiles whereas $\mathbb{Y}$ refers to different tumour types. The goal is often to find a learner modelling the relationship between features (genomic profiles) and (tumour) classes. This can also be considered as estimating a function $f(\boldsymbol{x}) : \mathbb{X} \rightarrow \mathbb{Y}$. For a two class problem, a classified case is assigned to class $+1$ if $f(\boldsymbol{x}) \geq 0$ and to the class -1 if $f(\boldsymbol{x}) < 0$. An example of such a decision function would be a hyperplane $f(\boldsymbol{x}) = \text{sign}(\langle \boldsymbol{\omega}, \boldsymbol{x} \rangle + b)$, with a hyperplane direction vector $\boldsymbol{\omega}$ and a constant b.

We assume that there exists an unknown probability distribution over $\mathbb{X} \times \mathbb{Y}$ with density $p(\boldsymbol{x}, y)$ that describes our data generating process. Our sample set $\mathbb{S}$ is now drawn identically and independently distributed (i.i.d) from this distribution. An example distribution would consist of the genomic profiles of all breast cancer patients in Germany from 1990 to 2000. We require that the distribution is stationary. This implies that the typical genomic profiles of patients at the time of surgery should not change over time.

This assumption seems to be very strong for our application. Often we discover experimentally caused changes (e.g., new therapeutic guidelines, increasing expert knowledge of an experimenter). Finally, I propose a competence estimation (section 4.4.2) to deal with this problem.

Once we obtain a decision function $f(\boldsymbol{x})$, we would like to evaluate it. How can the decision function be evaluated? How can the quality of fit between the training data $\mathbb{S}$ and the decision function be estimated?

We will also see later that the construction of an optimal decision function depends on the chosen criteria for the evaluation of a decision function.

2.3 Model assessment

2.3.1 Expected risk and loss functions

Ideally, the decision function should always assign the correct class to a case. However, if two cases share the same feature values but belong to different classes, both classes are not always separable.

Therefore, our goal is to find the function f_* from the function space $\mathfrak{F}$ that minimises the number of wrong assignments, or formally the expected risk (generalisation error) $R(f)$:

$$f_* = \arg\min_{f \in \mathfrak{F}} R(f). \tag{2.3.1}$$

This error is defined on the entire distribution (e.g. of all breast cancer cases) and not only on the cases observed:

$$R(f) = \int \mathcal{L}(y, f(\boldsymbol{x}))dp(\boldsymbol{x}, y). \tag{2.3.2}$$

$\mathcal{L}$ is a loss function, measuring the difference between the predicted ($f(\boldsymbol{x})$) and the true (y) outcome. Examples of loss functions used in this thesis are given in table 2.1.

2.3.2 Empirical risk

However, we do not know the underlying probability density $p(\boldsymbol{x}, y)$. But we can try to estimate a so-called empirical risk, based on the available cases $\mathbb{S}$:

$$R_{\text{emp}}(f, \mathbb{S}) = \frac{1}{M} \sum_{n=1}^{M} \mathcal{L}(y_n, f(\boldsymbol{x_n})). \tag{2.3.3}$$

A minimisation of the empirical risk can lead to overfitting (Fig. 2.1). In the case of overfitting, a complex decision function is fitted to the set of available cases such that the decision function reflects the training data and not the entire distribution. Overfitting can often be avoided by choosing simpler decision functions (e.g., linear or quadratic). Occam's razor, a general

Name	Description	Definition
Squared ($L2$) loss	Common loss function for regression problems	$\mathcal{L}_2(y, f(\boldsymbol{x})) = (f(\boldsymbol{x}) - y)^2$
0/1-loss	Empirical model assessment of classification problems	$\mathcal{L}_{0/1}(y, f(\boldsymbol{x})) = \begin{cases} 0 & \text{for } y = f(\boldsymbol{x}) \\ 1 & \text{for } y \neq f(\boldsymbol{x}) \end{cases}$
$L1$ loss	Derivation of boosting (see section 2.4.7)	$\mathcal{L}_1(y, f(\boldsymbol{x})) = \lvert f(\boldsymbol{x}) - y \rvert$
ϵ-insensitive $L1$ loss	Derivation of support vector regression	$\mathcal{L}_\epsilon(y, f(\boldsymbol{x})) = \max(0, \lvert y - f(\boldsymbol{x}) \rvert - \epsilon)$

Table 2.1: Loss functions

principle of inductive learning, states that the most likely hypothesis is the simplest one that is consistent with all observations.

Subsequently, the estimation of the expected risk can be improved by considering the complexity of the decision function.

Complex classifiers can be penalised by regularisation and the regularised empirical risk is given by:

$$R_{\text{reg}} = R_{\text{emp}}(f, \mathbb{S}) + C\Phi(f). \tag{2.3.4}$$

C is a penalty factor, balancing empirical risk and classifier complexity. The appropriate choice of C is critical. $\Phi : \mathfrak{F} \to \mathbb{R}_+$ denotes a regularisation operator, measuring the complexity or smoothness of the decision function $f(\boldsymbol{x})$.

Finally, it seems worthwhile to include the complexity of the decision function in the model assessment. This can be done by calculating the minimum description length.

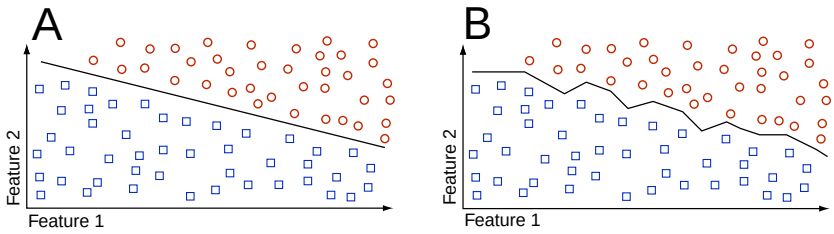


Figure 2.1: Two hypotheses were fitted to the training data. Hypothesis A uses a simpler decision function and is therefore much more likely to reflect the underlying distribution. Hypothesis B is likely to be overfitted.

2.3.3 Minimum description length

The idea of the minimal description length is to write the shortest binary program to reproduce a given data set [Ris78]. Surely, using regularities in the data set will help to find the shortest description and this underlying idea is used in compression algorithms. It's worthwhile to note that a Bayesian view of the minimal description length and a close link to coding theory also exists (see [Mac03]).

The minimum description length is defined as the shortest description (in bits) of the decision function: $L(f(\boldsymbol{x}))$ and the residuals of the training data given the decision function: $L(\mathbb{S}|f(\boldsymbol{x}))$. The model with the shortest description is defined by:

$$f_* = \arg\min_{f \in \mathfrak{F}} (L(f(\boldsymbol{x})) + L(\mathbb{S}|f(\boldsymbol{x}))). \tag{2.3.5}$$

A Bayesian (statistical) notation can be easily derived by maximising the probability $P(f|\mathbb{S})$ and using Bayes' law $P(f|\mathbb{S}) = \frac{P(\mathbb{S}|f)P(f)}{P(\mathbb{S})}$. Note that $P(\mathbb{S})$ is known for a given data set $\mathbb{S}$.

$$f_* = \arg\min_{f \in \mathfrak{F}} (-\log_2(P(f(\boldsymbol{x}))) - \log_2(P(\mathbb{S}|f(\boldsymbol{x})))) \tag{2.3.6}$$

The minimum description length can be used to compare different decision functions. It balances the complexity of the decision function, derived from the training data, and the empirical risk. A complex decision function has a larger code length than a simpler one. If the decision function is too simple, the empirical risk is higher and a larger code length is needed to encode the residuals of the training data.

2.3.4 Vapnik-Chernovenkis dimension

Another method to measure the complexity/capacity of decision functions is the VC-dimension [Vap95].

The VC-dimension θ of a class of functions $\mathfrak{F}$ corresponds to the maximum number of cases which can be shattered (learned with training error R_{emp}=0) by elements of this class for all possible labellings (class assignments) of these cases (Fig. 2.2).

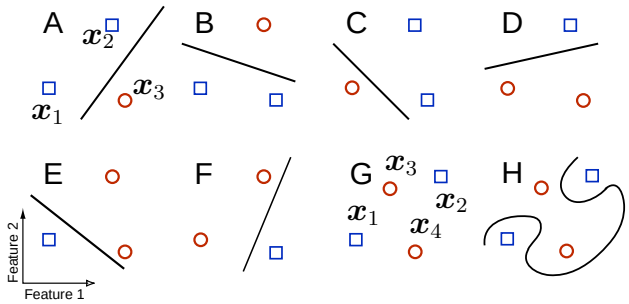


Figure 2.2: A) to F) A set of hyperplanes with two dimensions can shatter three cases $\boldsymbol{x}_1$ to $\boldsymbol{x}_3$ and has therefore a VC-dimension of three. Note that $\boldsymbol{x}_3$ belongs to class 2 ($y_i = -1$) in A), to class 1 ($y_i = +1$) in B) and to class 1 ($y_i = +1$) in C). Panel G) illustrates a class assignment for four points that cannot be separated by a linear hyperplane. However, H) shows that other nonlinear classifiers with a VC-dimension of at least four can still shatter (separate) four cases.

With a probability of at least $1 - \delta$, $f \in \mathfrak{F}$, $\delta > 0$ and R_{emp} be defined by $\mathcal{L}_{0/1}$, it holds that [Vap95]:

$$R(f) \leq R_{\text{emp}}(f, \mathbb{S}) + \sqrt{\frac{8}{M}\left[\theta\left(\log_2 \frac{2M}{\theta} + 1\right) + \log_2\left(\frac{4}{\delta}\right)\right]}. \qquad (2.3.7)$$

Therefore, a small VC-dimension θ prevents overfitting. The VC-dimension can also be used to derive support vector machines (see section 2.4).

Examples of VC-dimensions are:

- Class of hyperplanes with N dimensions: VC-dimension N+1.
- Support vector machines with radial basis kernels: infinite VC-dimension.

Unfortunately in practice, the bounded risk in (2.3.7) "is often neither easily computable nor very helpful" [MMR+01].

Finally, the simplest idea to evaluate a classifier is to apply it to unseen data.

2.3.5 Independent test set

The test error estimates the expected risk and is calculated from the number of classification errors on an independent test set.

The test set is created by randomly dividing the data set of all observed cases in three disjoint sets:

- a training set,
- a validation set and
- a test set.

The training set is used for classifier training, the validation set for estimating the expected risk of different classifiers and the test set for estimating the expected risk of the finally chosen classifier. See also section 4.3.

2.3.6 Cross-validation

If only a few cases are available then it is impossible to set aside an independent test set for model assessment. Instead, we can subdivide the available data into k approxiametely equally sized parts and use $(k-1)$ parts for training and one part for testing. Iteratively, each part is used once for testing and $(k-1)$ times for training. This procedure is called k-fold cross-validation.

The error estimation is then averaged over all parts, with $f^{-k(i)}(\boldsymbol{x_i})$ being the classifier trained on all but the ith part and n_k being the number of cases in part k:

$$R_{cross} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_k}\sum_{j=1}^{n_k}\mathcal{L}(y_{ij}, f^{-k(i)}(\boldsymbol{x_{ij}})). \tag{2.3.8}$$

Kohavi proposed a stratified cross-validation, where each part includes the same proportion of cases from each class [Koh95a].

If $K = M$ then cross-validation is called leave-one-out-cross-validation (LOO-CV). LOO-CV has a lower bias but a higher variance compared with a cross-validation with K=10 (ten-fold cross-validation) [HTF01].

An alternative algorithm, especially for small data sets, is called bootstrap.

2.3.7 Bootstrap

First, a random sample is drawn with replacement from the available data. The cases that are not part of this sample are called out-of-the-bag data. The sample contains approximately 63.2% of all cases. The classifier is trained with the random sample and tested on the out-of-the-bag data. This process is repeated and an average test error R_{test} is estimated.

Finally, the bootstrap error is calculated as a weighted average of the test error R_{test} and the training error on the whole dataset R_{emp}:

$$R_{boot} = 0.632 R_{test} + (1 - 0.632) R_{emp}. \tag{2.3.9}$$

The factor 0.632 is only based on plausibility and should be adapted in case of overfitting by considering a "no information error rate" [HTF01].

An empirical comparison of model assessment techniques and remarks on their reproducibility can be found in section 4.3.3.

2.4 Support vector machines

2.4.1 Key ideas

Following the definition of the VC-dimension (2.3.7), two different kinds of constructive learning algorithms can be considered [Vap95]:

1. Construct a machine with a given VC-dimension and minimise the empirical risk. This principle is implemented in artificial neural networks.
2. Construct a machine with a given empirical risk (e.g., $R_{emp} = 0$) and minimise the VC-dimension. This principle is used in support vector machines.

The support vector machine (SVM) classification algorithm combines two key ideas [Vap95].

The first idea of SVMs is to find an optimal separating hyperplane $f(\boldsymbol{x}) = \text{sign}(\langle \boldsymbol{\omega}, \boldsymbol{x} \rangle + b)$ with bias b that maximises the margin ρ between the two classes (e.g. tumour types). The margin ρ is defined as the minimum Euclidean distance of a case $\boldsymbol{x}_i$ from the decision hyperplane (see Fig. 2.3) and can

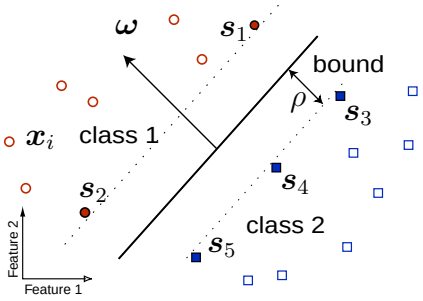


Figure 2.3: Linear decision function separating class 1 from class 2. Each circle or square refers to one case $\boldsymbol{x}_i$. Full circles and squares denote the support vectors $\boldsymbol{s}_1$ to $\boldsymbol{s}_5$. The margin is the distance from the hyperplane to a support vector.

be maximised by minimising the length of the hyperplane direction vector $\boldsymbol{\omega}$. Fig. 2.4 motivates the optimisation problem.

More formally, the VC-dimension θ of a separating hyperplane is bounded by the norm of the hyperplane direction vector $||\boldsymbol{w}||_2$:

$$\theta \leq \min\{r^2||\boldsymbol{w}||_2^2, M\} + 1 \tag{2.4.1}$$

where r is the radius of the smallest sphere around the training data. Note that r is fixed for a given data set $\mathbb{S}$. Furthermore, it is required that $R_{emp} = 0$ or $y_n(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b) \geq 1$ for $n = 1, .., M$. SVMs can only deal with two-class classification problems ($y_i \in \{-1, 1\}$). Subsequently, it follows this optimisation problem:

$$\begin{aligned} \underset{\omega,b}{\text{minimise}} \quad & \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 \\ \text{subject to} \quad & y_i(\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle + b) \geq 1 \text{ for } i = 1, .., M. \end{aligned} \tag{2.4.2}$$

Maximising the margin (2.4.3) is equivalent to minimising the length of the hyperplane direction vector (see section 2.4.2 for details). Therefore, the hyperplane found by the above problem is equivalent to the hyperplane found by maximising the margin:

$$\underset{\omega,b,\rho}{\text{maximise}} \quad \rho \tag{2.4.3}$$

$$\text{subject to} \quad \begin{cases} y_i(\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle + b) \geq \rho & \text{for } i = 1, .., M \\ \|\boldsymbol{\omega}\|_2^2 = 1 \end{cases}$$

By requiring the scaling of $\|\boldsymbol{\omega}\|_2^2 = 1$, a canonical form of the hyperplane $(\boldsymbol{\omega}, b)$ is defined.

If the cases are not separable (due to noise), a soft margin and so-called slack variables ξ_i, ξ_i^* are introduced. The slack variables ξ_i and ξ_i^* model a possible margin violation of case $\boldsymbol{x}_i$. The use of slack variables is penalised

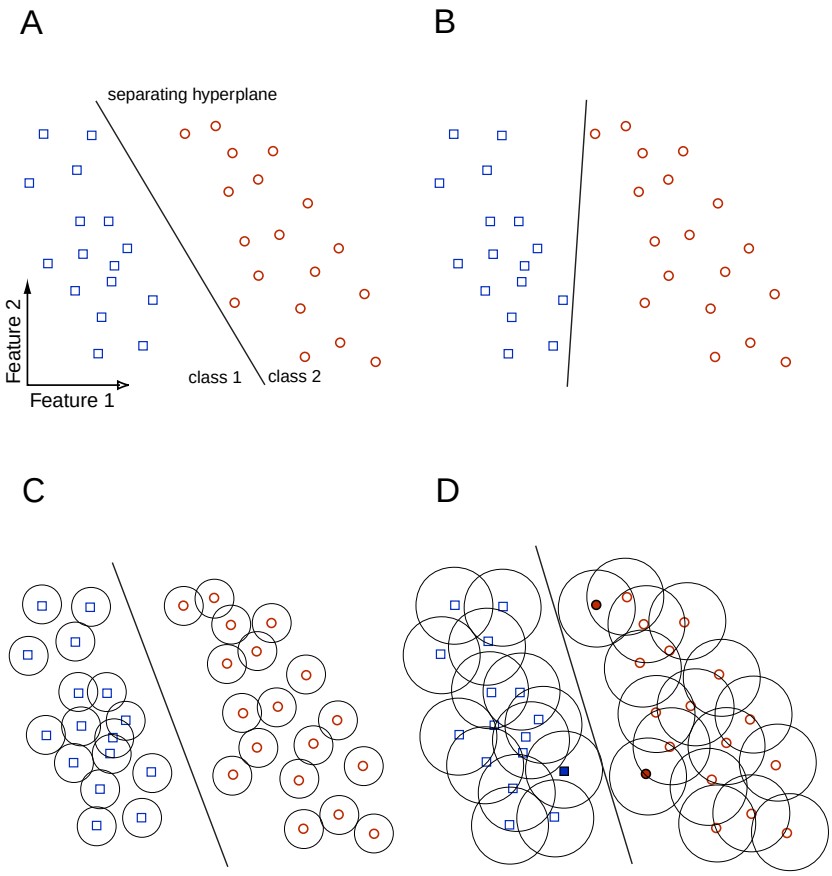


Figure 2.4: Different hyperplanes separate the data set (A and B). Maximising the margin around each data point determines the position of the hyperplane (C and D). Finally, three support vectors (bold) emerge.

by an additional regularisation parameter C. This parameter C controls the influence of outliers (cases on the wrong side of the decision hyperplane).

The new optimisation problem is given by:

$$\underset{\omega,b,\boldsymbol{\xi}}{\text{minimise}} \quad \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{M}\xi_i, \tag{2.4.4}$$

$$\text{subject to} \quad \begin{cases} y_i(\langle \boldsymbol{\omega}, \boldsymbol{x_i}\rangle + b) \geq 1 - \xi_i & \text{for } i = 1,..,M \\ \xi_i \geq 0 & \text{for } i = 1,..,M. \end{cases}$$

The second key concept of SVMs is a nonlinear mapping of cases from the input space $\mathbb{X}$ to a higher-dimensional space (called feature space) $\mathbb{F}$. It is not required to do the mapping explicitly in the higher-dimensional feature space because the (mapped) feature values only occur within scalar products in the algorithm. The kernel $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \Psi(\boldsymbol{x}_1), \Psi(\boldsymbol{x}_2)\rangle$ calculates these scalar products in the feature space. This is called the "kernel-trick". See also Fig. 2.5.

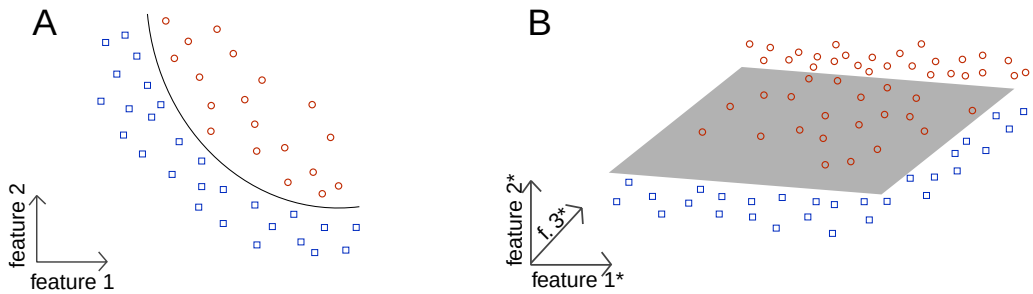


Figure 2.5: A projection from the two-dimensional input space (A) to the three-dimensional feature space (B) enables a separation by a linear hyperplane. Feature 1*, feature 2* and feature 3* (abbr. f.3*) are (non-)linear combinations of feature 1 and feature 2.

The resulting classifier

$$f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{M}\alpha_i y_i K(\mathbf{s_i}, \mathbf{x}) + b) \tag{2.4.5}$$

is linear in feature space, but not necessarily in input space. Kernelised versions of maximum margin discriminants are called support vector machines. The support vectors $\mathbf{s_i}$ are those training examples lying closest to the hyperplane and y_i are their class labels. The corresponding coefficients α_i and the

bias constant b reflect the solution of the quadratic programming problem and define the position of the separating hyperplane.

Different kernels can be applied to support vector machines. The linear kernel

$$K(\mathbf{s}_i, \mathbf{x}) = \mathbf{s_i^t x} = \sum_{j=1}^{N} (s_{ij} x_j) \tag{2.4.6}$$

and the polynomial kernel

$$K(\mathbf{s}_i, \mathbf{x}) = (c + \gamma \mathbf{s_i^t x})^d = (c + \gamma \sum_{j=1}^{N} s_{ij} x_j)^d \tag{2.4.7}$$

are examples where N is the dimensionality of the input space and c, d, γ are constants.

See section 2.4.4 for a detailed discussion regarding kernel functions.

Support vector machines have shown very good generalisation performance ([MLH03]). SVMs perform well on small training sets and always find the global optimum of the hyperplane in the training process. Thus, they are a good alternative to artificial neural networks.

2.4.2 Derivation of the support vector machine algorithm

In the following section, two ways to derive support vector machines are outlined. Vapnik motivated support vector machines by the large margin [Vap95]. However, SVMs can also be derived using regularisation theory instead of the large margin [Gir98, EPP00, SSM98].

Motivation of SVMs by a large margin

A support vector machine classifier motivated by the maximum margin would be the solution of the following problem:

$$\begin{aligned} &\underset{\omega, b, \rho}{\text{maximise}} && \rho \\ &\text{subject to} && \begin{cases} y_i(\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle + b) \geq \rho & \text{for } i = 1, .., M \\ \|\boldsymbol{\omega}\|_2^2 = 1. \end{cases} \end{aligned} \tag{2.4.8}$$

By requiring the scaling of $\|\boldsymbol{\omega}\|_2^2 = 1$, a canonical form of the hyperplane $(\boldsymbol{\omega}, b)$ is defined.

However, this is a difficult optimisation problem (due to nonlinear constraints). An equivalent optimisation problem is given by [Vap95]

$$\begin{aligned} \underset{\omega,b}{\text{minimise}} \quad & \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 \\ \text{subject to} \quad & y_i(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b) \geq 1 \text{ for } i = 1, .., M. \end{aligned} \tag{2.4.9}$$

The solution $\boldsymbol{\omega}_*$ of problem (2.4.8) and the solution $\boldsymbol{\omega}_{**}$ of problem (2.4.9) relate to each other as follows:

$$\boldsymbol{\omega}_* = \frac{\boldsymbol{\omega}_{**}}{\|\boldsymbol{\omega}_{**}\|}. \tag{2.4.10}$$

To construct a SVM that is nonlinear in the input space, the linear hyperplane is relocated in the feature space:

$$\begin{aligned} \underset{\omega,b}{\text{minimise}} \quad & \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 \\ \text{subject to} \quad & y_i(\langle \boldsymbol{\omega}, \Psi(\boldsymbol{x_i}) \rangle + b) \geq 1 \text{ for } i = 1, .., M. \end{aligned} \tag{2.4.11}$$

Note, that the parameter b and $\boldsymbol{\omega}$ are situated in the feature space. Therefore, the easier dual Wolfe problem L_D (2.4.13) is solved. This is obtained by analysing the primal problem according to Lagrange and introducing Lagrangian multipliers α_i:

$$\begin{aligned} \underset{\omega,b}{\text{minimise}} \quad & L_P = \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 - \sum_{i=1}^{M} \alpha_i (y_i(\langle \boldsymbol{\omega}, \Psi(\boldsymbol{x_i}) \rangle + b) - 1) \\ \text{subject to} \quad & \alpha_i \geq 0 \text{ for } i = 1, .., M. \end{aligned} \tag{2.4.12}$$

At the optimal (saddle) point, the first derivatives of L_P with respect to b and $\boldsymbol{\omega}$ are zero which leads to

$$\sum_{i=0}^{M} \alpha_i y_i = 0 \text{ and } \boldsymbol{\omega} = \sum_{i=0}^{M} \alpha_i y_i \Psi(\boldsymbol{x_i})$$

and can substituted back into L_P (2.4.12).

Finally, using $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Psi(\boldsymbol{x_i}), \Psi(\boldsymbol{x_j}) \rangle$, the dual problem L_D is given by

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximise}} \quad & L_D = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{M} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{subject to} \quad & \begin{cases} \alpha_i \geq 0 & \text{for } i = 1, .., M \\ \sum_{i=0}^{M} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.4.13}$$

The solution of this problem can be found by using a quadratic program solver like quadprog in Matlab. See also section 2.4.6.

In the case of noise, the original SVM problem is given by

$$\begin{aligned} \underset{\omega, b}{\text{minimise}} \quad & \frac{1}{2} \|\omega\|_2^2 + C \sum_{i=1}^{M} \xi_i, \\ \text{subject to} \quad & \begin{cases} y_i(\langle \boldsymbol{\omega}, \Psi(\boldsymbol{x_i}) \rangle + b) \geq 1 - \xi_i & \text{for } i = 1, .., M. \\ \xi_i \geq 0 & \text{for } i = 1, .., M. \end{cases} \end{aligned} \tag{2.4.14}$$

with the Wolfe dual

$$\begin{aligned} \text{maximise} \quad & \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{M} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{subject to} \quad & \begin{cases} 0 \leq \alpha_i \leq C & \text{for } i = 1, .., M \\ \sum_{i=0}^{M} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.4.15}$$

This problem is actually solved by an SVM. Note, that the only difference between the hard and soft margin optimisation problem of the SVM is an additional upper bound of the Lagrange multipliers α_i. Furthermore, all cases with a Lagrange multiplier $\alpha_i \neq 0$ are called support vectors and all cases with $\alpha_i = C$ are outliers (lying on the wrong side of the decision hyperplane).

Motivation of SVMs using regularisation theory

In terms of regularisation theory, the minimisation problem is given by:

$$\underset{f\in\mathfrak{F}}{\text{minimise}}\ C\sum_{i=1}^{M}\mathcal{L}(y_i, f(\boldsymbol{x_i})) + \frac{1}{2}\Phi(f(\boldsymbol{x})). \tag{2.4.16}$$

The function space $\mathfrak{F}$ is here a Reproducing Kernel Hilbert Space (RKHS) and the decision function $f(\boldsymbol{x})$ has an unique expansion

$$f(\boldsymbol{x}) = \sum_{j=1}^{\infty} c_j\phi_j(\boldsymbol{x}) \tag{2.4.17}$$

where ϕ_j are linearly independent functions and c_j are coefficients. It is assumed that one of the basis functions ϕ_j is constant, otherwise an additional parameter b has to be added.

The scalar product in the RKHS is given by

$$\langle f_1, f_2\rangle_{\mathfrak{H}} = \left\langle \sum_{j=1}^{\infty} c_{1j}\phi_j, \sum_{j=1}^{\infty} c_{2j}\phi_j \right\rangle_{\mathfrak{H}} = \sum_{j=1}^{\infty}\frac{c_{1j}c_{2j}}{\lambda_j} \tag{2.4.18}$$

and the norm has the form

$$\|f\|_{\mathfrak{H}}^2 = \langle f, f\rangle_{\mathfrak{H}} = \sum_{j=1}^{\infty}\frac{c_j^2}{\lambda_j} \tag{2.4.19}$$

where $\{\lambda_n\}_{n=1}^{\infty}$ is a decreasing positive sequence.

The regularisation operator $\Phi(f(\boldsymbol{x}))$ is the norm:

$$\Phi(f(\boldsymbol{x})) = \|f\|_{\mathfrak{H}}^2 = \sum_{j=1}^{\infty}\frac{c_j^2}{\lambda_j}. \tag{2.4.20}$$

By choosing $\Phi(f(\boldsymbol{x}))$ as $\|f\|_{\mathfrak{H}}^2$, the solution of the optimisation problem (2.4.16) has always the form [Gir98]

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} a_i K(\boldsymbol{x}, \boldsymbol{s}_i) + b. \tag{2.4.21}$$

The kernel function

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\boldsymbol{x_1}) \phi_j^*(\boldsymbol{x_2}) \tag{2.4.22}$$

has the reproducing property

$$f(\boldsymbol{x_1}) = \langle f(\boldsymbol{x_2}), K(\boldsymbol{x_2}, \boldsymbol{x_1}) \rangle_{\mathfrak{H}}. \tag{2.4.23}$$

and can be used to construct an RKHS.

The space $\{(\phi_j(\boldsymbol{x}))_{j=1}^{\infty}, \boldsymbol{x} \in \mathbb{X}\}$ is called the feature space $\mathbb{F}$ induced by the kernel K.

An example for a kernel function would be a radial basis kernel where the ϕ_j are Fourier components [EPP00]:

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \sum_{j=1}^{\infty} \lambda_j e^{\mathrm{i}2\pi j \boldsymbol{x_1}} e^{-\mathrm{i}2\pi j \boldsymbol{x_2}}). \tag{2.4.24}$$

The use of the ϵ-insensitive loss function in the optimisation problem leads to an approximation scheme of an SVM. Note, that we consider here a regression problem. However, any SVM classification problem can be solved as an SVM regression problem [EPP00].

The optimisation problem

$$\underset{f \in \mathfrak{F}}{\text{minimise}} \; C \sum_{i=1}^{M} \mathcal{L}_\epsilon(y_i, f(\boldsymbol{x_i})) + \frac{1}{2} \|f\|_{\mathfrak{H}}^2 \tag{2.4.25}$$

has to be replaced by an equivalent one to deal with the $\mathcal{L}_\epsilon$ loss function [Gir98]

$$\underset{f\in\mathfrak{F},\xi_i,\xi_i^*}{\text{minimise}} \quad C\sum_{i=1}^{M}(\xi_i+\xi_i^*)+\frac{1}{2}\sum_{j=1}^{\infty}\frac{c_j^2}{\lambda_j} \tag{2.4.26}$$

$$\text{subject to} \quad \begin{cases} f(\boldsymbol{x}_i)-y_i\le\epsilon+\xi_i & \text{for } i=1,..,M \\ y_i-f(\boldsymbol{x}_i)\le\epsilon+\xi_i^* & \text{for } i=1,..,M \\ \xi_i\ge 0 & \text{for } i=1,..,M \\ \xi_i^*\ge 0 & \text{for } i=1,..,M. \end{cases}$$

The Lagrangian corresponding to this problem with Lagrangian multipliers α_i, α_i^* and β_i is:

$$\begin{aligned}\underset{f\in\mathfrak{F},\xi_i,\xi_i^*,\boldsymbol{\alpha},\boldsymbol{\alpha}^*,\boldsymbol{\xi},\boldsymbol{\xi}^*}{\text{minimise}} \quad & C\sum_{i=1}^{M}(\xi_i+\xi_i^*)+\frac{1}{2}\sum_{j=1}^{\infty}\frac{c_j^2}{\lambda_j}+\sum_{i=1}^{M}\alpha_i^*(y_i-f(\boldsymbol{x}_i)-\epsilon-\xi_i^*) \\ & +\sum_{i=1}^{M}\alpha_i(f(\boldsymbol{x}_i)-y_i-\epsilon-\xi_i)-\sum_{i=1}^{M}(\beta_i\xi_i+\beta_i^*\xi_i^*)\end{aligned} \tag{2.4.27}$$

$$\text{subject to} \quad \alpha_i,\alpha_i^*,\beta_i,\beta_i^*\ge 0 \text{ for } i=1,..,M.$$

If the derivatives are zero, it follows

$$c_j=\lambda_j\sum_{i=1}^{M}(\alpha_i^*-\alpha_i)\phi_j(\boldsymbol{x}_i) \tag{2.4.28}$$

$$\text{and} \quad \sum_{i=1}^{M}(\alpha_i^*-\alpha_i)=0. \tag{2.4.29}$$

Substituting back into the definition of the decision function 2.4.17, one obtains:

$$f(\boldsymbol{x})=\sum_{i=1}^{M}(\alpha_i^*-\alpha_i)K(\boldsymbol{x},\boldsymbol{s}_i)+b. \tag{2.4.30}$$

The α_i and α_i^* have to be determined from the following quadratic programming problem [Gir98]:

$$\begin{aligned} \underset{\alpha_i,\alpha_i^*}{\text{minimise}} \quad & \epsilon \sum_{i=1}^{M}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{M} y_i(\alpha_i^* - \alpha_i) \\ & + \frac{1}{2}\sum_{i,j=1}^{M}(\alpha_j^* - \alpha_j)(\alpha_i^* - \alpha_i)K(x_i, x_j) \\ \text{subject to} \quad & \begin{cases} 0 \leq \alpha_i, \alpha_i^* \leq C & \text{for } i = 1, .., M \\ \sum_{i=1}^{M}(\alpha_i^* - \alpha_i) = 0 & \text{for } i = 1, .., M \end{cases} \end{aligned} \tag{2.4.31}$$

Summarising, SVMs for regression and classification can be derived within the regularisation framework by using a special loss function.

2.4.3 Support vector machine survival regression based on censored observations

Basic idea

Let us consider a special case of regression problems. The aim here is to estimate the time until an event occurs (e.g., healing, relapse, heart attack or death of a patient) based on a feature vector $\boldsymbol{x}_i$. Survival times are often only partially observed. This is the case for patients still alive at the end of a study or lost to follow-up. Incomplete observations are called censored observations.

This information is stored in a vector *cens* where each element is 0 for a censored and 1 for a not censored observation. Imagine a breast case study over 15 years (see Fig. 2.6). If a person leaves the study (for whatever reason) or is still alive at the end of the study, we don't know the true survival time of this person. However, we know that the true survival time is not lower than the time observed.

My idea to estimate survival times with a support vector machine is based on a special loss function $\mathcal{L}_{surv}$ (Fig. 2.7). This loss function is defined as

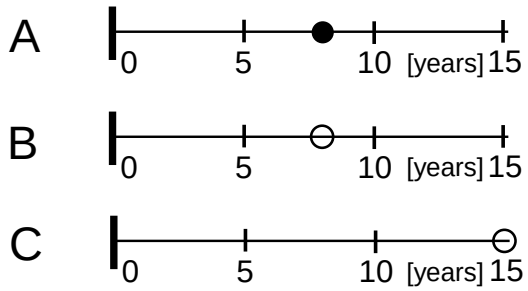


Figure 2.6: Censored observations in a study. A) Death of a patient was observed in year 8 (not censored). B) Patient left study in year 8 (censored). C) Patient was still alive at the end of the study (censored).

$$\mathcal{L}_{surv}(y_i, f(\boldsymbol{x_i})) = \begin{cases} 0 & \text{for } |y_i - f(\boldsymbol{x}_i)| \leq \epsilon \\ |y_i - f(\boldsymbol{x}_i)| - \epsilon & \text{for } y_i - f(\boldsymbol{x}_i) > \epsilon \text{ and } cens_i = 1 \\ |y_i - f(\boldsymbol{x}_i)| - \epsilon & \text{for } f(\boldsymbol{x}_i) - y_i > \epsilon \text{ and } cens_i = 1 \\ |y_i - f(\boldsymbol{x}_i)| - \epsilon & \text{for } y_i - f(\boldsymbol{x}_i) > \epsilon \text{ and } cens_i = 0 \\ 0 & \text{for } f(\boldsymbol{x}_i) - y_i > \epsilon \text{ and } cens_i = 0. \end{cases} \tag{2.4.32}$$

Important features of this loss function are:

- For non-censored observations, this loss function is equivalent to the ϵ-insensitive loss $\mathcal{L}_\epsilon$.
- Margin violations of censored cases are not penalised if the estimated survival time is greater than the observed survival time.

Note that here we are only dealing with right censored data, where we know the starting but not always the end point. However, in principle, all considerations also hold for left censored data.

The proposed regression method is only valid if the reasons for observing a censored observation are unrelated to the features $\boldsymbol{x}$ of a case.

Derivation of the optimisation problem

The optimisation problem given the $\mathcal{L}_{surv}$ loss function is:

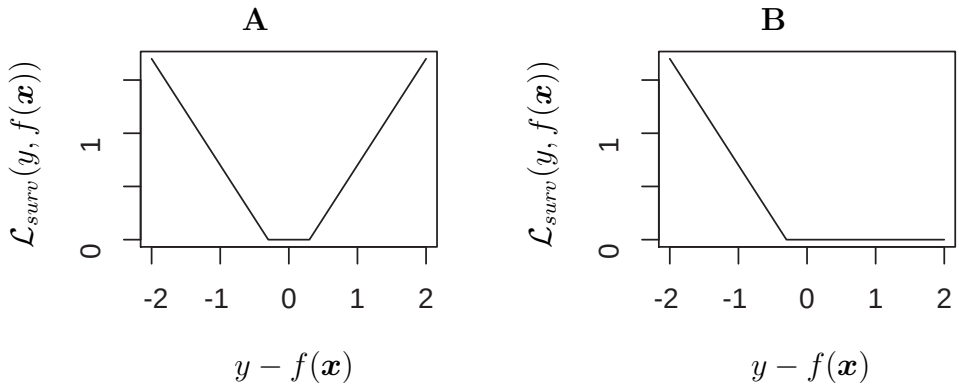


Figure 2.7: Loss function $\mathcal{L}_{surv}$ for $\epsilon = 0.3$. A) not censored case B) censored case.

$$\underset{f \in \mathfrak{F}}{\text{minimise}} \; C \sum_{i=1}^{M} \mathcal{L}_{surv}(y_i, f(\boldsymbol{x_i})) + \frac{1}{2} \|\boldsymbol{\omega}\|_2^2 \tag{2.4.33}$$

In analogy to the support vector regression, it has to be replaced by an equivalent one to deal with the $\mathcal{L}_{surv}$ loss function (compare (2.4.25)):

$$\underset{f \in \mathfrak{F}, \xi_i, \xi_i^*}{\text{minimise}} \quad C \sum_{i=1}^{M} (\xi_i (1 - cens_i) + \xi_i^*) + \frac{1}{2} \|\boldsymbol{\omega}\|^2 \tag{2.4.34}$$

$$\text{subject to} \quad \begin{cases} f(\boldsymbol{x}_i) - y_i \leq \epsilon + \xi_i & \text{for } i = 1, .., M \\ y_i - f(\boldsymbol{x}_i) \leq \epsilon + \xi_i^* & \text{for } i = 1, .., M \\ \xi_i \geq 0 & \text{for } i = 1, .., M \\ \xi_i^* \geq 0 & \text{for } i = 1, .., M \end{cases}$$

The overall penalty factor C is now replaced by two penalty vectors $\boldsymbol{C}$ and $\boldsymbol{C^*}$ to penalise case-specific margin violations.

C_i^* is set to C for all cases (as long as we consider right censored data). C_i is zero for censored cases and C for not censored cases.

$$\underset{f\in\mathfrak{F},\xi_i,\xi_i^*}{\text{minimise}} \quad \sum_{i=1}^{M}(\xi_i C_i + \xi_i^* C_i^*) + \frac{1}{2}\left\|\boldsymbol{\omega}\right\|^2 \tag{2.4.35}$$

$$\text{subject to} \quad \begin{cases} \langle \boldsymbol{\omega}, \Psi(\boldsymbol{x_i})\rangle + b - y_i \leq \epsilon + \xi_i & \text{for } i = 1,..,M \\ y_i - \langle \boldsymbol{\omega}, \Psi(\boldsymbol{x_i})\rangle - b \leq \epsilon + \xi_i^* & \text{for } i = 1,..,M \\ \xi_i \geq 0 & \text{for } i = 1,..,M \\ \xi_i^* \geq 0 & \text{for } i = 1,..,M \end{cases}$$

Note, that the parameter b and $\boldsymbol{\omega}$ are situated in the feature space. Therefore, the easier dual Wolfe problem is solved. This is obtained by analysing the primal problem according to Lagrange and introducing Lagrangian multipliers $\alpha_i, \alpha_i^* \eta_i, \eta_i^*$ for each inequality constraint:

$$\begin{aligned} \underset{\boldsymbol{\omega},b}{\text{minimise}} \quad L_P = \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 &+ \sum_{i=1}^{M} C_i\xi_i + C_i^*\xi_i^* - \sum_{i=1}^{M}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ &- \sum_{i=1}^{M}\alpha_i(\epsilon + \xi_i + y_i - \langle \boldsymbol{\omega}, \Psi(\boldsymbol{x_i})\rangle - b) \\ &- \sum_{i=1}^{M}\alpha_i^*(\epsilon + \xi_i^* - y_i + \langle \boldsymbol{\omega}, \Psi(\boldsymbol{x_i})\rangle + b) \end{aligned} \tag{2.4.36}$$

$$\text{subject to} \qquad \alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0 \text{ for } i = 1,..,M.$$

At the optimal (saddle) point, the first derivatives of L_P with respect to b and $\boldsymbol{\omega}$ are zero which leads to

$$\frac{\partial L_P}{\partial \boldsymbol{\omega}} = \boldsymbol{\omega} + \sum_{i=1}^{M}(\Psi(\boldsymbol{x_i})\alpha_i) - \sum_{i=1}^{M}(\Psi(\boldsymbol{x_i})\alpha_i^*) = 0 \tag{2.4.37}$$

$$\frac{\partial L_p}{\partial b} = \sum_{i=1}^{M}(\alpha_i) - \sum_{i=1}^{M}(\alpha_i^*) = 0 \tag{2.4.38}$$

$$\frac{\partial L_p}{\partial \xi_i} = C_i - \alpha_i - \eta_i = 0 \tag{2.4.39}$$

$$\frac{\partial L_P}{\partial \xi *_i} = C_i^* - \alpha_i^* - \eta_i^* = 0 \tag{2.4.40}$$

and can be substituted back into L_P (2.4.36).

Finally, using $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Psi(\boldsymbol{x_i}), \Psi(\boldsymbol{x_j}) \rangle$, the dual problem is given by

$$\begin{aligned} \underset{\alpha_i, \alpha_i^*}{\text{maximise}} \quad & -\epsilon \sum_{i=1}^{M} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{M} y_i (\alpha_i^* - \alpha_i) \\ & -\frac{1}{2} \sum_{i,j=1}^{M} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(x_i, x_j) \\ \text{subject to} \quad & \begin{cases} 0 \leq \alpha_i \leq C_i & \text{for } i = 1, .., M \\ 0 \leq \alpha_i^* \leq C_i^* & \text{for } i = 1, .., M \\ \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) = 0 & \text{for } i = 1, .., M \end{cases} \end{aligned} \tag{2.4.41}$$

which can be solved by a quadratic program solver.

Note, that the only difference from a normal SVM regression problem is that individual constraints limit the α_i for censored data points.

Discussion

An alternative regression method for censored observations is the exponential regression model [HL99]

$$T = e^{\beta_0 + \beta_1 x}. \tag{2.4.42}$$

The coefficients β_0 and β_1 are fitted by maximum likelihood estimation (MLE).

Therefore, the log-likelihood function is maximised:

$$L(\beta_0, \beta_1) = \sum_{i=1}^{M} cens_i \left[y_i - (\beta_0 + \beta_1 x_i) \right] - e^{[y_i - (\beta_0 + \beta_1 x_i)]}. \tag{2.4.43}$$

In order to obtain the MLE, the derivatives of $L(\beta_0, \beta_1)$ are set to zero.

$$\frac{\partial L(\beta_0, \beta_1)}{\partial \beta_0} = \sum_{i=1}^{M} \left(cens_i - e^{[y_i - (\beta_0 + \beta_1 x_i)]} \right) = 0 \tag{2.4.44}$$

$$\frac{\partial L(\beta_0, \beta_1)}{\partial \beta_1} = \sum_{i=1}^{M} x_i \left(cens_i - e^{[y_i - (\beta_0 + \beta_1 x_i)]} \right) = 0 \tag{2.4.45}$$

These equations have to be solved using an iterative numerical method.

The only SVM survival approach I am aware of, was presented in a talk by Härdle and Moro [HM04]. The idea is to divide the observed survival times into T periods. For each period, a classifier for all patients alive at period t is trained, this distinguishes patients that will die in period $t+1$ from patients that survive period $t+1$. Finally, the regression problem is reduced to T binary classification problems.

2.4.4 Kernel

Kernels intuitively measure the similarity between two cases. Kernels designed for a special application are an effective alternative to an explicit feature extraction.

A kernel function calculates the scalar product of two cases that are mapped from an input space $\mathbb{X}$ to a feature space $\mathbb{F}$:

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \langle \Psi(\boldsymbol{x_1}), \Psi(\boldsymbol{x_2}) \rangle \tag{2.4.46}$$

The mapping function $\Psi(\boldsymbol{x})$ is implicitly defined by the kernel function. However, for some finite-dimensional kernels the mapping function $\Psi(\boldsymbol{x})$ is quite intuitive. For the polynomial kernel and $d = 2$ the mapping in the two-dimensional case results in $\Psi(x_1, x_2) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2]^t$.

A necessary and sufficient condition for a symmetric function to be a kernel is its positive definiteness (Mercer's condition):

$$\sum_{i,j=1}^{M} \lambda_i \lambda_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \geq 0 \tag{2.4.47}$$

for any set of real numbers $\lambda_1, .., \lambda_M$ and any set of samples.

Moreover, any $K(\boldsymbol{x}_1, \boldsymbol{x}_2)$ can be expanded in its eigenfunctions

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \sum_{j=1}^{J} \lambda_j \phi_j(\boldsymbol{x_1}) \phi_j^*(\boldsymbol{x_2}) \tag{2.4.48}$$

with a possible mapping function [MMR+01]:

$$\Psi(\boldsymbol{x}) = (\sqrt{\lambda_1}\phi_1(\boldsymbol{x}), \sqrt{\lambda_2}\phi_2(\boldsymbol{x}), ..., \sqrt{\lambda_J}\phi_J(\boldsymbol{x})). \tag{2.4.49}$$

Examples of kernel functions are given in table 2.2 [Vap95, SS98, RSS05].

An example of a string kernel is given by the WD kernel of length d:

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k+1} Q(u_{kl}(\boldsymbol{x_1}) == u_{kl}(\boldsymbol{x_2})), \tag{2.4.50}$$

where $u_{kl}(\boldsymbol{x_i})$ is the sequence of length k starting at position l, $Q(true) = 1$, $Q(false) = 0$, $\beta_k = \frac{2(d-k+1)}{d(d+1)}$ and L the length of the string.

Note that the exponential, Gaussian, thin plate splines and both multiquadric kernels are translation invariant and isotropic. Translation invariant kernels depend only on the distance vector between two samples

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = K_S(\boldsymbol{x_1} - \boldsymbol{x_2}), \tag{2.4.51}$$

whereas isotropic kernel functions use only the norm of the distance vector:

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = K_I(\|\boldsymbol{x_1} - \boldsymbol{x_2}\|_2^2). \tag{2.4.52}$$

The question now arises, which kernel function should be chosen. For translation invariant kernels, the kernel should match prior knowledge about the frequency distribution (power spectrum, calculated as square of the Fourier transform of f) of the decision function f (see [Smo98]). Generally and without previous knowledge of the decision function, Gaussian RBF kernels are a good choice.

Name	Definition
Gaussian radial-basis-function kernel	$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = e^{-\gamma\|\boldsymbol{x_1}-\boldsymbol{x_2}\|_2^2}$
Exponential kernel	$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = e^{-\gamma\|\boldsymbol{x_1}-\boldsymbol{x_2}\|_2}$
Thin plate splines kernel	$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \|\boldsymbol{x_1} - \boldsymbol{x_2}\|_2^2 \ln \|\boldsymbol{x_1} - \boldsymbol{x_2}\|_2$
Sigmoidal kernel, only for some values of γ defined	$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \tanh(\gamma \boldsymbol{x_1} \boldsymbol{x_2} + \delta)$
Polynomial kernel of degree d	$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = (c + \gamma \boldsymbol{x_1} \boldsymbol{x_2})^d$
Multiquadric kernel	$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \sqrt{\|\boldsymbol{x_1} - \boldsymbol{x_2}\|_2^2 + c^2}$
Inverse multiquadric kernel	$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \frac{1}{\sqrt{\|\boldsymbol{x_1} - \boldsymbol{x_2}\|^2 + c^2}}$

Table 2.2: Kernel functions

2.4.5 Universal approximators

Support vector machine with various kernels (e.g., sigmoidal and RBF) are universal approximators ([HG03]). This means that they can approximate any reasonable decision function up to a desired accuracy. The same has been shown for feedforward neural networks (e.g., [HSW89]).

A function class $\mathfrak{F}$ possesses universal approximation capabilities if for any compact set $\mathbb{C} \subset \mathbb{R}^N$, any continuous function $g(\boldsymbol{x}) : \mathbb{C} \to \mathbb{R}^N$ and any $\epsilon > 0$ some $f \in \mathfrak{F}$ can be found such that

$$|f(\boldsymbol{x}) - g(\boldsymbol{x})| \leq \epsilon \text{ for all } \boldsymbol{x} \in \mathbb{C}. \tag{2.4.53}$$

If the function class $\mathfrak{F}$ is used for classification only it possesses universal approximation capabilities with respect to a probability measure P if for any measurable function $g(\boldsymbol{x}) : \mathbb{R}^N \to \{-1, +1\}$ and any $\delta > 0$ some $f \in \mathfrak{F}$ can be found such that

$$P(\boldsymbol{x}|f(\boldsymbol{x}) \neq g(\boldsymbol{x})) < \delta. \tag{2.4.54}$$

The proof for sigmoidal kernels is based on the universal approximation capability of feedforward neural networks [HSW89], whereas the proof for RBF kernels refers to radial basis networks and their universal approximation capability [PS91]. Polynomial kernels with unlimited degree d are also universal approximators. However, a data set of size $|\mathbb{S}|$ can be approximated by a polynomial kernel of degree $d = |\mathbb{S}| - 1$ [HG03].

2.4.6 Implementation and training time complexity

Two software packages implementing SVMs have been used in this thesis. SVM^{light} from Thorsten Joachims, University of Dortmund, Germany [Joa99] and libsvm from Chih-Jen Lin, National Taiwan University, Taiwan [CL01].

A quadratic programming (QP) solver for solving the SVM problem (2.4.15) is limited to approximately thousand samples. For larger problems, a "chunking", "decomposition" or "sequential minimal optimisation" technique is used [CST00].

Chunking is based on the fact that only support vectors are important for the solution of the QP (quadratic programming) problem. Iteratively, the

support vectors from the last step and cases violating the optimality conditions (KKT conditions) are included. Decomposition resolves the large QP problem into very small QP problems. In the case of sequential minimal optimisation a sub problem of only two cases is analytically solved at each iteration [Pla99a].

The empirical scaling of state-of-the-art implementations, like SVM^{light}, in terms of time complexity and the number of samples is between linear ($O(M)$) and cubic ($O(M^3)$) and most often quadratic ($O(M^2)$) [Joa99, Pla99a].

More recently, Hush presented an SVM training algorithm that is bounded by a quadratic complexity class ($O(M^2)$) [HSS05].

Tresp and Schwaighofer discusses several approaches to narrow down the computational complexity of the SVM learning algorithm by using a subset of the M cases in the training set as M_{base} base kernels [TS01]. The complexity of the learning algorithm is subsequently reduced to $O(M \times M_{base}^2)$.

Collobert and Bengio suggested a weighted ensemble of SVMs, where each SVM is trained on a subset of the available data [CBB02]. A modification of this approach, using equal weights of all classifiers, has been successfully applied in a cooperation of A. Vinayagam, R. König, J. Moormann, KH. Glatting, R. Eils, S. Suhai and myself [VKM+04].

2.4.7 Comparison of boosting and support vector machines

General idea of boosting

The underlying idea of boosting is to improve the classification accuracy of an ensemble of L weak classifiers $f^{(j)}$ by combining them such that they complement each other. Briefly, each classifier emphasises on cases in the learning process that were previously wrongly classified. Finally, the weighted majority vote of the weak classifiers determines the classification result:

$$f_{ensemble}(\boldsymbol{x}) = \text{sign}(\sum_{j=1}^{L} \alpha_j f^{(j)}(\boldsymbol{x})). \tag{2.4.55}$$

An example of a weak classifier would be a decision tree (see section 2.5.1) or a decision stump (tree with one decision node).

Adaboost

An example of a boosting algorithm is adaboost [FS96]. Adaboost assigns a weight to each sample. Weights may be resolved, in case the weak learner does not support weighted samples, by resampling with a probability relative to the weight. At the beginning, all weights are equal. Later the weights are altered according to the classification accuracy of each classifier $f^{(j)}(\boldsymbol{x})$. Classifiers with $R_{emp} = 0$ or $R_{emp} \geq 0.5$ are discarded.

The weights of all correctly learned samples ($\forall_{\boldsymbol{x_i}} f^{(j)}(\boldsymbol{x_i}) == y_i$) are updated as follows in Adaboost (version Adaboost.M1) [WF00]:

$$weight_{i,j} := weight_{i,j} \frac{R_{emp}(f^{(j)})}{1 - R_{emp}(f^{(j)})}. \tag{2.4.56}$$

The weights of all wrongly learned samples ($\forall_{\boldsymbol{x}_i} f^{(j)}(\boldsymbol{x}_i) \neq y_i$) remain unchanged:

$$weight_{i,j} := weight_{i,j}. \tag{2.4.57}$$

Subsequently, the relative weight of correctly learned samples decreases while the relative weight of wrongly learned samples increases. Later, all weights are normalised and the next iteration starts. Boosting terminates if the gain of accuracy is smaller than a threshold or a maximal number of iterations is reached.

For testing, a weighted average over all classifiers is used (2.4.55). The α_j are calculated from the accuracy of each classifier.

Boosting versus SVM

Interestingly, the expected risk R decreases with the number of classifiers or boosting iterations. This can be explained by considering boosting as a large-margin-optimisation algorithm.

Let us assume that $\alpha_j = \frac{\omega_j}{||\boldsymbol{\omega}||_1}$ and that each base function $f^{(j)}$ can be chosen from a set $H = \{f^{(j)} | j = 1, .., J\}$ of J base functions $f^{(j)}$. The decision function becomes

$$f_{ensemble}(\boldsymbol{x}) = \text{sign}(\sum_{j=1}^{L} \frac{\omega_j}{||\boldsymbol{\omega}||_1} f^{(j)}(\boldsymbol{x})). \tag{2.4.58}$$

Boosting can now be reformulated as a quadratic programming problem [Rät01, MMR+01] that maximises the margin ρ:

$$\begin{aligned} &\underset{\rho,\boldsymbol{\omega}}{\text{maximise}} && \rho \\ &\text{subject to} && \begin{cases} y_i(\sum_{j=1}^{J} \omega_j f^{(j)}(\boldsymbol{x_i})) \geq \rho & \text{for } i = 1,..,M \\ ||\boldsymbol{\omega}||_1 = 1. \end{cases} \end{aligned} \tag{2.4.59}$$

The maximisation of the margin ρ of a linear SVM can be written as :

$$\begin{aligned} &\underset{\rho,\boldsymbol{\omega}}{\text{maximise}} && \rho \\ &\text{subject to} && \begin{cases} y_i(\sum_{j=1}^{J} \omega_j P_j(\boldsymbol{x_i})) \geq \rho & \text{for } i = 1,..,M \\ ||\boldsymbol{\omega}||_2 = 1. \end{cases} \end{aligned} \tag{2.4.60}$$

P_j is here an operator that projects $\boldsymbol{x}$ onto the j-th dimension ($P_j(\boldsymbol{x}) = x_j$).

Boosting maps each case explicitly to a feature space spanned by the set of base functions. However, the use of the L1 norm ($\|\boldsymbol{x}\|_1 = \sum |x_i|$) in boosting instead of the L2 norm ($\|\boldsymbol{x}\|_2 = \sqrt{\sum x_i^2}$) of an SVM leads to a sparse solution in $\boldsymbol{\omega}$.

Training of an SVM leads to a sparse expansion in $\boldsymbol{\omega}$, where only some cases (support vectors) contribute to the decision hyperplane.

Finally, boosting uses only the most important dimensions in feature space, whereas SVMs use only the most important cases (support vectors).

Note that boosting implements a hard margin. [Rät01] introduced therefore a regularised boosting algorithm with a soft margin.

2.5 Other classifiers

Two other classification algorithms, logic regression and decision trees, are considered in this thesis. Both were chosen accordingly to their explanatory potential.

2.5.1 Decision trees

Basic idea

A decision tree consists of internal decision nodes and terminal leaves (nodes without children). Each internal node implements a decision and each path to a children represents one outcome of the decision. To calculate the decision function of the whole tree, the decision tree is recursively traversed until a leaf node is met which assigns the class label (Fig. 2.8).

Decision trees are piece-wise axis-parallel classifiers. Training includes the recursive partitioning of the input space to find the best separation of cases to classes. An exhaustive search for the best decision tree (smallest tree with $R_{emp} = 0$) is often impossible. Therefore, heuristics driven by information-theoretic measures like information gain or information gain ratio are used.

The basic training algorithm of a decision tree is recursively invoked and starts with the root node. The feature of the first decision node is selected by considering all features and comparing their importance. This can be done by calculating the information gain of all features, and all possible splits and choosing the one with the highest score. A perfect feature would separate all cases into sets such that each set only belongs to one class. Alternatively, the best available split is the one with the lowest impurity of the classes.

For each outcome of the decision a new children node is inserted. The recursion is initiated for all cases that belong to this new node and all remaining features.

The algorithm stops if the nodes of the tree are pure, and comprise only cases from one class, or if the number of cases for a node is below a threshold.

Let us consider an example regarding the classification of apples, and peaches and the features size, colour, roughness of the peeling and thickness of the peeling. The best feature is the colour with three outcomes: green, red and yellow. The subset with the green colour consists only of apples whereas the subsets with a red or yellow colour are mixed. Both subsets are further

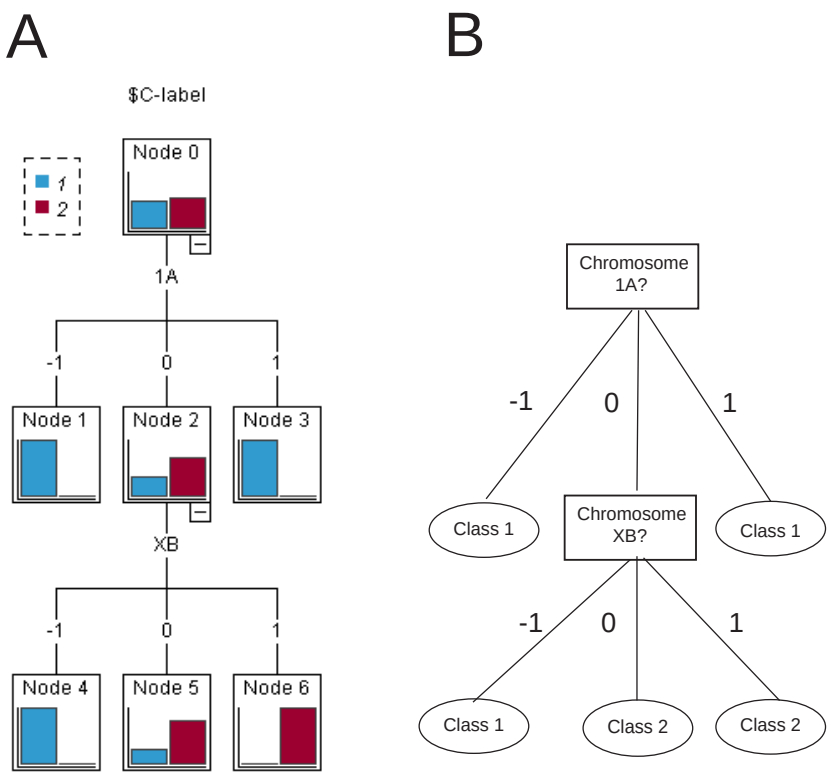


Figure 2.8: Decision tree for the separation of two tumour classes according to chromosomal (mouse) aberrations. -1 denotes a deletion, 0 a balanced situation and +1 an amplification. A) shows the separation of the two classes in each node of the tree. The histogram in node 0 visualises the overall distribution of tumour class 1 and tumour class 2. Node 1 and Node 3 are (almost) pure and belong to class 1. Node 6 on the other hand, contains cases from tumour class 2. The tree and the figure were generated using the algorithm C5.0 in an implementation of Clementine. B) sketches the decision tree and the class assignments. A case with a deletion of chromosomal band 1A and a deletion of chromosomal band XB would be assigned to tumour class 1 whereas a case with a balanced situation of chromosomal band 1A and an amplification of chromosomal band XB would be assigned to tumour class 2.

subdivided using the features size, roughness of the peeling and thickness of the peeling.

More formally, the algorithm starts by introducing a working set $\mathbb{T}$, that becomes initially the set $\mathbb{S}$ of all training cases, and a set $\mathbb{A}$ comprising all features. Depending on the class distribution of the cases in $\mathbb{T}$, one of the following actions is performed [WF00, RN95]:

- The number of cases in $\mathbb{T}$ is greater than or equal to a threshold and all cases in $\mathbb{T}$ belong to one class y :
  No further split is necessary. The decision tree for $\mathbb{T}$ is a leaf node associated with the class y. Terminate the algorithm for this branch.

- $\mathbb{T}$ contains no cases or the number of cases in $\mathbb{T}$ is below a threshold:
  Decision tree for $\mathbb{T}$ is a leaf (node without children) and the class association of this leaf has to be determined from prior knowledge. Alternatively, the most frequent class is chosen. Terminate the algorithm for this branch.

- The number of cases in $\mathbb{T}$ is greater than or equal to a threshold, the cases in $\mathbb{T}$ belong to at least two different classes and no features are left ($|\mathbb{A}| == 0$):
  Decision tree for $\mathbb{T}$ is a leaf (node without children) and the class association of this leaf has to be determined from background knowledge. Alternatively, the most frequent class is chosen. Terminate the algorithm for this branch.

- The number of cases in $\mathbb{T}$ is greater than or equal to a threshold, the cases in $\mathbb{T}$ belong to at least two different classes and at least one feature is left ($|\mathbb{A}| > 0$):
  Choose the best feature A_* from $\mathbb{A}$ that subdivides the set $\mathbb{T}$ into subsets such that different classes are separated by the subsets with the lowest impurity. A feature with l different outcomes separates $\mathbb{T}$ into l different subsets $\mathbb{T}_1^{(i)}$ to $\mathbb{T}_l^{(i)}$.
  A possible score for this choice is the information gain. The gain of feature A_i is calculated as

$$Gain(A_i) = I(\mathbb{T}) - \sum_{j=1}^{l} \frac{|\mathbb{T}_j^{(i)}|}{|\mathbb{T}|} I(\mathbb{T}_j^{(i)})$$

  with the information content of set $\mathbb{T}$ defined by

$$I(\mathbb{T}) = \sum_{k=1}^{K} \frac{-|\{(\boldsymbol{x}_i, y_i) \in \mathbb{T} | y_i == k\}|}{|\mathbb{T}|} \log_2 \left(\frac{|\{(\boldsymbol{x}_i, y_i) \in \mathbb{T} | y_i == k\}|}{|\mathbb{T}|} \right).$$

The information gain prefers features with a large number of different outcomes. The information gain ratio $GainRatio(A_i)$ corrects for this by taking into account the number of splits:

$$GainRatio(A_i) = \frac{Gain(A_i)}{(\sum_{j=1}^{l} \frac{|\mathbb{T}_j|}{|\mathbb{T}|} \log_2 \left(\frac{|\mathbb{T}_j|}{|\mathbb{T}|}\right))}.$$

If the feature is continuous, consider all possible split points. At most, $|\mathbb{T}| - 1$ different splits per continuous feature are possible.
Add an internal decision node with attribute A_*. Invoke the algorithm recursively for each subset $\mathbb{T}_1$ to $\mathbb{T}_l$ and the feature set $\mathbb{A}_{new} := \{\mathbb{A} - A_*\}$.
If the chosen feature was continuous, it should be considered again (with a different split point). The new feature set $\mathbb{A}_{new}$ in this case is the old feature set $\mathbb{A}$.

Decision trees use sometimes additional irrelevant features to find a tree classifying all samples. Therefore, post-pruning is applied to reduce the complexity of the classifier. Pruning replaces a subtree with a leaf node and prevents splitting by irrelevant features. The relevance of a split can be estimated by statistical tests.

From a logical point of view, decision trees are fully expressive within the class of propositional logic. That means, any Boolean expression can be represented by a decision tree [RN95].

Decision tree implementations

Examples of decision tree algorithms are ID3 and C4.5 ([Qui93]). The decision tree algorithm C5.0 arose from further development of C4.5 and is only available on a commercial basis. The time complexity of decision tree induction is $O(M \times \log(M))$ [WF00].

Recently, an ensemble method called random forests was developed [Bre01]. The idea behind random forests is to built an ensemble of decision trees with random feature selection and to combine their predictions by majority vote. Each tree is grown on a random sub sample of all cases that is drawn with replacement from the data set. Furthermore, for each tree and each node the best split is chosen among randomly selected features. Random forests show a good generalisation performance but a lack of interpretability.

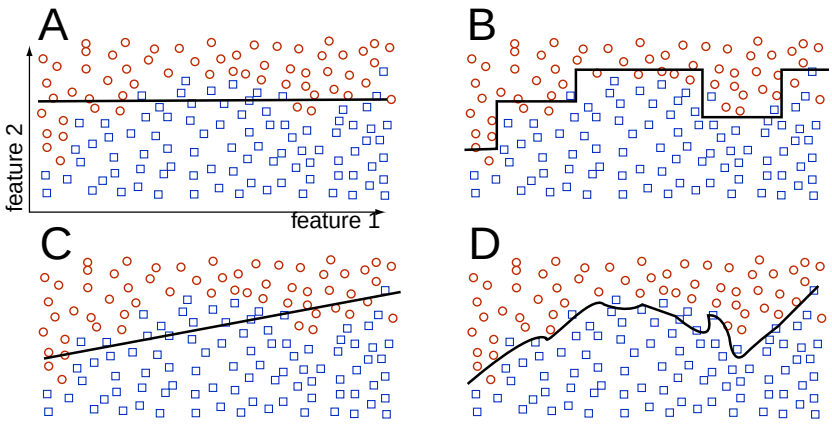


Figure 2.9: Axis-parallel (A), piece-wise axis-parallel (B), linear (C) and non-linear (D) classifier.

Nonlinear decision trees

Conventional decision trees can only learn piece-wise axis-parallel problems (see Fig. 2.9). However, nonlinear decision trees can also be constructed. One possible solution is to generate new features as combinations of existing features and to include them in the learning process. This is similar to an explicit transformation from the input space $\mathbb{X}$ to a feature space $\mathbb{F}$.

Ittner and Schlosser proposed the use of original features, the squares of the original features and all pairwise products of original features [IS96]. For a two-dimensional input space, this leads to a features space of five dimensions: $x_1, x_2, x_1^2, x_2^2, x_1x_2$. Note that this transformation is equivalent to the one employed by a polynomial kernel of degree $d = 2$ (and a two-dimensional input space).

Conventional decision trees are univariate, that means they use only one continuous or discrete feature at each decision node. However, multivariate decision trees employing multivariate linear or multivariate nonlinear decision nodes are also possible. For example, nonlinear decision trees using a neural network or an SVM at each node were proposed.

More generally, [YA01] suggested an omnivariate decision tree where at each node a univariate, multivariate linear or multivariate nonlinear decision function is chosen according to a statistical test.

2.5.2 Logic regression

Logic regression is a regression and classification method to construct classifiers with a good interpretability [RL03, KR05, KRLH01]. Formally, logic regression fits a model $f(\boldsymbol{x}) = b_0 + b_1 L_1(\boldsymbol{x}) + ... + b_n L_n(\boldsymbol{x})$ where $L_j(\boldsymbol{x})$ is any Boolean expression of binary feature vectors. A logic regression classifier only uses one Boolean combinations of binary features: $f(\boldsymbol{x}) = L_1(\boldsymbol{x})$.

A Boolean expression consists of binary features which are combined using the following three operators:

- $\bigvee$ OR
- $\bigwedge$ AND
- $\neg$ NOT

An example would be the expression:

$L=$ ("aberration of MDM2" or "aberration of B1143G9" or "aberration of B438N16" or "aberration of CDK4") or ("aberration of B443B14" and "aberration of B1007B5" and "aberration of GliGli" and "aberration of B112B19").

This expression reflects differences between pleiomorphic and dedifferentiated liposarcoma [FSW+02].

Boolean expressions can also be represented graphically as a logical tree (Fig. 2.10). Each element of this tree is a node and has either zero or two children (sub nodes). A node without children is called a leaf and consists of a binary features or its conjugate (negation of the feature). All other nodes comprise operators ("$\bigvee$" or "$\bigwedge$").

The evaluation of all possible Boolean combinations is intractable. Instead, a stochastic search algorithm, simulated annealing [KGV83], is used. A drawback of this strategy is that stochastic search algorithms do not always find a global optimum.

In brief, simulated annealing starts with one tree. A score is defined as the misclassification accuracy of this tree. The tree is changed by a randomly chosen operation ("move"). If the score of the new tree is higher than the score of the old tree then this move is accepted. Otherwise, the move is only accepted with a probability depending on the score of the two trees and an internal parameter of the annealing algorithm (temperature).

Possible operations ("moves") are:

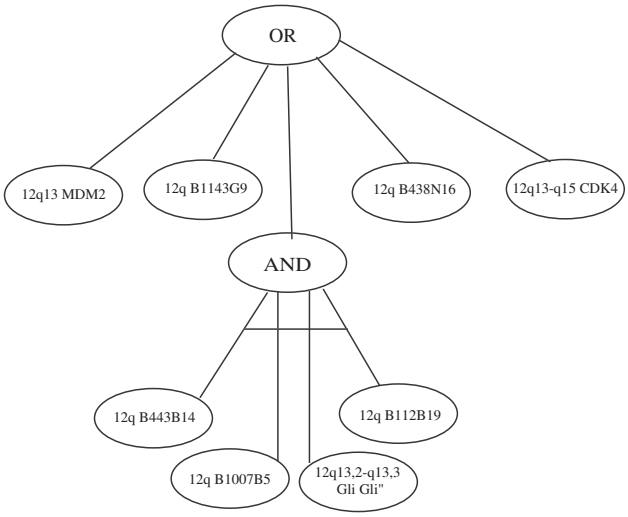


Figure 2.10: And-or-tree representing differences between pleiomorphic and de-differentiated liposarcoma.

- Exchange of two leaves
- Exchange of an operator (”and”, ”or”)
- Growing and pruning (at any node that is not a leaf)
- Growing and pruning (at any leaf)

Neighbours of logic expressions are defined by the number of moves according to this list. Note that any logic expression can be reached from any other logic expression in a finite number of steps.

The model complexity is defined as the (maximum) number of leaves and should be chosen according to a cross-validation or a validation data set. Another approach proposed by the authors for model selection is a permutation test.

2.6 Discussion

In principle, two different kinds of constructive learning algorithms can be considered.

A Construct a machine with a given complexity and minimise the empirical risk. This principle is implemented in artificial neural networks and logic regression.

B Construct a machine with a given empirical risk (e.g., zero) and minimise the complexity. This principle is used in support vector machines and boosting.

For the first type of learning machine, a search over different complexity classes (e.g., number of hidden neurons or number of trees) has to be implemented to avoid overfitting. In the presence of noise, the second type of learning machine requires the optimisation of a regularisation parameter C.

Support vector machine classifiers offer a good generalisation ability, both from the empirical and theoretical point of view. SVMs are also universal approximators and can therefore learn any reasonable decision function up to a desired accuracy. In conclusion, they are a good choice for many classification problems.

Three problems remain:

- The appropriate choice of a similarity measure / kernel for a given classification problem is critical. Even if the resulting support vector machine with this kernel represents a universal approximator, this does not necessarily lead to a large margin [HG03].
- A support vector machine classifier does not reveal explanatory rules.
- The complexity of the learning algorithm is approximately quadratic.

Convex boosting algorithms like Adaboost* were not tested in this thesis due to the fact that no implementation was (publicly) available.

Other classifiers may be the first choice if the training set is very large or includes many missing values. An example of classifiers that perform well despite of missing values are decision trees. Large data sets require algorithms with linear training time complexity like sparse grids [GGT01, GG02].

In the need for explanatory rules, classifiers like decision trees or logic regression seem to be appropriate.

Machine learning involves the search through a space of available decision functions (hypotheses). The goal is to find the simplest hypothesis that fits the available data and prior knowledge. Problems directly derived from this definition that were raised in the introduction are:

1 What is the function space / space of available hypotheses?

2 What is a useful search strategy?

3 How can the quality of the fit between data and hypotheses be estimated?

4 How can the simplicity/complexity of a hypothesis be assessed?

The function space of the classifiers under investigation consisted of

- univariate decision functions (stumps: decision trees with one node)
- piece-wise axis-parallel decision functions (decision trees)
- multivariate linear decision functions (linear discriminant analysis)
- multivariate nonlinear decision functions (SVMs, neural networks, boosting)

Search strategies are

- global optimisation (SVM, boosting)
- local greedy search (decision trees)
- stochastic search algorithms / simulated annealing (logic regression)
- exhaustive search (search for best parameter C of an SVM)

The fit between data and hypotheses can be estimated using different loss functions like $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_\epsilon$ and $\mathcal{L}_{0/1}$. Different loss functions lead to different learning machines.

The complexity of a hypothesis can be estimated by the VC-dimension or the minimum description length.

Chapter 3

Genomic profiles

3.1 Genomics and cancer

3.1.1 Cancer

Cancer is a disease characterised by unlimited cell division, independence from cellular control and invasiveness of cells [HW00]. If a population of tumour cells (a neoplasm) remains confined and does not metastasise or invade neighbouring tissues, it is called a benign (noncancerous) neoplasm; otherwise, a malignant (cancerous) neoplasm. Metastasis is a process by which tumour cells escape their local environment, invade their surroundings, penetrate lymphatic and blood vessels, migrate to remote sites and establish new loci of growth elsewhere in the body. Cancer is a disease of animals and humans, whereas benign tumours also occur in plants [DH96].

The development of tumours, called tumorigenesis, appears to be caused by the combination of activating (genomic) alterations in proliferation-stimulatory genes (oncogenes) and repressing aberrations in proliferation-inhibitory genes (tumour-suppressor genes) [LKV98, Pop00]. These aberrations can be inherited via the germline or occur spontaneously [BGP03].

Oncogenes promote cell growth, cell division and angiogenesis (the growth of new blood vessels) and can be identified in tumours by amplification, by over-expression of their transcript or protein and by activating translocations or mutations. Many oncogenes encode growth factors or growth factor receptors (e.g., EGFR, VEGFR). Oncogenes are often carried by viruses (prefix v, example *v-ras*). Their cellular counterparts are called proto-oncogenes (pre-

fix c, example *c-myc*). Proto-oncogenes become activated through genomic changes such as translocations, insertions and amplifications [Lew00].

Tumour suppressor genes inhibit cellular growth, repress cell-division, inhibit metastasis or induce apoptosis (programmed cell death). They are typically lost or damaged in cancer cells. A loss of one copy of a tumour suppressor gene, either the one contributed by the mother or the one contributed by the father, is called loss of heterozygosity of this gene (LOH, see 3.5). LOH of tumour suppressor genes in cancer cells is often associated with an inactivating point mutation of the remaining copy of the gene. This leads to loss of function of the protective gene products. Examples of tumour suppressor genes are the *Rb* and *p53* gene. *p53* is very important as it triggers the activation of apoptosis and growth arrest pathways [Ore03].

Aberration patterns are strong prognostic markers in cancer patients and can be used to define disease subtypes and predict response to therapy. A close link between genomic instability and cancer progression exists. Furthermore, different stages of tumour progression can be correlated with the acquisition of specific genomic alterations. These range from point mutations to deletions, insertions and chromosomal translocations. For tumour specific aberrations see reference [VK04] for review.

Genomic aberrations also appear to play an important role for understanding hereditary mental diseases (e.g., alcoholism, schizophrenia). However, this is beyond the scope of this thesis.

3.1.2 Chromosomal aberrations

Humans and mice, like many other higher organisms, are diploid; they are characterised by two sets of homologous chromosomes in each normal cell. This is equivalent to a DNA copy number of two for each autosome (2 copies). Humans possess 46 chromosomes (44 autosomes and 2 sex chromosomes), whereas the diploid chromosome number of mice is 40.

Aneuploidy refers to a variation in chromosome number. A missing chromosome from a diploid organism is called monosomy, whereas an additional chromosome is called trisomy. Cells with higher chromosome numbers are called polyploidic. More common examples are Turner's syndrome, in which females only have one X chromosome (XO); Klinefelter's syndrome, in which males have an extra X chromosome (XXY); and one variation of Down's syndrome, called trisomy 21, in which individuals have an extra chromosome 21.

Variations in the arrangement of chromosome segments are called a translocation. Broken chromosomes rejoin into nonhomologous combinations. An inversion is characterised by a change in the order of segments within a chromosome. The expression of a gene may be modified if it is translocated to another chromosome (position effect). An example is the Philadelphia chromosome, which is found in patients with chronic myeloid leukaemia (CML). Here, an abnormally small chromosome 22 is caused by translocation of parts of its genetic material to chromosome 9. This brings the *abl* oncogene from chromosome 9 under the regulatory influence of the oncogene *bcr* on chromosome 22, resulting in increased tyrosine kinase activity of *abl*.

Variations in the copy number of chromosomal segments can be summarised as deletions, gains and high level amplifications. Loss of a chromosomal segment is called a deletion. One extra segment of a chromosome is called a gain and more than one a high level amplification. The amount of loss or gain of genetic material can be quantified as copy number ratios, as described later.

Finally, a genomic profile characterises all genomic aberrations in a given sample or patient (Fig. 3.1).

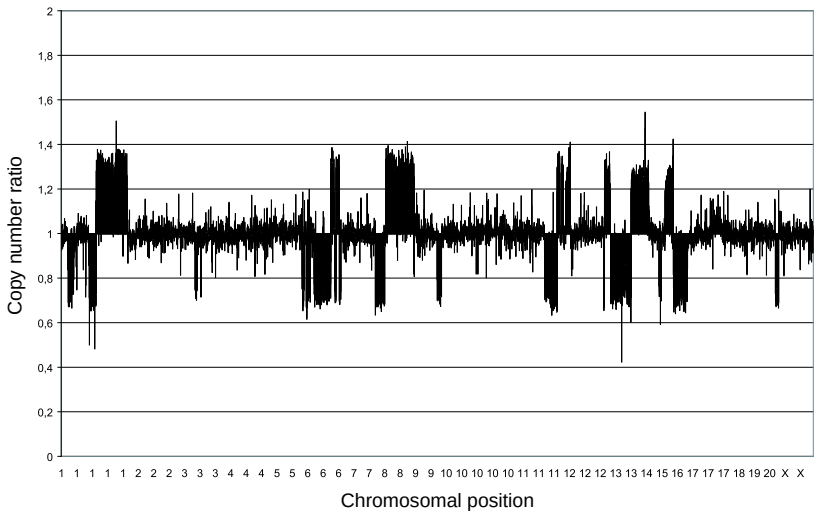


Figure 3.1: Genomic profile of a breast cancer tissue analysed in [SRS+06]. The copy number ratios (y-axis) of all measurements (chromosomal fragments) are ordered by their genomic position (x-axis). The x-axis depicts the chromosomes. Note that each chromosome has a different length and the measurements are not equidistant.

3.2 Classical CGH

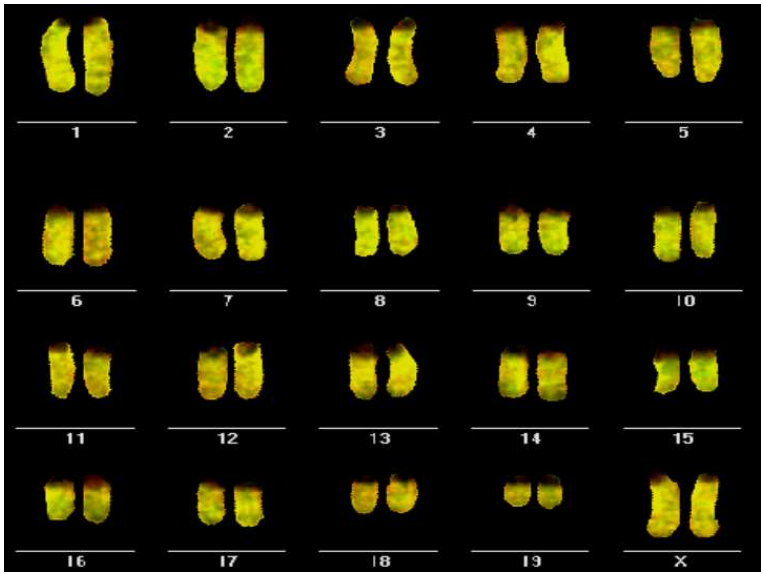
Classical comparative genomic hybridisation (CGH) is a well-established, molecular cytogenetic method, which allows the detection of chromosomal imbalances in entire genomes [KKS+92, DMSJ+93, LJBL00]. This technique is widely used in routine molecular diagnostics. Classical CGH has contributed greatly to our current state of knowledge regarding genomic alterations in cancer (e.g.,[BHDM+95, BWD+96, MHM+01, RJB+02]). More than 150 new CGH studies were published in 2004 and referenced in Medline.

For CGH, test DNA (from the tumour) and reference (normal) DNA are labelled with different fluorochromes (test DNA red and control DNA green) and co-hybridised onto metaphase chromosomes from normal cells. The metaphase-chromosomes are acquired from the blood of a healthy donor. Test and reference DNA compete for binding sites on the metaphase chromosomes. The hybridisation probability depends on the relative DNA copy numbers of test and reference DNA. When hybridising the two differently labelled DNA on a normal metaphase spread, imbalances can be detected as colour changes in chromosomes. The quantitative measurement of the colour ratio profiles along each chromosome yields the DNA copy number differences between sample and reference DNA. Digital image processing and analysis are performed by commercially-available CGH analysis software.

The resolution of classical CGH for detection of chromosomal gains or losses has been estimated in the range of 3-10 million base pairs [KGM+99, BPS+98]. CGH is unable to detect balanced translocations (translocations without copy-number-changes).

Traditionally, CGH profiles have been evaluated according to the International System for Human Cytogenetic Nomenclature (ISCN) [Mit95]. ISCN is a formal language for describing DNA copy number changes, among others. It covers low level gains (”rev ish enh”), high level gains (”rev ish ampl”) and losses (”rev ish dim”). For example, loss of the chromosomal band 4p16 would be specified as ”rev ish dim(4p16)”.

A



B

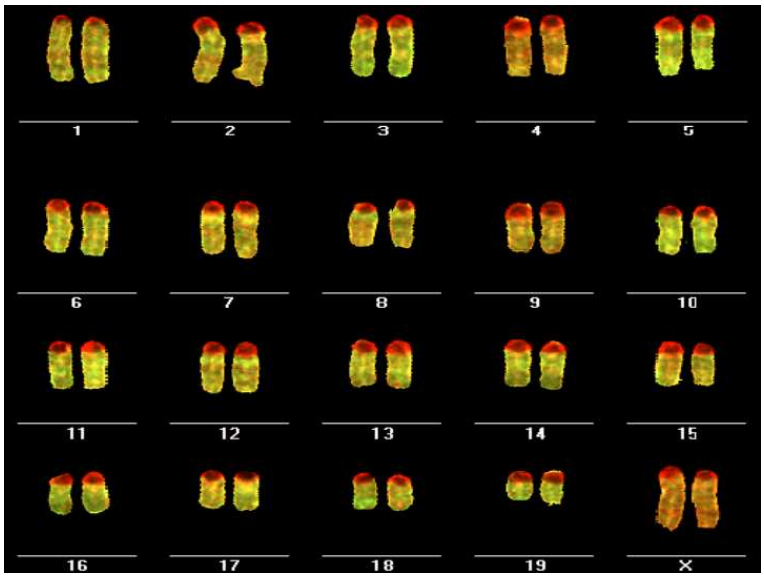


Figure 3.2: Sorted metaphases from a single normal (A) and tumour cell (B) after hybridisation. All 20 mouse chromosomes are shown. Colours encode the copy number ratios. The aberrations of chromosome 4 can be detected by visual inspection. The respective copy number ratios are shown in Fig. 3.3. Figure taken from a draft of [HGS+08].

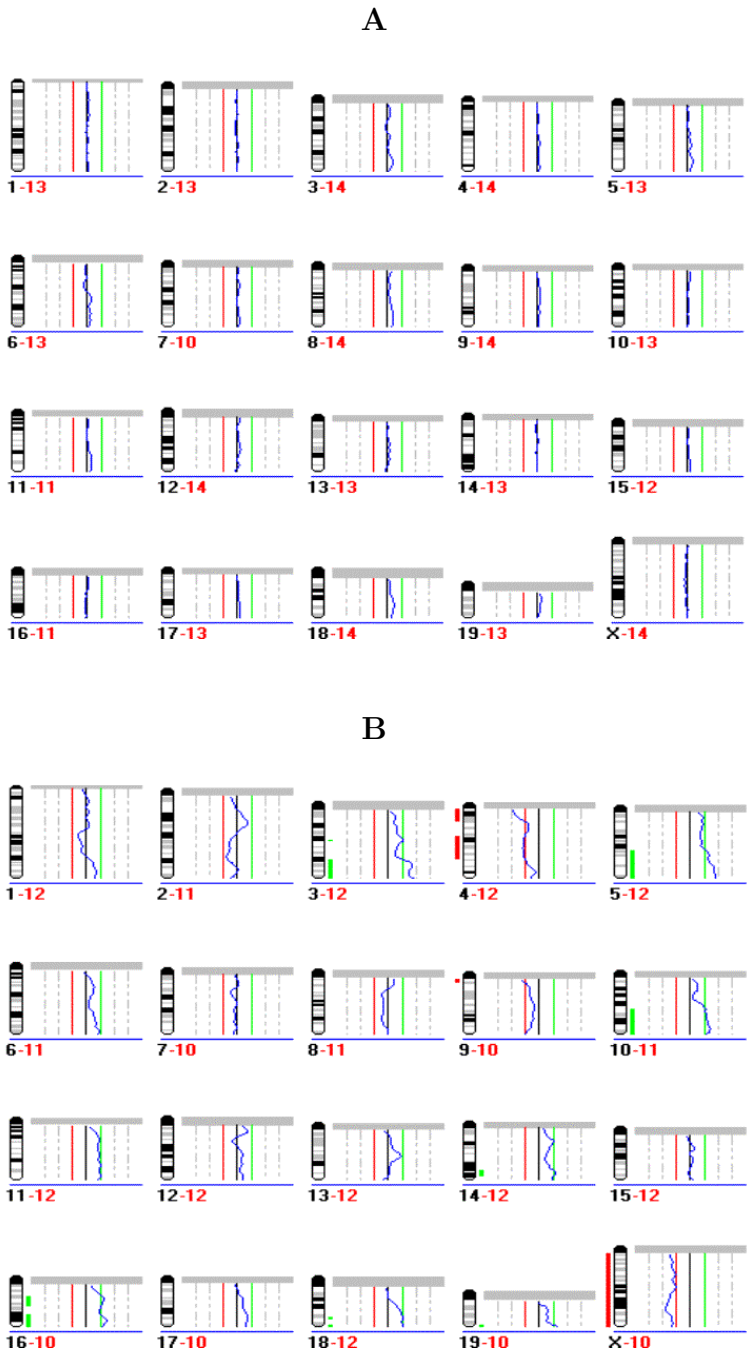


Figure 3.3: CGH profiles from a single normal (A) and tumour cell (B). All 20 mouse chromosomes are shown. The chromosomes are sketched in black, the copy number ratios in blue and the thresholds for gains and losses as thin red / green lines. Losses and gains are marked in red and green next to chromosomes and could only be detected in tumour cells. Figure taken from a draft of [HGS+08].

3.3 Matrix-CGH

More recently, conventional CGH has been further developed to increase genomic resolution [STLS+97, PSS+98, SNS+01, WLS+01]. Matrix-CGH (also called array-CGH) is based upon array technology to detect genomic imbalances. In brief, an array of DNA fragments (e.g., BAC - Bacterial Artificial Chromosomes) is immobilised on a glass chip. Each spot on this array represents a unique DNA sequence or chromosomal locus. Genomic test DNA, isolated from the tumour and reference (normal) DNA are labelled with different fluorochromes (e.g., Cy3 / Cy5) and mixed. The comparative genomic hybridisation of test and reference DNA is performed against genomic fragments, instead of using chromosomes as in conventional CGH. The fluorescence signals for test DNA and control DNA are measured and correlated to the copy number ratios of test and reference DNA (Fig. 3.5).

Finally, gains or losses of chromosomal material can be detected with a resolution that is dependent upon the size and spacing of the immobilised DNA fragments (Fig. 3.4). Typical resolutions are between 100 thousand base pairs and 1 million base pairs. Note that one genomic fragment usually includes several genes. The average length of a BAC genomic fragment is 100 to 300 thousand base pairs, whereas the typical length of a gene is 10-15 thousand base pairs. Higher resolutions in the order of 50-100 thousand base pairs are expected from oligonucleotide arrays [BZL+04].

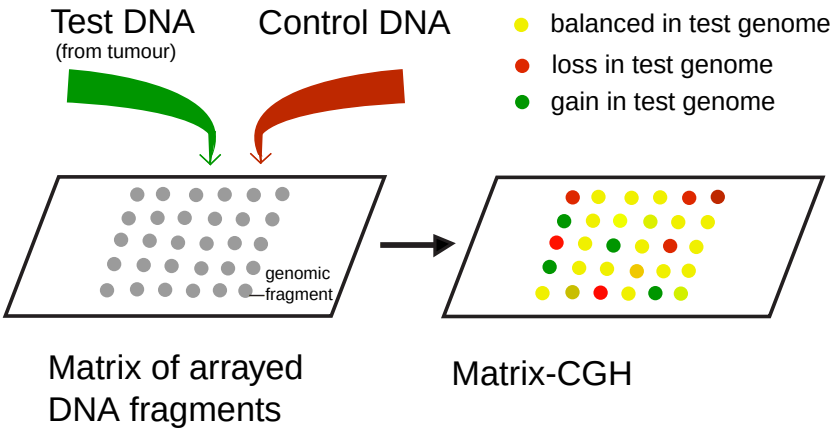


Figure 3.4: Screening of genomic imbalances using matrix-based comparative genomic hybridisation.

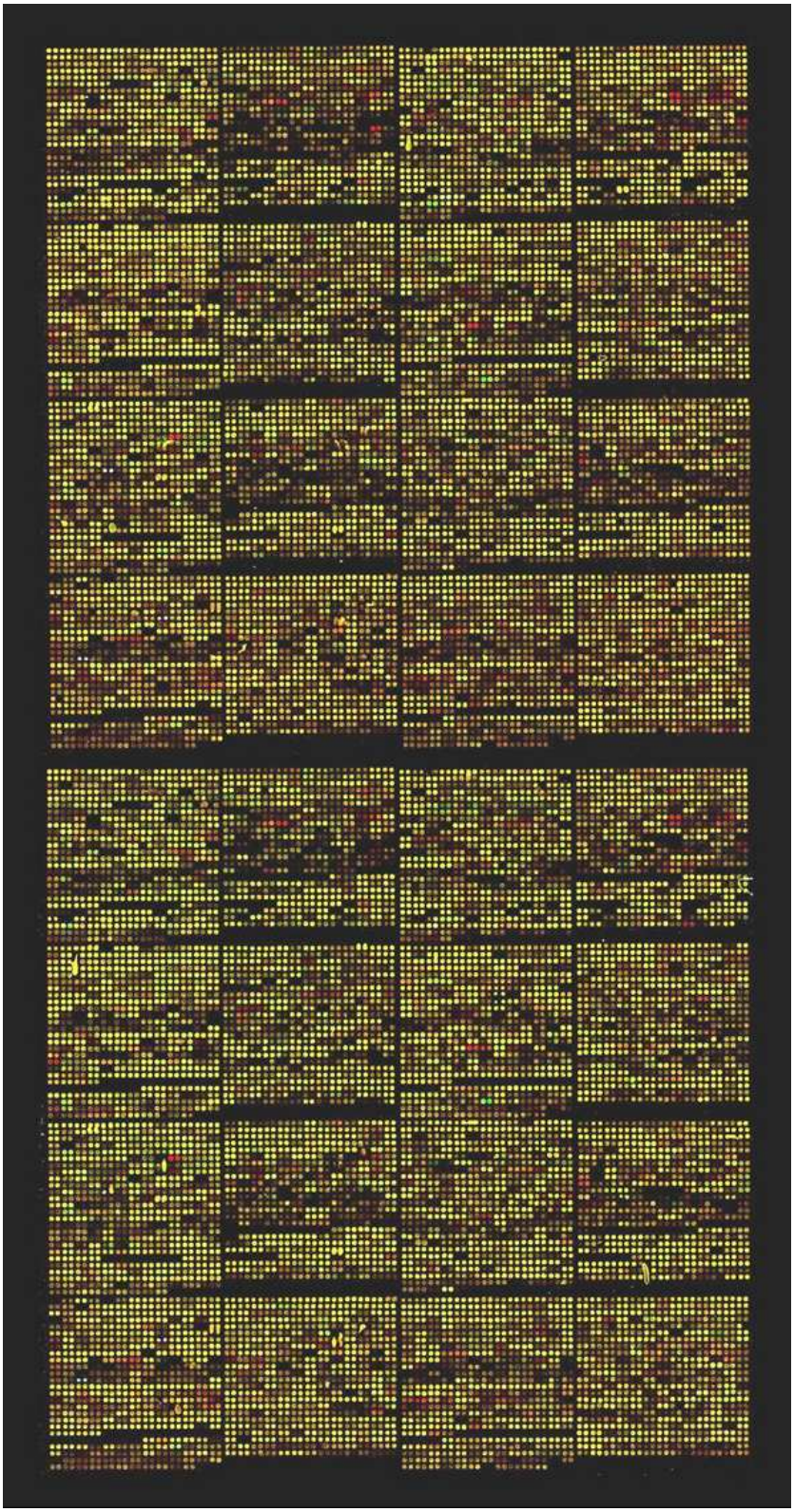


Figure 3.5: Matrix-CGH slide after hybridisation. Each spot belongs to one measurement. This slide belongs to a case analysed in [SRS+06].

3.4 Copy number ratios

The amount of loss or gain of genetic material within a genomic region can be quantified as a copy number ratio. Each genomic region is covered by a genomic fragment and this genomic fragment is used to determine the copy number ratio within this region.

The copy number ratio equals one, when the chromosomal region is balanced by two copies (diploid) of the reference DNA and two copies of the test DNA.

Expected copy number ratios of different aberration states are:

- 0 for a double loss (0 copies of test DNA)
- 0.5 for a single loss or monosomy (1 copy of test DNA)
- 1 for a balanced region without genomic aberrations (2 copies of test DNA)
- 1.5 for a single gain or a trisomy (3 copies of test DNA)

Higher copy number ratios characterise higher degrees of amplification. However, not more than 8 copies are usually observed.

Finally, from a theoretical point of view, the distribution of log-2 copy number ratios may be modelled as a mixture of a normal distribution of balanced copy number ratios, a normal distribution of losses and a normal distribution of gains (Fig. 3.6A).

Copy number ratios from real measurements differ from theoretically expected copy number ratios (Fig. 3.6C). Variations from theoretical copy number ratios may be caused by a mixture of tumour and stromal cells, amplification (PCR) failures, inhomogeneous tumour samples and experimental artefacts. Inhomogeneous tumour samples consist of different clones with different aberration patterns. Mixtures of tumour and non-tumour-tissues can often be resolved using microdissection, a labour-intensive technique by which individual tumour cells are captured under a microscope.

It is interesting to note that an amplification protocol exists that reveals the genomic content of one single (tumour) cell [KSKS$^+$99]. This protocol was used in two studies which are analysed in this thesis [HGS$^+$08, SMH$^+$05].

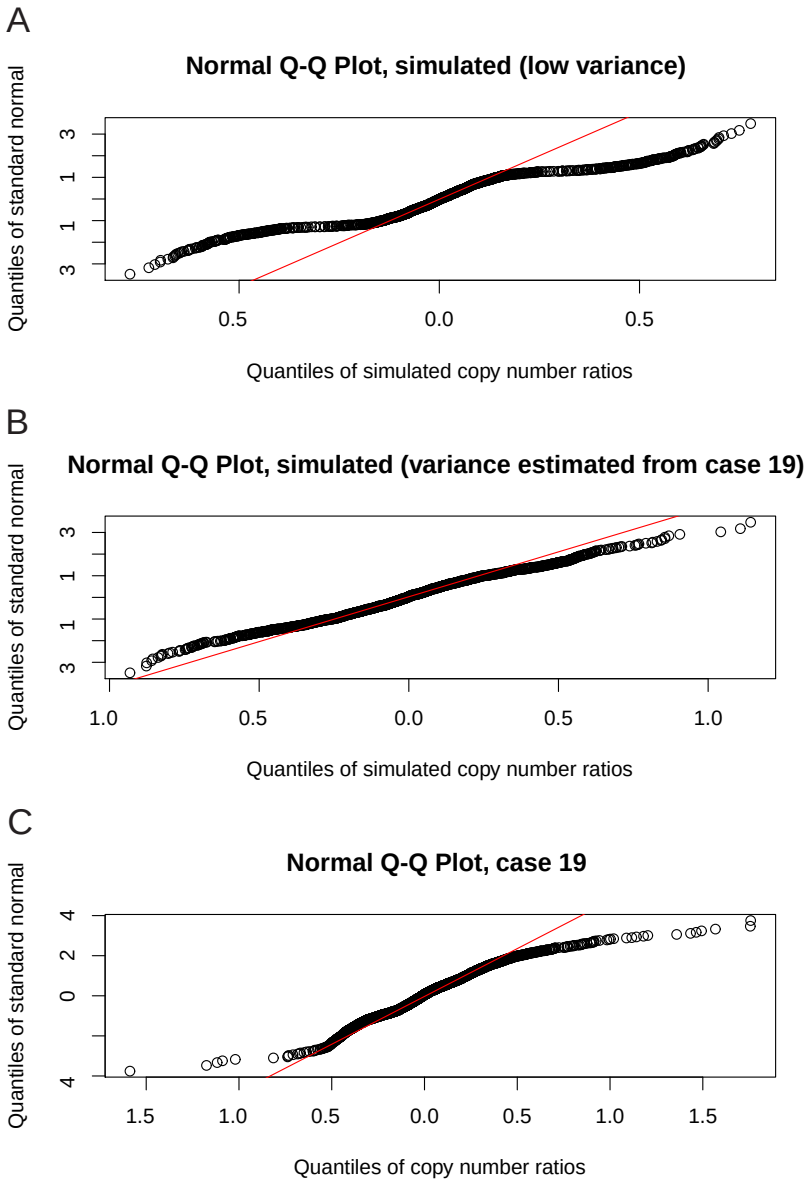


Figure 3.6: Quantile-quantile plot of copy number ratios.
(A) Q-Q plot of simulated ratios with a low variance.
Assumptions: 10% losses (log_2 mean ratio: -0.5), 10% gains (log_2 mean ratio: 0.5).
(B) Simulated Q-Q plot with a higher variance, estimated from case 19.
(C) Q-Q plot of measured ratios from a breast cancer case [SRS+06].

3.5 Loss of heterozygosity

Loss of heterozygosity (LOH) is characterised by loss of one DNA copy, that copy having been contributed either by the mother or the father. Its occurrence in cancer cells is often associated with tumour suppressor genes, where the remaining copy of the gene can be inactivated by a single point mutation [BGP03, OH00].

The detection of LOH is an experimental method to find deletions of genetic material with a high resolution. A test of LOH can only be performed if a given patient or organism is informative (heterozygous) at a given locus in normal cells. This means that this patient or organism possesses two different copies of DNA (alleles) at this locus, one from the father and one from the mother. If the patient/organism has a constitutional homozygosity (two identical copies, one from each parent), this method cannot be used at this locus.

LOH is identified if an informative patient has heterozygosity at a given locus in normal cells (e.g., blood cells) and an absence of heterozygosity at this locus in tumour cells. In brief, the amount of DNA from each allele from normal cells is compared with the amount of DNA from each allele in tumour cells.

To minimise non-informative measurements, polymorphic markers (microsatellites and single nucleotide polymorphisms (SNPs)) are used. A polymorphism is a DNA sequence alteration with at least two different alleles at a locus in a population, where the frequency of the most common allele is less than 99%.

3.6 Gene expression profiles

Although this thesis deals with genomic profiles, we will shortly discuss gene expression profiles, as many methods can be utilised to assess both genomic and gene expression profiles.

Gene expression profiles reflect the status of RNA transcripts (the first major step in protein synthesis) in cells and tissues. The target probe is messenger-RNA (m-RNA) instead of DNA for genomic profiles. m-RNA is isolated from tumour and control cells, reversely transcribed to cDNA (complementary DNA), labelled with two different fluorescent dyes and co-hybridised to

complementary probes. Finally, the fluorescence intensities reflect the relative expression value.

3.6.1 DNA microarrays

cDNA microarrays or oligonucleotide microarrays can be used to measure the expression of thousands of genes in a single experiment. Briefly, an array of DNA spots, namely genomic DNA fragments or oligonucleotides, is attached to a (glass) chip. The labelled cDNA is then co-hybridised against these DNA spots.

3.6.2 CESH

Comparative expressed sequence hybridisation (CESH) is a technique that reveals gene expression patterns according to chromosomal locations, in a manner similar to the way in which CGH detects copy number changes [LWC$^+$01, LWW$^+$03, GWHMR$^+$04, VDWPW04]. In brief, reverse-transcribed test and reference RNA are differentially labelled and co-hybridised to normal metaphase chromosomes. The resolution of CESH is low compared to microarray gene expression arrays but no prior sequence information of genes or cloning is required. Furthermore, CESH can be performed by using existing CGH / fluorescence in situ hybridisation expertise, equipment and software.

3.6.3 Comparison of genomic profiles with gene expression profiles

The features of genomic profiles are genomic fragments, whereas the features of gene expression profiles are EST (expressed sequence tags) or cDNA clones. Consequently, the feature values of genomic profiles are copy number ratios, whereas the feature values of gene expression profiles are gene expression ratios.

One noteworthy difference between genomic profiles and gene expression profiles is the importance of local correlation patterns of copy number ratios. Genomic aberrations often affect not only single genes, but also larger regions or even entire chromosomes. The correlation between copy number ratios of adjacent genomic fragments therefore is higher than the correlation of gene expression ratios of adjacent genes.

Another difference is that the normal state of genomic profiles is known (2 copies) whereas the normal expression profile of normal cells differs with respect to the cell type and metabolic condition of the cell.

In addition, copy number ratios above or below a threshold have a clear biological meaning (gain, loss, or balanced). Defining such a threshold is much more difficult for gene expression ratios.

3.7 Therapeutic interference points

The final goal and an immense challenge of cytogenetic cancer research is the development of specific anticancer drugs. Such drugs should inhibit the growth-stimulatory activity of oncogene proteins and reactivate the growth-inhibitory effect of tumour-suppressor proteins. Targeted anticancer drugs should have a higher level of efficacy and specificity and fewer side effects than cytotoxic drugs currently used.

Examples of such drugs are the kinase inhibitor Gleevec and the monoclonal antibodies Rituxan, Herceptin and Avastin [HZ03].

Gleevec (imatinib mesylate) inhibits a family of tyrosine kinases and is in clinical use for treatment of chronic myelogenous leukaemia (CML) and gastrointestinal stromal tumour(GIST) patients [Saw03]. This kinase inhibitor targets the activation of a protease through fusion (CML) and by activating point mutations (GISTs).

Avastin is an anti-angiogenesis agent currently being used in metastatic colon cancer patients [IL04]. Herceptin (trastuzumab) targets the *HER2/neu* oncogene in breast cancer patients [RRB$^+$01]; Rituxan (rituximab) the *CD20* gene of patients with non-Hodgkin's lymphoma [CLB$^+$02]. Many other drugs are in the developmental stages (e.g., [Hor04, Saw03]).

At this point bioinformatics comes into play. Cancer is a heterogeneous disease and specific therapies will only cure a subgroup of patients. The identification of such a responder subgroup for each new drug is a prerequisite for clinical success.

Moreover, even with specific anticancer drugs, a major challenge of cancer treatment is tailoring the appropriate therapy for a specific patient. Rituxan, for example, is patient-specific administered as a single therapy or in combination with chemotherapy.

Chapter 4

Workflow for machine learning of genomic profiles

4.1 Introduction

In this chapter, a machine learning strategy for detecting aberration patterns associated with a particular tumour type or disease state is proposed. Two important aspects of such a workflow are a multi-resolutional analysis and an automatic assignment of aberrations.

Traditionally, genomic and gene expression profiles are analysed by machine learning without considering the local correlation pattern of the measurements. Here, I discuss ideas for a multi-resolutional analysis of genomic profiles.

Copy number ratios reveal genomic alterations, i.e. gains and losses. It seems therefore appropriate to discretise copy number ratios. The challenge here is to find optimal thresholds that define genomic aberrations and separate different (tumour) classes.

Genomic profiles are often incomplete. The machine learning community uses basic methods for handling missing values (e.g., [WF00]) only. In this chapter, I discuss methods for imputing missing values that are motivated by statistics and the biology of genomic profiles.

In addition to traditional issues of machine learning, I also consider the task of explaining the results derived from machine learning to a user who might not be familiar with this field. This includes a case-based analysis of the

classification, an estimation of the classification certainty and the competence of the classifier.

The workflow focuses mainly on support vector machines (SVMs) and decision trees as classifiers. Several similarity measures / kernels are compared with regard to their applicability for genomic profiles.

Finally, the workflow includes the following steps (Fig. 4.1):

1 Data preprocessing and feature selection
   - Data preprocessing
   - Handling missing values
   - Multi-resolutional analysis
   - Discretisation (assignment of losses and gains)
   - Feature selection

2 Classifier design and evaluation
   - Classifier selection
   - Classifier adaptation
   - Classifier assessment

3 Case-based analysis of a classification
   - Qualitative explanation
   - Competence estimation
   - Classification certainty

My contributions are a preprocessing method for classical CGH data [STJE05], a wavelet-based analysis of genomic profiles (presented as a poster at the Gordon Research Conference on Molecular Genetics, July 18-23, 2004, Oxford), a qualitative assessment of classifiers (presented as a talk at the 26th annual conference of the German Classification Society (GfKl), July 22-24, 2002, Mannheim), an explanation component for classification algorithms [SMF+03] and the workflow itself (not yet published).

The idea of the classical CGH preprocessing was conceived in cooperation with Stefan Joos. I programmed a prototype and supervised the final implementation by Bernhard Tausch. The imputation strategy in section 4.2.4 was developed together with Daniel Stange.

The explanation component was also discussed in a diploma thesis by Jasmin Müller [Mül02], which I supervised.

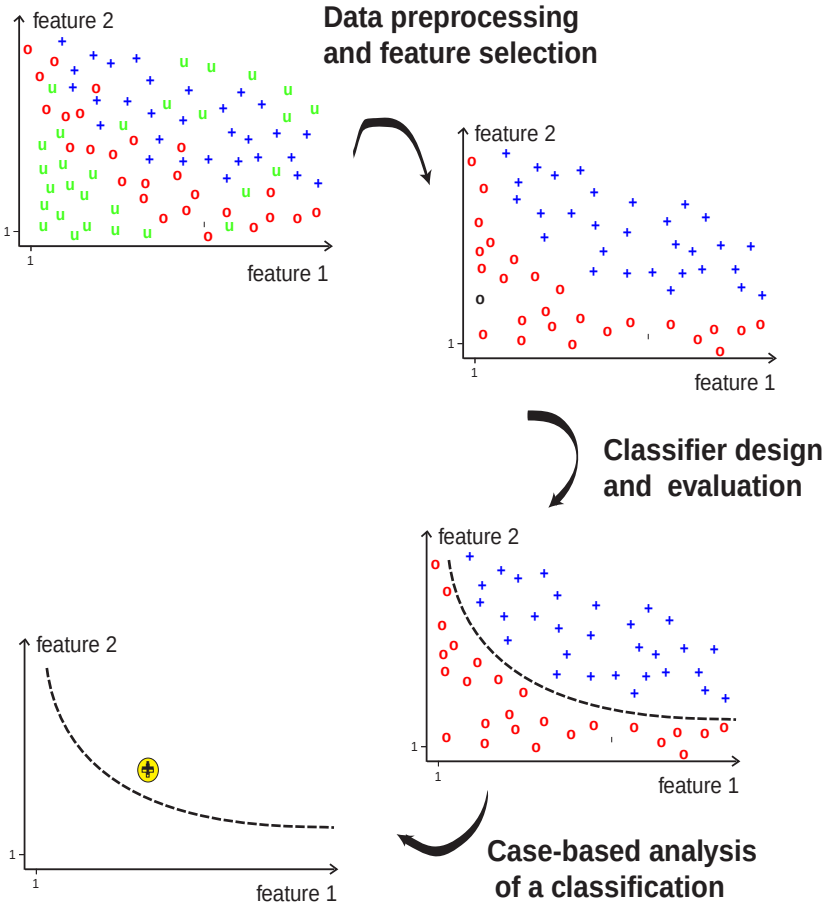


Figure 4.1: General workflow for machine learning of genomic profiles as suggested in this chapter.

4.2 Data preprocessing and feature selection

The preprocessing of data sets is the first step in a successful data mining workflow [WF00] and includes

- Quality-control of the data (filtering)
- Handling missing values and
- Multi-resolutional preprocessing.

The specific preprocessing steps depend on the type of genomic profile and the data set. Important characteristics of each type of genomic profile are summarised in table 4.1.

Profiletype	Dimensionality	Discrete / Continuous	Missing values [%]	Local spatial correlation
LOH	<200	Discrete	40 %	Yes
CGH	100 - 300	**Discrete** or continuous	< 5 %	Yes
Matrix-CGH	300 - 10.000	**Continuous** or discrete	< 5 %	Yes

Table 4.1: Short description of the different types of genomic profiles. The high number of missing LOH values is caused by non-informative measurements. CGH and matrix-CGH-profiles can be analysed as discrete or continuous profiles. Discrete genomic profiles consist of ordinal measurements (loss < balanced < gain). The percentage of missing values of CGH data depends on the inclusion or exclusion of measurements from error-prone regions (e.g. centromeres). All numbers were estimated by the author of this thesis. A local spatial correlation between the measurements characterises all genomic profiles.

4.2.1 Preprocessing of classical CGH data sets

Program CGH-profiler

Chromosomal imbalances detected by classical CGH are described in a complex syntax based on the International Standard for Cytogenetic Nomenclature (ISCN, [Mit95]). For example, loss of the chromosomal band 4p16 would be specified as "rev ish dim(4p16)". This semantic description of chromosomal imbalances hinders a large-scale statistical analysis across different experiments, e.g. for finding aberration patterns associated with a particular disease type or state.

Together with Bernhard Tausch and Stefan Joos, I therefore developed the program CGH-profiler, which circumvents the ISCN nomenclature by automatically assigning a mean (or median) copy number ratio to each chromosomal band. Finally, this is the basis for an automatic assignment of losses, gains and high-level gains.

CGH-Profiler imports data from different CGH system vendors and directly transfers the data into a table format that is readily accessible for subsequent statistical analysis. CGH-profiler comes with different consistency checks, calculates various statistics and automatically assigns a mean (or median) copy number ratio to each chromosomal band. Import of CGH profiles from different CGH system vendors is already supported; its extension to other systems can be readily achieved through Perl scripts.

CGH profiler can also be used to analyse comparative expressed sequence hybridisation (CESH) data. CESH reveals gene expression patterns according to chromosomal locations in a similar manner as CGH detects chromosomal imbalances (see section 3.6.2).

I compared losses and gains automatically detected by CGH-profiler with those described by conventional CGH analysis (encoded in ISCN) for two data sets (data not shown) and found a high degree of accordance. Notably, conventional CGH evaluation often characterises large regions as gain or loss whereas the ratio value, as determined by the programme CGH-profiler, is only altered in part of the entire region.

Data mining of CGH profiles requires a matrix representation of those profiles. An alternative to my approach is an ISCN-to-matrix parser [BC01]. Such a parser is useful for large repositories of CGH studies (e.g., `www.progenetix.net`, providing more than 10818 cases from 383 publications). However, a direct transformation of profile values to a matrix representation

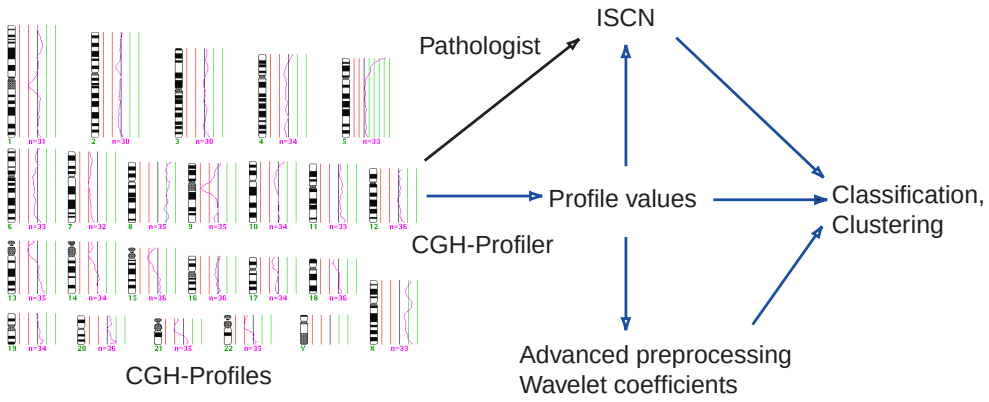


Figure 4.2: Workflow for analysing classical CGH profiles using the program CGH-profiler. CGH-profiler translates CGH profiles into a table of profile values which can be further used for data mining tasks like classification and clustering. Historically, CGH-profiles are encoded using the ISCN syntax.

is more efficient.

The program CGH-profiler has only been used for CGH-analysis in humans so far. An extension to CGH profiles in other species can be easily achieved by adopting the mapping file used for band assignments.

Implementation and usage

CGH-profiler deals with genomic profiles of classical CGH data sets. Here, a profile value is a measurement of a given biological case, a chromosomal location and a hybridisation (to a specific metaphase). Each hybridisation can be considered as a repeat (of the measurement). The number of profile values of each chromosome corresponds to the length of this chromosome.

The program CGH-profiler includes the following processing steps:

- Conversion of CGH profile values to a meta format independent of the used CGH system
- Profile cleansing, consistency check of the chromosomal length
- Interpolation to a given length (adjustable, e.g., 128 points per chromosome) by using cubic (Akima) splines
- Calculation of median copy number ratios from all hybridisations (repeats) of a given case
- User defined exclusion of certain regions (centromeres, telomeres, tumour specific bands)
- Assignment of mean (or median) copy number ratios to chromosomal bands

The calculation of median copy number ratios from all hybridisations is equivalent to combining repeated measurements. Here, the median is used to deal with outliers. The assignment of mean (or median) copy number ratios to a chromosomal bands can be considered as a reduction of the resolution. However, the reason behind this step is to calculate a number with a biological meaning. The choice of the mean or median takes the user of the program.

The CGH profile values must be exported from a commercial CGH system. To become independent of the particular CGH system, the exported profile values are then transformed to a meta format using a Perl script. The resulting meta format file includes all profile values of all cases for all chromosomes. The parsing and transformation of CGH profile values from

two popular CGH systems, namely CytoVision (Applied imaging, `http://www.appliedimagingcorp.com/`) and Isis CGH (Metasystems, `http://www.metasystems.de`), are supported. Profiles from other CGH system vendors may be integrated using adapted Perl scripts.

A consistency check of all profile values is performed to exclude wrongly assigned profiles. All profile measurements of a chromosome are excluded from further analysis if the difference between the length of the profile and the mean length of the respective type of chromosome is larger than a user defined threshold (e.g. 15%).

The remaining profile values of a chromosome are interpolated to a new profile of a given length. This is a prerequisite for a consistent merge of all measurements. I used cubic Akima and Fritsch/Carlson splines (polynomials of degree 2) for this interpolation implemented in the matpack library (`http://www.matpack.de`). The number of interpolation points can be defined by the user, the predefined value is 128. The number 128 is optimal for the subsequent signal processing using wavelets.

From all profiles of a given case and chromosome, the median or mean copy number at each interpolation point is calculated. The choice of median or mean is optional to the user.

The CGH measurements of some chromosomal regions (e.g. those containing a large number of highly repetitive sequences) are not reliable [KKP+94], especially after PCR (polymerase chain reaction) amplification of the genomic DNA. The measurements of certain regions should therefore not be used for an automatic analysis. According to the expert knowledge of Dr. Stefan Joos, all centromeres, some telomeric regions, chromosome 19 and the sex chromosomes are excluded. However, the user can specify all critical regions in a configuration file. The ratios of all excluded regions can be marked as NA (value not known) or balanced.

The mean or median profile of each case and chromosome can be mapped to an ISCN-400-ideogram (schematic chromosomal representation) without subbands [Mit95] so that a single mean value is assigned to each chromosomal band. The predefined mapping file is based on the ISCN-400-ideogram and a resolution of 128 interpolation points. E.g., band 1p36 is located from 1/128 to 13/128 on an ideogram. The mean value of this band is therefore the mean of the profile values 1,..,13. This data representation is the starting point for further analysis. Using threshold values the medium copy numbers can be readily translated into semantic expressions, namely losses (threshold <0.75), gains (threshold >1.25), high level gains (threshold >2) and balanced.

This CGH-preprocessing method has been published [STJE05] and can be downloaded from `http://www.dkfz-heidelberg.de/ibios/archive/ressources/CGHProfiler`.

4.2.2 Preprocessing of matrix-CGH data sets

The fluorescence signals for test DNA and control DNA are measured by a scanner (e.g., Axon 4000B scanner, Axon Instruments, Burlingame, USA). An example of a resulting image is given in Fig. 3.5 (chapter 3). Note, that some spots (encircled measurement areas) are disturbed by dirt, spotting errors and hybridisation failures. Each slide is therefore manually checked by the experimenter. This process is supported by an automatic filtering of spots (measurements) without errors. See also section 4.2.4.

Filtering

Filtering criteria are

- Ratio of intensity and local background (e.g., lower threshold 3 for inclusion)
- Ratio of mean and median intensity of a spot (e.g., upper threshold 1.3 for inclusion)
- Standard deviation of duplicates (repeated measurements, e.g, upper threshold 0.25 for inclusion)

Finally, spots with errors are marked and excluded from further analysis. Spots that have errors for just a few cases are treated as missing values.

Normalisation

A normalisation procedure corrects for systematic spatial or intensity biases. Here, the same principles apply as for normalising cDNA-microarrays [Itt05] and a good choice is a block-wise normalisation by Loess [BIAS03].

Duplicate treatment

Each genomic fragment is generally spotted in several copies. This is equivalent to a repeated measurement and enables a statistical assessment. The

standard deviation of all duplicates is used as filtering criteria (as stated before). Finally, the copy number ratio of a genomic fragment is calculated as the median value of all included measurements (from a given sample).

Assignment of chromosomal mappings

Each genomic fragment represents a segment of the DNA and is characterised by its chromosomal position (in kilo bases). However, due to the on-going sequence analysis of the genome this chromosomal position was often unstable. Therefore, a re-assignment of the chromosomal position of all genomic fragments at the time of the analysis was necessary. However, for future applications of this workflow (and fully sequenced genomes) this step should no longer be necessary.

State-of-the-art in identifying genomic aberrations

Genomic aberrations often affect not only single genes, but also larger regions or even entire chromosomes. Subsequently, methods were developed that recognise regions with approximately the same copy number ratio. Some methods return the position of a region only, whereas others try to characterise a region as loss, balanced or gain (Fig. 4.3). However, all methods assign the same copy number ratio to all fragments within a region of a case.

Examples are:

GLAD (based on adaptive weights smoothing) [HST+04] Glad estimates the breakpoints of piecewise constant functions. Each piecewise function is fitted by the adaptive weights smoothing procedure. The number of breakpoints is determined by a penalised likelihood. Glad returned biologically meaningful regions for the data analysed. Furthermore, Glad could also assign losses and gains to each region using a clustering algorithm. However, the provided assignments were not always biologically reasonable (according to biological experts) and finally not used.

aCGH (based on a Hidden Markov Model) [JMM+04] Jong et al suggested a Hidden Markov Model, where each state represents one region with similar copy number ratios. The transition probabilities are the copy number differences between two regions. The number of breakpoints is calculated by the AIC (Akaike Information Criterion).

DNAcopy (based on circular binary segmentation) [OVLW04] This method recursively splits genomic profiles into two or three segments with equal copy numbers. A split is performed iff the t-statistic of the split is higher than the t-test statistics of a random split. The method stops if no further split is meaningful.

Smith-Waterman dynamic-programming [PRM+05] This algorithm is based on the Smith-Waterman dynamic programming. It selects regions where the copy number ratios of adjacent genomic fragments are above (below) a threshold t. The selection of t is crucial. Price suggested the use of the difference between a male and a female hybridisation (at the y-chromosome) as a first guess. Next, a permutation test of each region is performed and regions without significance are removed. This methods seems to be appropriate for mental disorders and germline aberrations. However, for tumour profiles it is intractable due to their high number of aberrations.

CLAC (clustering along chromosomes) [WKP+05] is a clustering approach to detect chromosomal losses and gains. Briefly, an agglomerative clustering algorithm is used for each chromosomal arm. Subtrees representing gains or losses are selected according to the size of the subtree, the differences of the copy number ratios at the edges of the subtrees and the average copy number ratio of the subtree. Finally, a false discovery rate (FDR) is calculated according to the number of genomic fragments selected in a tumour and a normal (control) case. Problems with this method are the unstable implementation and the required normal-normal (control) hybridisations.

Hot spot detection using the fused lasso [TW07] is based based on the lasso, a penalised regression algorithm. The constraints of the lasso are estimated based on the mean and the median absolute deviation of smoothed genomic profiles (using lowess).

In the end, I used Glad for estimating the change points of genomic profiles (but not for the assignment of losses and gains). As two reviews [LJKP05, WF05] recently revealed that DNAcopy may outperform Glad, future applications should consider DNAcopy.

4.2.3 Preprocessing of LOH data sets

A special preprocessing of LOH data sets is not necessary.

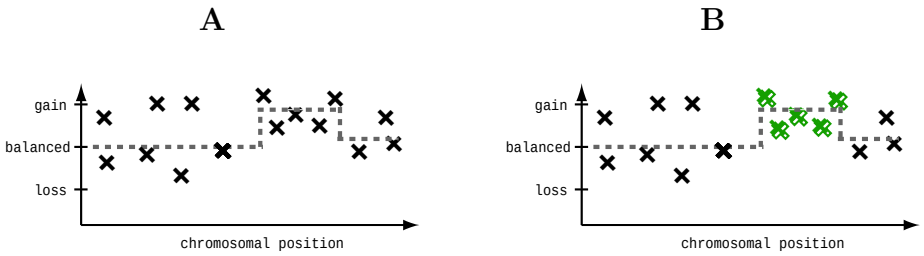


Figure 4.3: Identification of copy number changes. Each cross depicts one measurement, the x-axis the chromosomal positions and the y-axis the copy number ratios. Some algorithms detect the edge of a copy number change only (A), whereas other algorithms additionally assign gains and losses. Measurements that were detected as chromosomal gains are visualised as green wide crosses (B).

4.2.4 Handling missing values

Reasons for missing values are often experimental errors like irregularities in the production and detection of spots, dirt and scratches on the slides, inhomogeneous hybridisation and low signal-to-noise ratios. Measurements with a high variance of all replicates are excluded and therefore also lead to missing values. Figure 3.5 (chapter 3) shows some of these errors for a matrix-CGH data set.

The distribution of missingness can be divided into [SG02]:

- Missing at random (MAR),
- Missing completely at random (MCAR) and
- Missing not at random (MNAR).

Missing at random occurs when the distribution of missingness does not depend on the (true) feature values of missing data. However, "MAR allows that the probability of missingness depends on observed but not on missing data" [SG02]. If the distribution of missingness depends on missing data, then the distribution of missingness is called missing not at random. Missing completely at random means that the distribution of the missingness does not depend on observed or missing data and is a subtype of MAR. Most imputation algorithms require MAR. However, the distribution of missingness for array-CGH data is sometimes MNAR. The higher probability of missingness of lower copy number ratios is caused by a lower signal-to-noise-ratio.

Generally, the following strategies to deal with missing values are possible:

- Complete-case-analysis

- Imputation of missing values
- Model estimation with missing values

Complete-case-analysis

In a complete-case-analysis, all cases with at least one missing feature value are discarded and the analysis is based on complete cases only. If the dimensionality of the data set is high, this strategy will dramatically reduce the number of available cases. A complete-case-analysis is reliable for a MCAR distribution only and therefore should not be applied if the probability of missingness is related to any feature or the class identifier. The bias introduced by a complete-case-analysis may be reduced by re-weighting the remaining complete cases according to the distribution of all cases or the population. For genomic profiles, the complete-case-analysis method is not applicable due to the high dimensionality of the data sets. Most of the cases suffer from at least one missing value and would therefore be discarded. Imagine a data set with 300 genomic fragments and 1% missing values. A complete-case-analysis would discard $(1 - 0.99^{300}) = 95\%$ of all cases.

After all, I used a complete-case-analysis for cases with missing class labels. That is to say, all cases with missing class labels were discarded.

Imputation

Imputation strategies discussed in the literature comprise [SG02]:

- Single imputation
- Multiple imputation
- Maximum likelihood estimation

Single imputation replaces a missing item with one plausible value, whereas multiple imputation replaces it with many (different) values. Maximum likelihood estimates missing values according to a parametric model of the observed data.

Single imputation includes the following strategies [SG02]:

- Imputing univariate means
- Imputing from univariate distributions

- Imputing conditional means
- Imputing from conditional distributions

The replacement of missing values by the mean (median) value of the corresponding genomic fragment is a strategy commonly used. However, it reduces variances and neglects covariances. Discrete genomic profiles (classical CGH) can only be replaced with the median. Yet, the median is almost always the balanced situation and therefore a "safe" strategy at the expense of the power to detect differential aberrated genomic fragments.

Imputing from univariate distributions means that a missing value of genomic fragment x_m is replaced by a random sample of all observed values $x_1, ..., x_n$ for this genomic fragment x (hot deck) or from a parametric model of the distribution. This method also neglects correlations. Moreover, it may insert genomic aberrations in balanced regions and is consequently not discussed further.

Imputing from conditional means is often based on regression methods. The values of the missing values are estimated using a regression model based on other genomic fragments or samples. This method can be applied only if a correlation structure within the data set exists. Regression models based on linear regression [SAG+05], k-nearest neighbour [TCS+01], SVD (singular value decomposition) regression [TCS+01] and support vector regression [WLJF06] have been successfully applied to gene expression data. These methods could also be used to estimate missing values of continuous genomic profiles. For discrete genomic profiles, classification methods like k-nearest neighbour, support vector machines or random forests could be applied.

Joernsten et al. proposed a regression model that combines different imputation methods [JWWO05]. The weight of each imputation method is calculated using a simulation strategy. Briefly, a given data S set is imputed using the k-nearest neighbour method leading to a new data set S'. Thereafter, missing values for this data set S' are generated with the probability of missing values in the original data set S. This process is repeated 20 times and generates 20 new data sets S^1 to S^{20} with artificially missing values. Moreover, it is required that the missing values of S^1 to S^{20} are distinct from the originally missing values of S. Each imputation method is assessed given the difference between imputed values of S^1 to S^{20} and the original values of S. Finally, the weight of each imputation method is determined according to its performance. The drawback of this method is the computational effort.

Imputation from unconditional distributions is based on a conditional distribution of missing values given the observed values. The missing values may

be replaced by a regression prediction plus a residual error drawn from a normal distribution. Imputing from a conditional distribution almost always leads to unbiased estimates under MAR.

A drawback of single imputing is that the uncertainty about missing values is not taken into account in the analysis after the imputation. This leads to an underestimation of the variance.

The underlying idea of multiple imputation is to replace a missing value by many (≥ 2 and typically ≤ 10) plausible values [BLS06]. Multiple imputing reflects the uncertainty about a missing value. First of all, each missing value is replaced by a member of a distribution of plausible values for this missing value. This process is n times repeated. Thereafter, each replaced data set is independently analysed. Finally, all analyses are combined. For a scalar, the combined estimate is just the average whereas its variance is calculated as a modified sum of the variances within each analysis and the variance between all analyses.

Commonly used regression-based multiple-imputing methods are:

- Bayesian least squares
- Predictive mean matching
- Local random residuals.

Bayesian least squares draws the values of missing values from a linear regression plus normal distributed random noise with appropriate residual variance. Multiple values are drawn for each missing value. If more than one value per case is missing, then a multi-variate regression and a joint normal model has to be employed.

Predictive mean matching estimates all values (missing and observed) from a case $\boldsymbol{x}_m$ with missing values according to a linear regression plus normal distributed noise. Thereafter, the case $\boldsymbol{x}_c$ that is closest to these predicted values is determined. The missing values of $\boldsymbol{x}_m$ are replaced by the observed values of this closest case $\boldsymbol{x}_c$. This process is repeated multiple times.

Local random residuals uses a linear regression to predict all values (missing and observed) of a case with missing values. Thereafter, the K most similar cases are determined. One of these cases is randomly chosen and called $\boldsymbol{x}_c$. The missing values of $\boldsymbol{x}_m$ are replaced by the predicted values of $\boldsymbol{x}_m$. Furthermore, the residuals from the randomly chosen case $\boldsymbol{x}_c$ (predicted values $\widehat{\boldsymbol{x}}_c$ minus observed values $\boldsymbol{x}_c$) are added. This replacement algorithm is applied multiple time using different regression estimates.

Biologically motivated imputation strategy

Given the correlation structure of genomic profiles, missing values can be imputed using the copy number ratios of genomic fragments adjacent to the missing value. Nevertheless, the measurements considered should be restricted to the same chromosomal arm. Measurements from two different telomeres (chromosomal ends of two different chromosomes) are not correlated, even though they are next to each other in a matrix representation of the data set.

For discrete genomic profiles, an imputation strategy is as follows (Fig. 4.4):

- A missing value is imputed by an amplification iff all six measurements next to it are amplifications and belong to the same chromosomal arm.
- A missing value is imputed by a deletion iff all six measurements next to it are deletions and belong to the same chromosomal arm.
- Otherwise, the missing value is imputed by assuming a balanced measurement.

The size of the kernel (number of neighbours; six in the previous example) depends on the resolution of the array and the signal-to-noise-ratio.

Model estimation with missing values

Model estimation with missing values is based on algorithms that can cope with missing values. Examples are decision trees [Qui93] or a special implementation of support vector machines [PDBSDM05] using an adapted loss function. As no implementation of this SVM was publicly available, it was not applied in this thesis.

Quinlan compared different methods of decision trees to deal with missing values [Qui89]. A simple but effective strategy of decision trees to deal with missing values is to use surrogate splits. A missing value is replaced by the value from the attribute with the most similar partitioning (of all cases). More advanced methods explore all subtrees of a node with a missing value and calculate the probabilities of different class assignments. For the induction of the tree, missing values are not considered while calculating the splits, but the information gain is adapted accordingly.

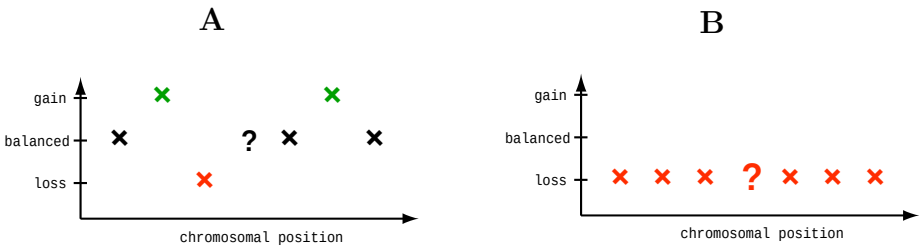


Figure 4.4: Imputation of discretised genomic profiles motivated by their biological correlation structure. The missing value (question mark) is replaced by a deletion iff all observed values are deletions too (B). Otherwise, it is replaced by a balanced measurement (A).

Conclusion

Based on the theoretical considerations presented before, the following methods to deal with missing values of genomic profiles should be taken into account.

Missing values of continuous genomic profiles can be imputed using

- mean values of adjacent values
- values estimated by a regression or classification method

Missing values of discrete genomic profiles can be imputed using

- median value (most often the balanced state)
- values estimated by a classification method

The imputation of LOH data sets seems not to be useful according to the high number of approx. 40% not informative (missing) values. Therefore, robust algorithms, that can deal with missing values, should be applied instead.

Finally, I used the following strategy to deal with missing values of genomic profiles: If a feature or case included too many missing values it was excluded. Features or cases with no more than 5 or 10% (depending on the data set) missing values were imputed using the values of adjacent genomic fragments. For LOH data, the decision tree algorithm C5.0 was applied without any imputation.

4.2.5 Multi-resolutional preprocessing

The combination of several copy number ratios of adjacent genomic fragments decreases the resolution and enhances the reliability of the result. Single gains and losses without support from adjacent genomic fragments can be discarded as they often represent experimental errors or genomic fragments with a wrongly assigned chromosomal position.

Ideas for a multi-resolutional preprocessing of genomic profiles include

- Moving average,
- Identification of segments with similar copy number ratios and
- Wavelets.

Alternative solutions to deal with the curse of dimensionality would have been:

- Increasing the number of samples and
- Incorporation of previous knowledge.

Previous knowledge was used for excluding genomic fragments with a high error rate (e.g., genomic fragments with cross-hybridisation problems). The number of samples was always limited by the availability of tumour probes and financial limits. For future applications of this workflow, higher genomic resolutions will be possible due to increased sample numbers.

Moving average

The simplest algorithm is the combination of overlapping or adjacent measurements using a sliding window approach. I used this method successfully for the LOH data. The drawback of a moving average is that it does not preserve the edges of aberrated regions.

Identification of segments with similar copy number ratios

More advanced methods detect chromosomal regions with similar copy number ratios and assign the same (estimated) copy number ratio to all measurements inside a region. These methods were discussed in section 4.2.2.

After all and for continuous genomic profiles, I used the Glad algorithm that searches for piecewise constant functions. For each detected region, it assigns

the copy number ratio of an estimated piecewise constant to all measurements inside this region.

Wavelets

One idea for a multi-resolutional preprocessing is based on a wavelet transformation. Wavelets are mathematical functions that divide genomic profiles into different frequency components with a resolution adequate to their scale. The wavelet transformation is a refinement of the Fourier transformation. The underlying idea of all transformations is that the transformed representation of the data facilitates the analysis of the data.

Wavelets have been successfully applied for denoising of images, image compression (JPEG2000), EEG (electroencephalogram) and ECG (electrocardiogram). Recently, wavelets have also been used for denoising of continuous genomic profiles [HSG+05].

Mathematically, wavelets decompose a signal (a genomic profile) into a set of basis functions [Mal99]:

$$Wf(u,s) = \int_{x=-\infty}^{\infty} f(x)\psi_{u,s}^{*}(x)dx. \tag{4.2.1}$$

$\psi_{u,s}^{*}(x)$ are the complex conjugated basis functions. Each basis function is called a wavelet, has a mean value zero

$$\int_{x=-\infty}^{\infty} \psi_{u,s}(x)dx = 0 \tag{4.2.2}$$

and is generated from a basic wavelet (mother wavelet) by scaling (parameter s) and translation (parameter u)

$$\psi_{u,s}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-u}{s}\right). \tag{4.2.3}$$

The simplest example of a (mother) wavelet function is a Haar wavelet:

$$\psi(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x \leq 1 \\ 0 & else. \end{cases} \tag{4.2.4}$$

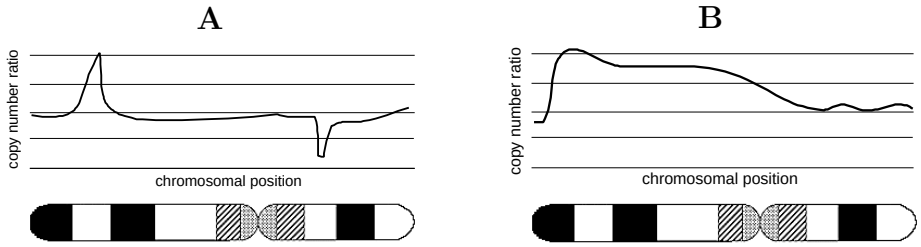


Figure 4.5: Chromosomal aberrations of high and low frequency are plotted against an ideogram (schematic chromosomal map). A) Distinct chromosomal regions: high frequency B) Large chromosomal regions: low frequency

However, I used the Daubechies wavelet family [Mal99] due to their compact support. A closed formal representation of this family does not exist and they are iteratively calculated by a filter bank approach.

The signal-processing terminology is based on the frequency of signals. How is such a frequency defined for genomic profiles?

The frequency of a genomic aberration is determined by its length. Large genomic aberrations are equivalent to low frequencies whereas distinct genomic aberrations have high frequencies (Fig. 4.5). Finally, each aberration can be described one-to-one by its frequency (length) and location (in kB).

Fig. 4.6 shows a wavelet analysis of the cancer cell line HL60. The large aberration of chromosome 5 is reflected by the wavelet coefficient of a low frequency whereas the distinct aberration of chromosome 8 leads to a wavelet coefficient of a high frequency. Please note that the spatial resolution of the wavelet coefficient with a higher frequency is much better than the wavelet coefficient of a lower frequency.

Due to the fact that the genomic fragments on the array-CGH chip are not strictly equidistant, copy number ratios on (128) virtual equidistant genomic fragments using an interpolation algorithm (splines) are computed (using the algorithm described in section 4.2.1). This is a prerequisite for the application of wavelets.

My idea was to use all wavelet coefficients above a given threshold as input of a classifier. In retrospect, this approach did not outperform the use of the original features (copy number ratios after a multi-resolutional preprocessing).

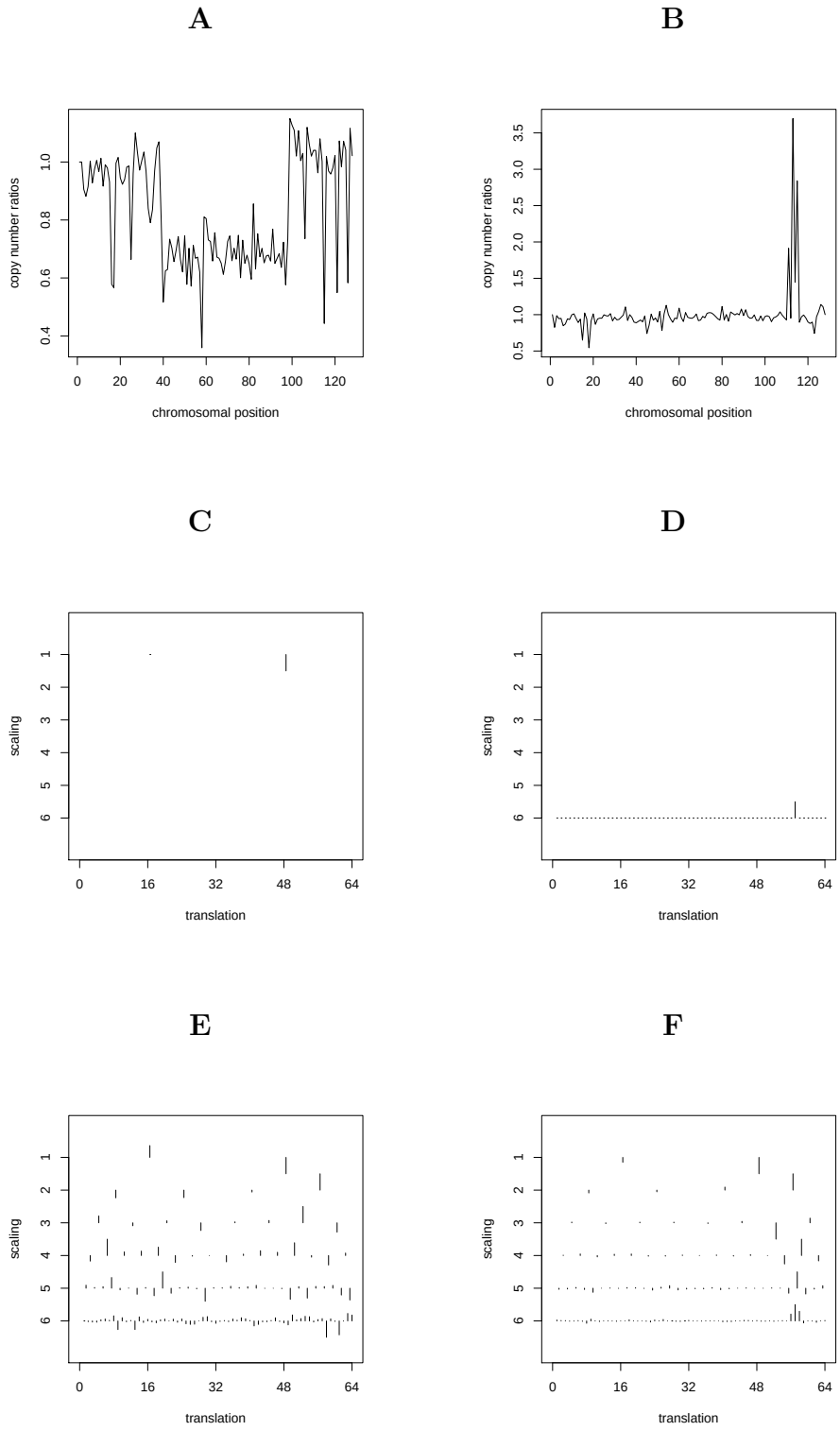


Figure 4.6: Wavelet analysis of the cancer cell line HL60. Genomic profiles of chromosomes 5 (A) and 8 (B), their most important (Daubechies) wavelet components on chromosome 5 (C) and 8 (D), and all wavelet coefficients on chromosome 5 (E) and 8 (F). The underlying matrix-CGH-measurements were kindly provided by Bernhard Radlwimmer. All calculations were performed using the statistical software package R and the wavethresh library.

4.2.6 Discretisation and encoding of features

From a theoretical point of view, a simple discretisation algorithm, using a threshold of 0.5 and 1.5 for losses and gains respectively, would be sufficient. However, due to experimental noise and the fact that a mixture of normal (e.g., stroma) and tumour cells is analysed, this approach is not possible.

Moreover, some tumour types may share a gain of genomic material but could be still distinguished regarding the copy number ratio. By way of example, one tumour type is characterised by a low level gain (copy number ratio 1.5) and the other by a high level gain of chromosome 1q (copy number ratio 2). Therefore, I propose two automatically chosen thresholds for the discretisation of gains and losses.

One way to determine such discriminating thresholds would be to estimate the classification accuracy of all possible thresholds. However, the computational effort of such a simulation (together with a model selection and a cross validation) would have been intractable.

Therefore, I estimated both thresholds by means of statistics. Briefly, I calculated the number of discriminating genomic fragments (between two classes) for each possible threshold. A genomic fragment was called discriminating iff the number of deletions (for thresholds below 1) or amplifications (for thresholds above 1) differed significantly (as measured by the Chi-square test or the Fisher test). Furthermore, I calculated the same statistics 100 times for all possible thresholds and permuted class labels. Next, I calculated the 95% quantile of the number of discriminating genomic fragments for each possible threshold from the random experiments and compared it to the number of discriminating genomic fragments from the original data (Fig. 4.7). Finally, I chose the threshold with the largest difference between the number of discriminating regions in the data set and in the random experiments (as determined by a Chi-square test).

I suggested and applied the described discretisation strategy together with a feature selection (see below) in a study published in [SRS+06] and described in section 5.2.1. In this study, I analysed the best threshold for gains in the interval [1,3] and for losses in the interval [0.3, 1] automatically using the training data only (inside the cross-validation). Finally, the classification accuracy increased from 56% (classification of continous profiles after a multiresolutional preprocessing using "GLAD") to 65%.

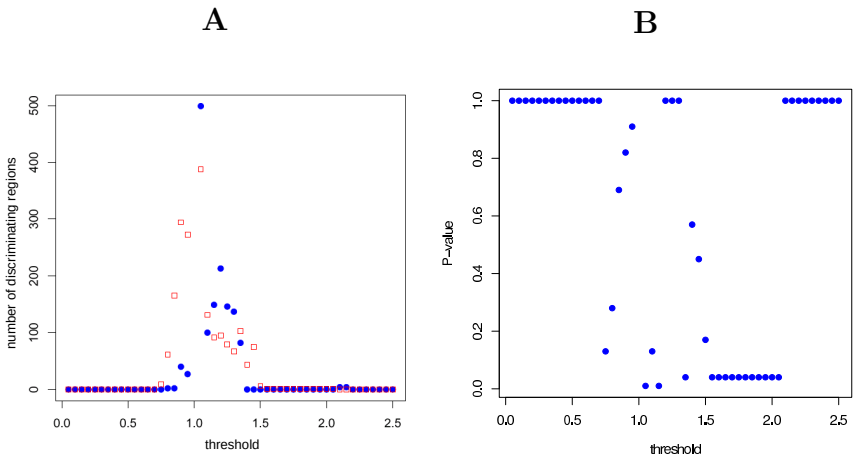


Figure 4.7: Threshold estimation for the discretisation. A) Number of discriminating regions (y-axis) for each analysed threshold (x-axis). Blue (full) dots denote the number of discriminating regions for the analysed data set and red (empty) squares the 95% quantile of the random experiments. B) P-values for the discrimination using each threshold. The best discriminating thresholds are 1.05 and 1.15. Note that the p-values of all thresholds below 1 are above 0.05. This indicates that the deletions of this data set are not as important as the amplifications.

4.2.7 Feature selection

I applied classifiers that can deal with high-dimensional data. Thus, a feature selection is not necessary. However, the feature selection may enhance the classification accuracy and the interpretability of the model.

Following the discretisation of genomic profiles, I selected genomic fragments only that showed aberrations in at least 10% (or 20%) of all cases. That is, each included aberration had to affect at least 4 patients (sample size 20...40). An aberration affecting less than 4 cases is of little biological interest.

The features have to be selected independently from the class labels and with respect to the training set (or an inner cross validation) only. Thereafter, the same feature selection has to be applied to the (unseen) test data.

4.3 Classifier design and evaluation

4.3.1 Classifier selection

Different classifiers were theoretically analysed in chapter 2. Finally, I used support vector machines and the decision tree algorithm C5.0 (in an implementation of Clementine) as classifiers according to their generalisation performance and the availability of stable implementations.

Altogether, a support vector machine seems to be a good choice of a classification algorithm. A linear kernel of an SVM is appropriate for small data sets. A polynomial kernel "counts" the number of aberrations. Therefore, it is interesting for discrete profiles. An RBF kernel is characterised by its universal approximator ability (section 2.4.5). I used an RBF kernel for continuous profiles.

All kernels mentioned do not specifically reflect the biology (and spatial correlation structure) of genomic profiles. An interesting alternative would have been a special (string) kernel (e.g. [RSS05]). However, an appropriate preprocessing of the data set and a standard kernel should be equivalent to a specialised kernel function.

I suggest the use of decision trees for data sets with many missing values (LOH data sets). Interestingly, the performance of the SVM and C5.0 was comparable. This is caused by the low (observed) non-linearity of the given data sets (low number of samples and high dimensionality). Boosting of the decision trees did not improve the results.

4.3.2 Classifier adaptation

For an SVM, the choice of the kernel, the kernel parameters and the regularisation constant C is crucial for a good generalisation performance for a given data set. Subsequently, we search for a parameter vector $\boldsymbol{p} = (p_1, .., p_i)^t$ that minimises the expected risk of an SVM classifier f:

$$\hat{\boldsymbol{p}} = \arg\min_{\boldsymbol{p}} R(f(\boldsymbol{p})). \tag{4.3.1}$$

An analytic solution is not possible. However, an upper bound of the parameter C for SVM-regression and the RBF-kernel can be estimated from the

range of the target variable y [CM04]. In the case of outliers, C should be selected as

$$C = \max(|\overline{y} + 3\delta_y|, |\overline{y} - 3\delta_y|), \tag{4.3.2}$$

where $\overline{y}$ is the mean and δ_y the standard deviation of the target variables y.

For SVM-classification, the estimation of the risk $R(f)$ for each parameter vector $\boldsymbol{p}$ can be based on methods described in sections 2.3 and 4.3.3. I used a cross validation or a leave-one-out cross validation.

The different parameters p_1,..,p_i are not independent. Therefore, a separate tuning of each parameter is not possible.

I employed an exhaustive search within a reasonable parameter space of the support vector machine (as proposed by the authors of the implementation libsvm [CL01]). For each parameter vector $\boldsymbol{p}_j$ the performance of the classifier is assessed (Fig. 4.8). The described grid search is time consuming but can easily be parallelised.

Alternatives to an exhaustive parameter search would have been a gradient descent algorithm using upper bounds of the empirical risk [CVBM02] or a global optimisation algorithm based on Gaussian processes [FZ05].

For a reliable assessment of the classifier performance, the parameter tuning has to be performed using the training data only (see section 4.3.4).

4.3.3 Classifier assessment

An assessment of a trained classifier can answer two different questions.

The first one relates to the separability of two classes and biologically reads as "Are different tumour types distinguishable?" The second question refers to a global feature ranking and reads as "Which aberrations distinguish one tumour type from another?"

Methods to answer the first question are based on an estimation of the classification accuracy and will be discussed in the next part of this section. The second question will be discussed in the third part of this section.

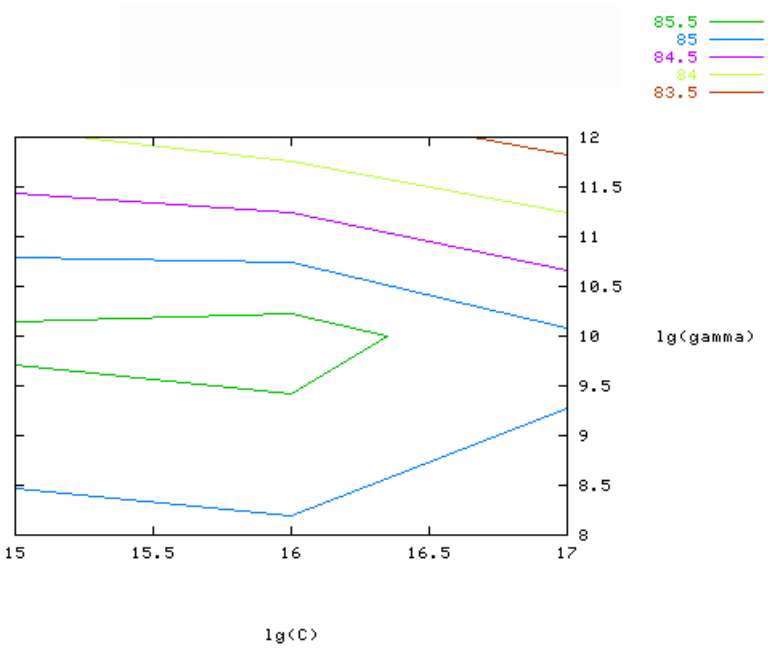


Figure 4.8: Grid parameter search visualised by the SVM implementation libsvm [CL01]. The x-axis depicts the parameter C and the y-axis the parameter γ (both on a logarithmic scale). The classification accuracy is encoded by an equivalent colour.

Quantitative classifier assessment

Different methods for a numerical model assessment were considered in section 2.3 from a theoretical point of view. But which method should be used as a reliable estimate for the reader of a biological journal? Which method reflects the separability of (two) tumour classes?

An empirical comparison of different model assessment techniques revealed diverse results.

Kohavi compared 0.632 bootstrap, leave-one-out cross validation (LOO-CV) and ten-fold cross validation (10-fold-CV), amongst others, using the decision tree classifier C4.5 and data sets from the UCI repository [Koh95a]. The real-world data sets from this repository are commonly used to compare machine learning algorithms. He observed a higher variance of the LOO-CV and a larger bias of the bootstrap. Finally, he recommended a stratified ten-fold cross-validation.

A study of small-sample microarrays using the classifiers kNN (k-nearest neighbour), lda (linear discriminant analysis) and the decision tree algorithm CART revealed a high variance of both LOO-CV and 10-fold-CV [BND04]. LOO-CV and 10-fold-CV showed a comparable performance. In conclusion, the authors recommended the computationally expensive bootstrap.

Molinaro et al compared different resampling methods on simulated gene expression data sets [MSP05]. The classification algorithms lda, dda (diagonal linear discriminant analysis), kNN and CART were applied. The authors concluded that LOO-CV "generally performed quite well", with the exception of unstable classifiers like CART. 10-fold-CV was quite comparable and suggested for larger samples. The authors included a feature selection and in this circumstance a cross validation (LOO-CV and 10-fold-CV) was better than a bootstrap.

A comparison of model assessment methods using an SVM-classifier [ABR+05] and data sets from the UCI repository revealed that a LOO-CV outperforms a 10-fold-CV. However, a boostrap with 100 replicates was better and a bootstrap with 10 replicates worse than a LOO-CV.

From a theoretical point of view, the LOO-CV sometimes overestimates the prediction error. In a data set without a correlation between the feature values and the class labels, a classifier would predict the class according to the majority class of all cases in the learning set. A classifier in LOO-setting would therefore always learn the wrong class (the class that is not left out)

and predict an accuracy of 0% (assumption: balanced design, both class labels have a share of 50%). Such a failure occured in the data set shown in section 5.2.4, where the predicted accuracy was clearly below the 50% expected of a random assignment in a two-class-problem.

To summarise, LOO-CV performs badly for unstable classifiers. LOO-CV and 10-fold-CV have comparable results, although 10-fold-CV is characterised by a higher bias and LOO-CV by a higher variance. For small data sets of approx. 10 samples, a 10-fold-CV and a LOO-CV would be the same. Bootstrap has a low variance but sometimes a high bias.

Finally, I decided to use a LOO-CV estimator for most of the experiments with a support vector machine. However, almost all (biological) conclusions were backed up by another classifier (often the decision tree C5.0) in an implementation of another software package (Clementine). The 10-fold-CV for the decision tree was chosen according to the aforementioned problems of the LOO-CV with unstable classifiers.

Qualitative classifier assessment

The algorithms discussed in this part of the section answer the question "Which aberrations distinguish one tumour type from another?" This is based on a global analysis of the classifier. In section 4.4.1, a case-based analysis of a classifier is introduced. A case-based qualitative analysis answers the question "Why does a given tumour sample belong to tumour type B?"

For separable classes (classification accuracy 100%), I propose an algorithm that identifies feature subsets such that each subset can be used to distinguish both tumour types (classes). Features with a low importance for the classification are recursively discarded and the SVM retrained. An alternative approach would have been an analysis of all possible feature subsets. However, this is an NP-complete problem.

Next, the question arises, whether the subsets found represent statistically relevant features. I use permutation tests and calculate a p-value for each discriminating subset found. The underlying test statistics is based on the hyperplane distance.

The QP optimisation problem of the SVM can not always be solved effectively. However, difficult and time-consuming optimisation problems indicate that the separation of both classes is difficult. Therefore, the learning process of the SVM is stopped iff the time consumed for the QP-problem using

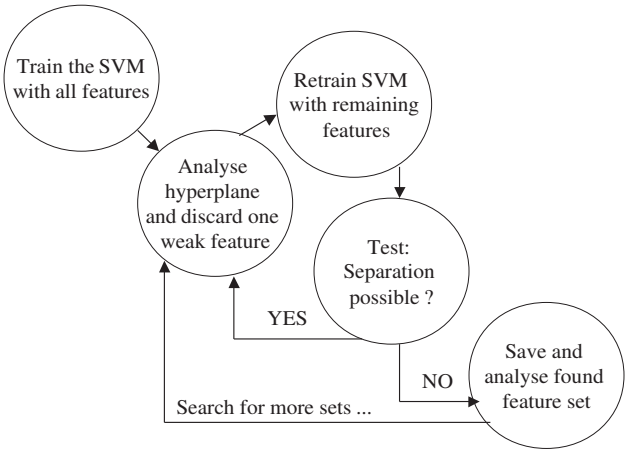


Figure 4.9: Algorithm for the identification of discriminating feature subsets.

a feature subset takes much more time than the original problem with all features.

Taken together, the algorithm reads as (Fig. 4.9):

1 Start with an SVM classifier trained with all features

2 Select the feature with the lowest importance for the classification

3 Discard this feature and retrain the SVM with all remaining features

4 Assess whether a separation of both classes with the remaining features is still possible. If a separation is still possible, then go to step two. Otherwise, a minimal discriminating subset of features was found.

5 Assess the importance of the minimal discriminating subset found. Save an "important" subset.

6 Discard all members of important subsets and redo the analysis (step two).

Finally, and-or-trees can be used to represent the feature sets found (Fig. 4.9). An and-or-tree describes the composition of an expression in terms of sub-expressions which are combined by "and" or "or" nodes. An and-node is true iff all of its successors are true whereas an or-node is true if at least one successor is true.

For non-separable data sets (classification accuracy below 100%), a feature ranking was calculated according to [GWBV02]. Briefly, the most important separating features were identified from the trained SVM classifier according

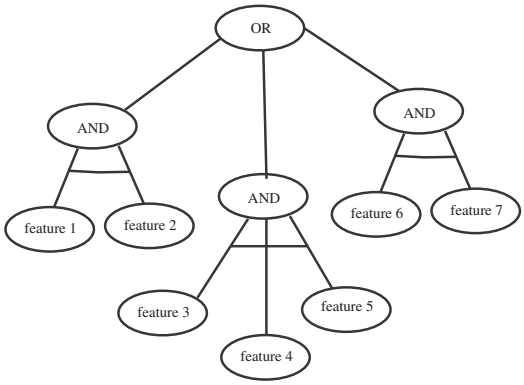


Figure 4.10: Representation of discriminating feature subsets using an and-or-tree.

to the absolute value of each component of the hyperplane direction vector. Guyon et al. also proposed a recursive feature elimination and used the classification accuracy as criterion. However, the differences between the classification accuraccies of feature subsets of varying sizes were not significant. Therefore, the estimation of the optimal size of a discriminating feature subset (given the analysed genomic profiles) was very difficult and finally not used.

4.3.4 Reliability and reproducibility

Reliability and reproducibility are important issues in statistics and machine learning. For classification tasks, reliability and reproducibility are not straightforward, due to the complexity of the algorithms and the size of the data sets. Furthermore, the estimation of the classification accuracy often depends on preprocessing steps.

It is important to base all parameter estimations on the training data set only. For smaller data sets, where cross validation is the best choice, all preprocessing steps have to be included in two nested cross validation loops. The outer cross-validation loop estimates the classification error; whereas the inner cross-validation loop selects the optimal parameter of the preprocessing steps and the hyperparameter of the classifier [RHPM04], [Sal97].

Furthermore, a prerequisite for the interpretation of the classification error is that the test data set is balanced. This means that the test data set contains the same number of cases for each class. Otherwise, the specificity and sensitivity of the classifier have to be taken into account.

4.4 Case-based analysis of a classification result

Tailoring the appropriate therapy for a specific patient is an important challenge in cancer treatment. Genomic profiling in tumour research can improve current diagnostic and prognostic classification schemes and may provide important information for choosing the appropriate therapy. A vast number of classifiers based on genomic or gene expression profiles have been proposed for different diagnostic tasks in the literature. However, there are only a few classifiers in clinical routine use.

Therefore it seems worthwhile to think about strategies to adapt the knowledge from classifiers described in the literature for clinical decision support systems. Such a clinical decision support system should include a core classification algorithm, a knowledge base, interfaces with the hospital information system and an explanation component for the physician.

The classifier and the knowledge base are a trained model of a support vector machine and therefore well established. Yet, little has been done to develop a plausible explanation scheme for the user of a standard support vector machine. But this explanation scheme is indispensable for most clinical applications. Here, I introduce an explanation scheme based on a few features (genomic fragments) with a high explanatory value for a classified case. Furthermore, I propose an explicit competence model to classify cases only which are within the competence area of the classifier. The system estimates the applicability of the support vector machine classifier for an unknown case to be classified and provides a qualitative explanation for each classification.

4.4.1 Qualitative explanation of classification results

To develop an explanation component, it is worthwhile having a look at human decision making.

Imagine the task of classifying fruits as apples. For each fruit, we search for an explanation as to why this fruit was classified as “apple” or as “not-an-apple”. Apparently, a human classifier can easily describe why an object was recognised as an apple or not. Each classification outcome can be explained in a few words, using a few features like the red, green or yellow colour or the ellipsoidal shape. Yet, our explanation of apples depends on the apple we classify. The classification of a little red apple is described due to its colour

and the classification of a blotchy green apple due to its shape.

Another example relates to the important question of marrying. Charles Darwin thought about it in 1887 [GTG99]. He wrote down many arguments for and against a marriage.

> MARRY
> Children—(if it please God)—constant companion, (friend in old age) who will feel interested in one, object to be beloved and played with—better than a dog anyhow—Home, and someone to take care of house—Charms of music and female chitchat. These things good for one's health. Forced to visit and receive relations but terrible loss of time. My God, it is intolerable to think of spending one's whole life, like a neuter bee, working, working and nothing after all.—No, no won't do.—Imagine living all one's day solitarily in smoky dirty London House.—Only picture to yourself a nice soft wife on a sofa with good fire, and books and music perhaps—compare this vision with the dingy reality of Grt Marlboro' St.
>
> Not MARRY
> No children, (no second life) no one to care for one in old age....Freedom to go where one liked—Choice of Society and little of it. Conversation of clever men at clubs.—Not forced to visit relatives, and to bend in every trifle—to have the expense and anxiety of children—perhaps quarrelling. Loss of time—cannot read in the evenings—fatness and idleness—anxiety and responsibility—less money for books etc—if many children forced to gain one's bread.—(But then it is very bad for one's health to work too much) Perhaps my wife won't like London; then the sentence is banishment and degradation with indolent idle fool—
>
> Charles Darwin, 1887 (cited after [GTG99])

How could he come up with a decision? Weighting all features in a linear model? Would a person report the decision making process as $0.1 \times \text{children} + 0.2 \times \text{conversation\_with\_clever\_men\_at\_clubs}$? Charles Darwin based his decision on one important argument only: "constant companion" and wrote down beneath this argument "q.e.d. marry - marry - marry". Finally, he married his cousin one year later.

In general, it has been shown that human decision making is often based on heuristics [GTG99] and a few features only. Conscious decision making has a limited capacity. Therefore, humans take a subset of available features into account only when they decide [Dij04]. Even experts differ from non-experts in their choice of the features used, not in their number [Sha92]. Interestingly, the assumed natural limit for humans in decision processes is approximated as seven features or dimensions [Mil56].

Explanation scheme

The basic idea of explaining a complex, numerical, high dimensional classifier (based on many features) is to approximate the decision function for a

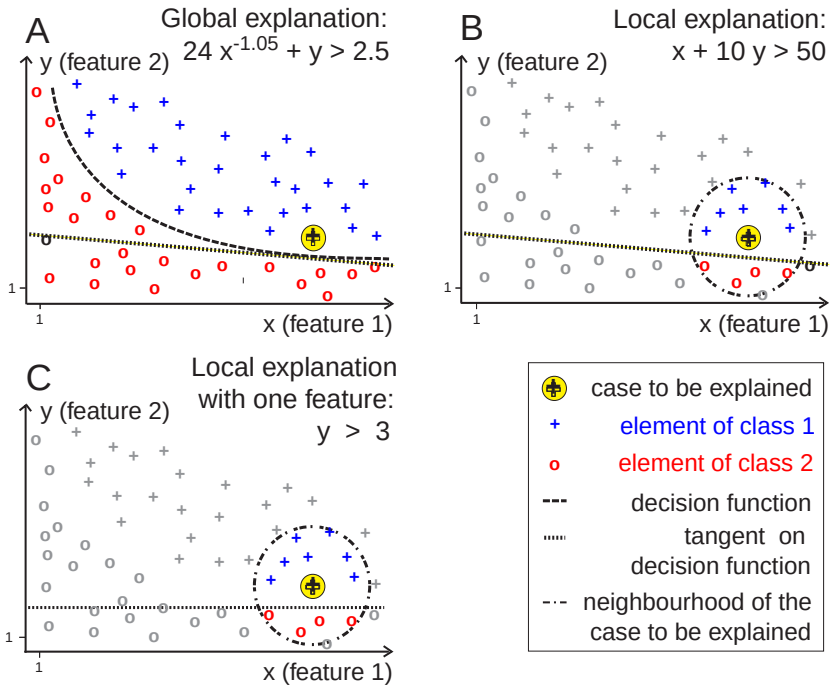


Figure 4.11: Schematic explanation of a classifier for the separation of class 1 from class 2. A) Global, complex and non-linear explanation of the high-dimensional classifier. The tangent on the case to be explained is shown. B) Local explanation by a linear function within a neighbourhood of the case to be explained. The linear function is the tangent of the non-linear classifier (A) on the case to be explained. C) Local simplified explanation after local feature selection. The resulting axis-parallel decision function correctly separates both classes within the small neighbourhood and is based on one feature only.

given case (e.g., a tumour sample) by an easily understandable linear and low-dimensional classifier. This approximation is sufficient within a small neighbourhood of the classified case only and based on a few characteristic features (Fig. 4.11). Because humans usually do not explain their decisions in the form of weighted linear models, the explanation is further reduced to a few characteristic features of the linear model. This can be regarded as a local feature selection in a small region around the classified case. The selected features provide a reliable basis for a case-specific explanation of the classification. Thus we can reformulate the question "How can the outcome of a classifier be explained?" to "Which features are important for the classification of a given case?"

Local approximation

Features with a high explanatory value for a classified case have a large impact on the classification outcome. Variations of these features will likely change the classification result. A case-based sensitivity analysis of all features, i.e. the perturbation of one feature value at a time and monitoring the classification variation, would provide a ranking of all features but would demand high computational costs.

Here, I suggest only analysing the first partial derivatives of a modified SVM decision function (4.4.2) to find features with a high explanatory value.

The original decision function of an SVM (4.4.1) is not differentiable:

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{M} y_i \alpha_i K(\boldsymbol{s}_i, \boldsymbol{x}) + b\right). \tag{4.4.1}$$

$\boldsymbol{s}_i$ are the support vectors and α_i, y_i their weights and class labels respectively. M is the number of support vectors and K the kernel function. The corresponding coefficients α_i, and the bias constant b reflect the solution of a quadratic programming problem and define the position of the separating hyperplane.

Therefore the sign-function is replaced by the tanh-function in analogy to neural networks [Bis95]:

$$f(\boldsymbol{x}) = \tanh\left(\sum_{i=1}^{M} y_i \alpha_i K(\boldsymbol{s}_i, \boldsymbol{x}) + b\right). \tag{4.4.2}$$

For a linearly separable decision function (linear kernel)

$$K(\boldsymbol{s}_i, \boldsymbol{x}) = \sum_{j=1}^{N} (s_{ij} x_j) \tag{4.4.3}$$

the first partial derivatives of the decision function are:

$$\frac{\partial f(\boldsymbol{x})}{\partial x_j} = \frac{\sum_{i=1}^{M} y_i \alpha_i s_{ij}}{\cosh^2 \left(\sum_{i=1}^{M} y_i \alpha_i \sum_{k=1}^{N} x_k s_{ik} + b \right)}. \tag{4.4.4}$$

N is the dimensionality of the input space (number of features).

Similarly, the explanation of a non-linear classifier is based on the first partial derivatives of the non-linear decision function. E.g., for the polynomial kernel

$$K(\boldsymbol{s}_i, \boldsymbol{x}) = (c + \gamma \sum_{j=1}^{N} s_{ij} x_j)^d \tag{4.4.5}$$

the first partial derivatives are:

$$\frac{\partial f(\boldsymbol{x})}{\partial x_j} = \frac{\sum_{i=1}^{M} y_i \alpha_i \gamma d \left(c + \gamma \sum_{k=1}^{N} x_k s_{ik} \right)^{d-1} s_{ij}}{\cosh^2 \left(\sum_{i=1}^{M} y_i \alpha_i \left(c + \gamma \sum_{k=1}^{N} x_k s_{ik} \right)^d + b \right)} \tag{4.4.6}$$

c, d and γ are constants.

Local feature ranking

All features $j = 1..N$ are ranked for each classified case $\boldsymbol{x} = (x_1, x_j, ..., x_N)^t$ according to their explanatory value. The importance of a feature for the classification of a given case is proportional to its explanatory value and calculated from the first partial derivatives.

Since the denominator of the partial derivatives from the linear decision function (4.4.4) is equal for all features x_j of a classified case, the explanatory value of each feature is therefore independent from the classified case and defined as:

$$e\left(x_j\right) = \left| \sum_{i=1}^{M} y_i \alpha_i s_{ij} \right|. \tag{4.4.7}$$

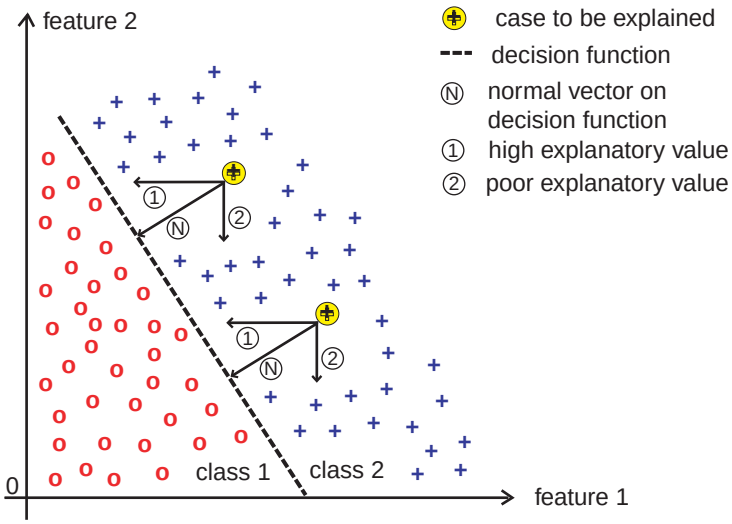


Figure 4.12: Schematic feature ranking of a linear classifier. The feature rankings for the explanations of both classified cases are identical.

From a geometric point of view, the explanation can be obtained from the normal vector of the decision hyperplane. Features with a high explanatory value correspond to a high absolute value of the component of this feature in the direction vector of the decision hyperplane (Fig. 4.12).

The denominators of the partial derivatives from non-linear decision functions, e.g., for the polynomial kernel, are also equal for all features x_j. The explanatory value of each feature $x_j, j = 1..N$ depends on the classified case $\boldsymbol{x} = (x_1, x_j, ..., x_N)^t$ (Fig. 4.13) and is defined as:

$$e\left(x_j\right) = \left| \sum_{i=1}^{M} y_i \alpha_i \gamma d \left(c + \gamma \sum_{k=1}^{N} x_k s_{ik} \right)^{d-1} s_{ij} \right| . \tag{4.4.8}$$

Local feature selection

The features with the highest ranking are selected for the explanation. The number of features presented can be determined *a priori* by the user or calculated by a recursive replacement strategy. This recursive replacement strategy can also be used for the verification of the explanation.

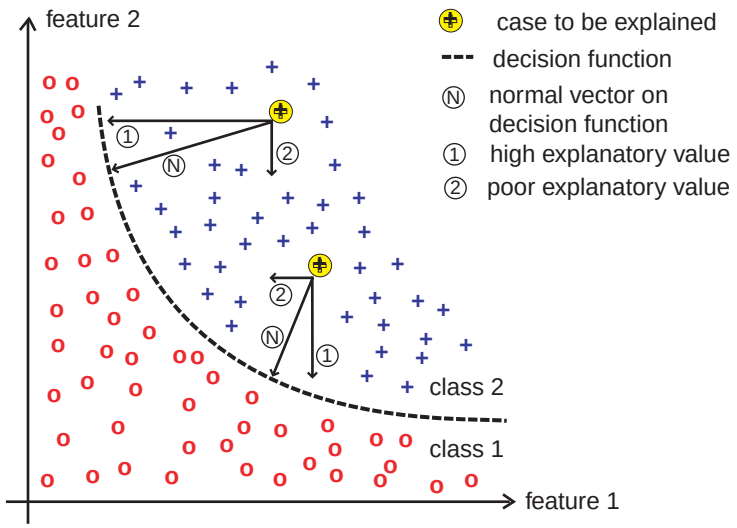


Figure 4.13: Local approximation and feature ranking of a non-linear decision function. Different feature rankings for the explanations of two classified cases were obtained.

The idea of the recursive replacement strategy is to falsify the classification with a minimal number of alterations to the features of the classified case and to use this alteration process as an explanation. The value of the highest-ranked feature of a classified case from class 1 is replaced by the median feature value of all cases from class 2. If the classification of the modified case switches to class 2, an explanation for the classification of the case is found. Otherwise, the values of the next highly-ranked features are additionally replaced. The same strategy can be used for a classified case from class 2.

The resulting explanation comprises all highly-ranked features required for a change in the classification result.

It is worthwhile to mention that the number of features found using the recursive replacement strategy is case-specific, even for a linear classifier.

Adaption for missing and atypical values

To overcome some limitations of my explanation scheme, it was adapted for data with missing values and atypical cases from regions with a low-density distribution. If a feature value is missing for the case to be explained, it is also omitted in the explanation. Furthermore, a feature is skipped if it is

highly-ranked but of little explanatory value for the case to be explained. An example may illustrate an atypical case for a linear classifier. The prostate specific antigen (PSA) marker is normally increased in patients with prostate cancer. So we may use the attribute PSA to distinguish the class "cancer" from the class "control". However, to explain why a patient with a low PSA level has prostate cancer the attribute PSA is not appropriate.

Discussion

I proposed an explanation scheme for support vector machine classifiers based on the selection of highly-ranked attributes in a case-based context of a decision problem. More precisely, the features chosen for an explanation were identified by local linear approximation of the complex decision function. This linear approximation served as a basis for local feature ranking and local feature selection.

An application of this explanation scheme on the liposarcoma data set can be found in section 5.2.3.

The new idea is to use a case-based analysis of the classifier. A global ranking of all features (calculated from the decision hyperplane) would provide a rough estimate of their case-based importance only. For an example data set, shown in section 5.2.3, the global and case-based ranking differ even for a linear kernel, e.g. due to missing values.

My explanation concept based on a few features is similar to the decision-theoretic concept of lexicographic ordering. Lexicographic ordering refers to a rank-order of all features. A comparison of attribute values is done attribute-by-attribute until a feature discriminates between the two classes. The strategy of lexicographic ordering assumes that a significant change of the discriminating feature outweighs all less important features [GTG99]. Interestingly, this decision strategy is followed by oncologists in the assessment of therapeutic options for primary breast cancer by focussing on the most important attribute "chance of cure" [RSM87].

The application of artificial neural networks (ANNs) in diagnostic decision making has been proven to show great potential (see [Lis02] for review). Heckerling compared different feature ranking algorithms of ANNs, yet from a global and not a case-based point of view. Although the explanation concept presented here was developed for SVMs, it can be readily extended to other classification schemes such as ANNs. The first order partial derivatives of an ANN decision function could be evaluated by applying a sensitivity analysis

or differentiable activation functions analogous to a paper by Hashem et al. [Has92]. I focussed on SVMs because of their superior generalisation ability.

It might be argued that complex classifiers like SVMs and ANNs should be replaced by intuitive classifiers like decision trees, being more adequate for explanations. However, a decision tree is inappropriate to deal with non-linearly combined features. Boosting simple and weak classifiers [FS96] also lacks an intuitive explanation ability. Furthermore, the natural feature limit for humans in decision processes should be taken into account. Thus, even in the case of a more intuitive or linear classifier, the number of features for an explanation should be restricted.

Another explanatory method for support vector machines was suggested by Barakat and Diederich [BD05]. Briefly, all support vectors are extracted, their class labels are discarded and predicted by the SVM. Then, a rule-based machine learning technique (e.g., C5.0) is applied to this data set. The goal here is to reveal rules upon which the predicions of the SVM are based.

An alternative to machine learning approaches is a case-based reasoning system (CBR) [Kol92]. A CBR infers the class label of a new case from similar training cases. These cases represent the basis for an explanation of the classification. The support vectors of the trained support vector machine could be used in a similar manner. The similarity measure (metrics) of a CBR also provides information about the classifier. A sensitivity analysis of this similarity measure for a new case could therefore reveal a case-based explanation, analogous to the explanation scheme described here.

Another alternative would be a classic decision support system. As it requires that the knowledge about different classes is explicitly given in a formal language (e.g., first order logic, frames), it is not appropriate. A DSS embodies a knowledge database, an inference engine and often an explanation component. The formal representation of the knowledge remains a great challenge, particularly in an emerging and dynamic field like molecular genetics.

Future research should be directed towards the explanation of classification problems with more than two classes. Furthermore, effective arguments should be adapted for each user considering his/her values, preferences and knowledge [CM01]. Consequently, I propose a usefulness factor for each feature to strengthen the relevance of an explanation for the user. The usefulness of an explanation could depend on the newness (information gain) and relevance for clinical actions (actionability, [ST95]). If a physician is not

familiar with an important attribute of the classification task then this attribute should be included in an explanation. For 'actionability', a drug expert system for physicians should base its explanations on alterable features (like the drug dosage) instead of fixed features (like the age of a patient).

4.4.2 Competence estimation

A machine learning classifier will return an answer for each question posed. However, some questions are beyond the knowledge of the classifier. Imagine that a classifier for estimating the female breast cancer risk is applied to a male patient. If the classifier has never been trained with male breast cancer patients, the classifier is unlikely to be competent. The same problem occurs if a classifier is trained with adult patients but is applied to children.

Therefore, I propose a competence check of the classifier as a part of a decision support system. This competence check should also incorporate knowledge about the applicability of a classifier in a given domain. A classifier may be useful for women only, or not competent for classifying patients above the age of 55. Therefore, important features of the patient have to be requested from the user (or the hospital information system).

Generally speaking, a classifier seems to be competent if it has been trained with cases similar to the case to be classified (Fig. 4.14). However, if the case to be classified is atypical, then the competence of the classifier regarding this case is very weak.

Mathematically, the case to be classified has to be drawn from the same distribution as the training data. Therefore, the competence of a classifier can easily be computed using a multi-dimensional (kernel) density estimation. Another algorithmic solution would be the use of a one-class support vector machine [SPST$^+$01].

For neural networks, Bishop [Bis94] proposed a similar "confidence measure"

$$\sigma_y(\boldsymbol{x}) = \sqrt{p(\boldsymbol{x})}, \tag{4.4.9}$$

where $p(\boldsymbol{x})$ is the unconditional density of the data.

Bishop suggested to use a threshold for the novelty based on an estimation of $\sigma_y(\boldsymbol{x})$ with respect to an independent test set. Furthermore, he recommended to perform the novelty test prior to the preprocessing of the data.

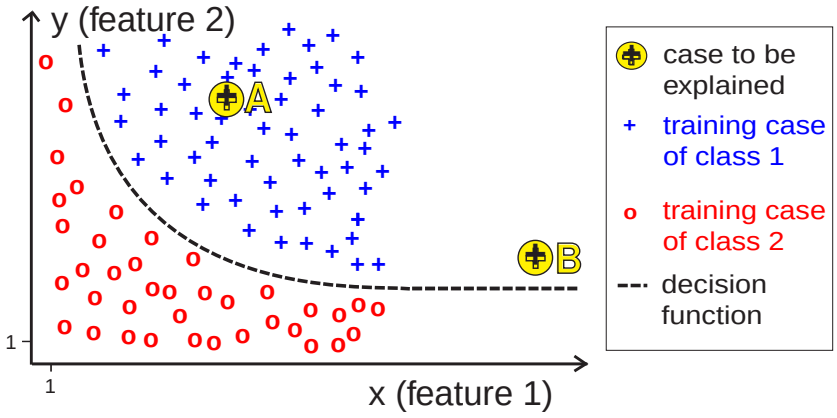


Figure 4.14: The competence of the classifier is estimated from the density of training cases in its neighbourhood. The classifier is much more competent regarding case A compared to case B.

Habermas et al [HZET07] proposed a case-based "reliability framework" for various classifiers. It is based on a local accuracy estimation

$$r(\boldsymbol{x}) = 1 - \frac{1}{M_v} \sum_{i=1}^{M_v} \mathcal{L}_{0/1}(\boldsymbol{v}_i, f(\boldsymbol{v_i})) \text{ for all } \{\boldsymbol{v}_i | D(\boldsymbol{v}_i, \boldsymbol{x}) < d_v, \boldsymbol{v}_i \in \mathbb{V}\}. \tag{4.4.10}$$

The "validation pool" $\mathbb{V}$ is drawn from the same distribution as the training data. For a case to be classified, similar cases $\{\boldsymbol{v}_i | D(\boldsymbol{v}_i, \boldsymbol{x}) < d_v, \boldsymbol{v}_i \in \mathbb{V}\}$ from the validation pool are selected which are within a maximal distance d_v from the query case $\boldsymbol{x}$. Based on this case-specific validation set, a local accuracy is determined. If the local neighbourhood is empty, that means all cases of the validation pool $\mathbb{V}$ have a distance from the query case $\boldsymbol{x}$ greater than d_v, the case is referred to as a "no-neighbour case". Crucial is here the choice of the distance function D and the distance threshold d_v.

4.4.3 Classification certainty

The classification certainty provides the user of a decision support system with a number determining the reliability of a classification result. Classified cases next to the decision hyperplane are much more unreliable compared to cases further from the decision hyperplane.

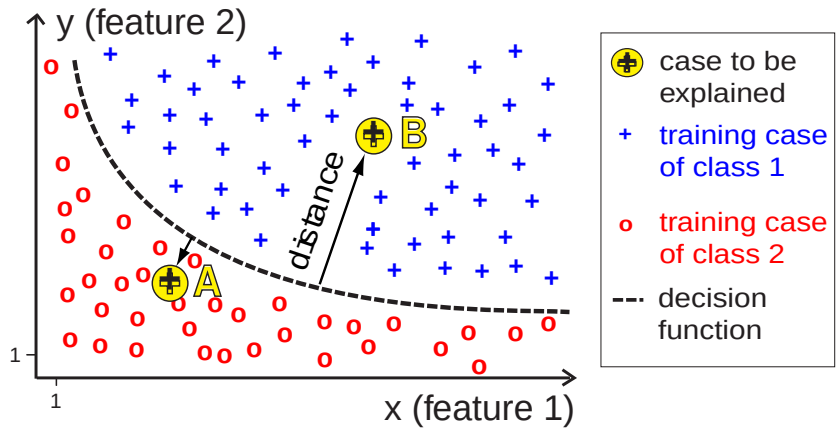


Figure 4.15: The classification certainty is calculated from the distance to the hyperplane. Accordingly, the classification of case A is much more unreliable than the classification of case B.

Additionally, I used this classification certainty to visualise the assignment of tumour probes and the corresponding feature patterns (e.g. Fig. 5.4 and 5.16).

If the distances to the hyperplane were normally distributed, then a posterior probability $P(y = class\_1|\boldsymbol{x})$ could directly be estimated from the distances to the hyperplane using Bayes' rule.

Platt suggested to estimate the posterior by mapping the distances to a sigmoid function [Pla99b]:

$$P(y = class\_1|\boldsymbol{x}) \approx \frac{1}{1 + exp(A\left(\sum_{i=1}^{M} y_i \alpha_i K\left(\boldsymbol{s}_i, \boldsymbol{x}\right) + b\right) + B)} \qquad (4.4.11)$$

The parameter A and B of this sigmoid function are calculated from the training data by solving a maximum likelihood function. I used an implementation by Lin et al [LW93].

Interestingly, the local accuracy estimation [HZET07]

$$r(\boldsymbol{x}) = 1 - \frac{1}{M_v} \sum_{i=1}^{M_v} \mathcal{L}_{0/1}(\boldsymbol{v}_i, f(\boldsymbol{v_i})) \text{ for all } \{\boldsymbol{v}_i | D(\boldsymbol{v}_i, \boldsymbol{x}) < d_v, \boldsymbol{v}_i \in \mathbb{V}\}. \tag{4.4.12}$$

could be used for the estimation of the classification certainty as well (see section 4.4.2 for details).

For a multi-class classification problem, the classification probabilities can be estimated as proposed by Huang et al [HWL06] based on error correcting output codes [DB95].

Chapter 5

Applications

In this thesis, I developed a workflow for classifying genomic profiles and applied it to various data sets. Please note that the workflow presented in chapter 4 evolved over time. Subsequently, some data sets were analysed with preliminary methods.

The main parts of this chapter have previously been published. My main contribution to the following projects concerned the preprocessing of the data sets, the visualisation, classification and a basic statistical analysis. However, I will focus here on the classification tasks and unsupervised machine learning methods (like clustering, principal component analysis and self-organising maps).

Table 5.1 provides an overview about the dimensionality (number of features), size (number of cases), type of genomic profile (matrix-CGH, LOH or classical CGH) and the chosen classification method of all data sets.

Three of the following studies analyse single disseminated tumour cells (projects 1, 5 and 6). These cells can be detected in bone marrow of cancer patients by staining with the anti-cytokeratin antibody. Approx. 1-2 cytokeratin-positive cells can be found in 1 million bone marrow cells of cancer patients without known metastases. Furthermore, the presence of cytokeratin-positive cells has a prognostic impact on the survival of these patients [BPM$^+$00, Kle03].

Data set	Dimen-sionality	Size	Profile-type	Classification method
Early metastasis (project 1)	60	32-42	CGH	SVM
Ductal and lobular breast cancer (project 2)	6000	40	matrix-CGH	SVM
Liposarcoma (projects 3 and 4)	228	16	matrix-CGH	SVM
Early breast cancer (project 5)	2464	21	matrix-CGH	SVM
LOH analysis of single cells (project 6)	48	86	LOH	Decision tree

Table 5.1: Short description of the classification tasks.

5.1 Classical CGH

5.1.1 Early metastasis in HER-2 transgenic mice

The description of the first analysis is based on [HGS+08] and unpublished results. The CGH-analysis of single cells and microdissected tumour samples was done by Yves Hüsemann and Jochen Geigl under the supervision of Christoph Klein, University of Munich. My main contributions were a classification analysis, the calculation of the tumour growth and the analysis of the similarity/dissimilarity between primary tumours and distant tumour cells.

Introduction

The tumour size of most human cancers, including breast cancer, correlates positively with the development of metastases. The prevailing doctrine is that tumour cells capable of dissemination and metastasis generally evolve late during tumour growth. However, metastases also develop in patients with small cancers or even in the absence of detectable primary tumours (so-called ”cancer of unknown primary”).

To determine when carcinoma cells disseminate and how metastases arise,

we followed cancer progression from earliest detectable epithelial alterations to metastasis in a mouse-model (HER-2 transgenic mice). Such a mouse mimics progression and gene expression profiles of human breast cancer. Female mice, which are hemizygous for the constitutively activated rat HER-2 gene, start to express the oncogene at the onset of puberty (weeks 3-4 of age) when the mice become responsive to steroid sex hormones. At week 7, morphological changes in the breast, comparable to an atypical hyperplasia, become regularly visible. Between weeks 15-18, the mice develop in situ carcinomas and between week 22-30 all breast glands are transformed to invasive cancers. The mice are euthanised between weeks 27 and 33 when primary tumours exceed 1.5 cm in diameter and at about the same time metastasis to the lung becomes macroscopically detectable.

The HER-2 negative siblings of these mice (wild type mice) remained tumour free.

Methods

Classification analysis of clonal relationship

I applied the support vector machine (SVM) implementation libsvm (`www.csie.ntu.edu.tw/~cjlin/libsvm`) as the classifier in a leave-one-out cross-validation (LOO CV) design with a grid parameter search within the LOO CV. A weighted SVM for unbalanced data and the RBF kernel were used. The confidence intervals for the classification accuracy values were calculated as published [Mit97]. My conclusions were confirmed using a balanced LOO CV design and the classifier C5.0 in clementine (`http://www.spss.com/clementine`). The ranking of the chromosomal regions (regarding their importance for the classification) was calculated from the trained classifier according to the absolute value of each component of the hyperplane direction vector [GWBV02]. Classification probabilities were calculated according to Platt [Pla99b] in an implementation of libsvm [CL01].

Statistical and bioinformatical analysis of clonal relationship

The similarity between aberrations of two encoded (amplification +1, deletion -1) samples was obtained using the Manhattan distance. Briefly, the similarity is the sum of all differences in aberrations at which a difference between a deletion and amplification counts two and the difference between a balanced area and an amplification or deletion counts one. The p-value was calculated using an exact, paired, one-sided Wilcoxon signed rank test.

Clustering

I applied the clustering of proteases using Euclidean distances, complete linkage for proteases and average linkage for cases. The stability of the cluster was assessed using BRB ArrayTools developed by Dr. Richard Simon and Amy Peng [MRF+02].

Calculation of tumour progression over time

I calculated the tumour areas and tumour volumes from 407 mammary glands of 41 mice assuming the shape of an ellipse/circle and ellipsoid/sphere respectively for each tumour. The tumour size/area of a mammary gland without a tumour was set to zero. The curve was fitted using Friedman's scatterplot smoother [Fri84]. HER-2 positive cells from 31 samples (28 mice) and CK positive cells from 33 samples (31 mice) were calculated as the sum of single disseminated cells and the number of aggregates. Measurements from similar time points (+/- 1 week) were consolidated. An offset of +/-0.3 weeks was used to draw HER-2 and CK positive cells in one plot. The number of positive CK and HER-2 cells from non-transgenic mice were measured in 25 mice (CK) and 24 mice (HER-2) at five time points (weeks 4, 9, 18, 22, 29) and connected by a dotted line. The area of lung metastases was measured using the PALM Robo V1.2.3 software and calculated as the sum of all metastases found. Values were averaged over two tissue sections. Four operated mice and 14 non-operated mice were measured. The curves were fitted using Friedman's scatterplot smoother. All analyses were performed using the statistical language R (www.r-project.org).

Microarray analysis

Contamination by stromal cells was excluded by laser microdissection. After global mRNA amplification, PCR amplified cDNA fragments were digoxigenin labelled and non-radioactive hybridised to nylon filters. The cDNA array comprised 41 molecules belonging mostly to the families of matrix-metalloproteases (MMPs), cathepsins and their inhibitors. Significance analysis of microarrays (SAM) was performed as published [TTC01].

Detection of micrometastases and single disseminated cells

Micrometastases to the lung were detected using an antibody against the HER-2 transgene. Normal lung tissue did not express the antigen. For detection of tumour cells disseminated to the bone marrow, antibodies against the HER-2 transgene and an antibody against epithelial cytokeratin (CK) were used. Both antibodies showed negligible background positivity in the bone marrow of non-transgenic mice with the cytokeratin antibody being

more specific.

CGH analysis

Laser microdissection of several areas of the primary tumour was performed. Metastases from the lungs were also microdissected whereas the disseminated tumour cells isolated from the bone marrow were analysed as individual cells. The CGH profiles obtained were translated into a table, after dividing each chromosome into three parts (A-C).

Results and discussion

Single disseminated cancer cells could already be detected in bone marrow and lung tissue in week 9 - a time point at which tissue morphology is comparable to atypical hyperplasia. The cancerous origin of these single disseminated cancer cells was confirmed by analysing their (aberrated) genomic profiles using CGH. The kinetics of tumour cell dissemination to bone marrow (BM) was found to be constant until week 27 despite an enormous increase of tumour cells at the primary sites. The same phenomenon was observed for lung micrometastases (Fig. 5.2 and 5.3). Initiation of metastasis apparently occurred very early in tumourigenesis and became less likely as the tumours grew.

We compared the genetic aberrations of the primary tumours to those of the disseminated cells and lung metastases (Fig. 5.1). While the classical progression model would predict that tumour cells accumulate chromosomal abnormalities at the primary site and display them also at the distant site, early dissemination and parallel outgrowth would result in vast divergence. Thus, the genomic comparison determines whether the disseminated cells are similar to the predominant clone within a primary tumour. We analysed all primary tumours of two animals in week 27 (PT; n=20), their lung metastases (LM; n=8) and disseminated cancer cells isolated from the bone marrow (BMDT; n=10) by CGH.

We asked whether or not the observed patterns of CGH abnormalities in LM and BMDT were more closely related to the primary tumour than the 10 different primary tumours of a mouse to each other. On average, the genetic distance between primary tumours and BMDT or LMs was similar to the distance between the polyclonal primary tumours (for mouse 91 between 10-13 divergent aberrations and for mouse 102 between 6-8). Therefore, we could not find support for the hypothesis that the BMDTs or LMs are derived from the predominant clone of the primary tumour.

To check, whether genomic profiles of tumour cells from different organs show variations, we designed a classifier trained to distinguish between primary tumours (PTs; n=26), LMs (n=16) and BMDTs (n=16) from various animals. PTs and BMDTs or LMs could be separated with identical, high accuracy (80%; the 95% confidence interval being 60-90%; Fig. 5.4 and 5.5). By ranking the chromosomal regions that were most informative for the classification result, we could determine a panel of chromosomal abnormalities characteristic of tumour cells growing at different anatomical sites. Primary tumours and lung metastases were differentiated by a loss of chromosome XB, 15C and 4C in the primary tumour, while chromosome 15C is mostly gained in lung metastasis. BMDTs less frequently displayed alterations on chromosome 4, X and 17 than primary tumours. LMs and BMDTs could hardly be differentiated (classification accuracy approx. 50%; Fig. 5.6).

Furthermore, we compared the expression of invasion-associated proteases in atypical hyperplasia at week 9 and large carcinomas at week 27. We found a significantly higher expression of cathepsins (Ctsz, Ctsb, Ctsf, Ctsl, Cstb,Ctsd, Ctsh), members of the MMP system (Mmp2, Mmp14, Mmp11, Timp3) and two caspases (Casp2 and Casp9) at the early stage of tumour development (false discovery rate q = 4.7% for week 9 compared to week $\geq$27). Of these, Ctsz (p-Value 0.0046), Ctsd (0.0046), Mmp2 (0.009), Ctsh (0.0132), Ctsb (0.0215) were highly significant applying a one-sided Wilcoxon test (corrected after Hochberg). The two groups (week 9 vs. $\geq$ 27) could be visually separated by cluster analysis of the gene expression patterns (cluster robustness index = 0.901; Fig. 5.7) with the exception of few outliers. A principal component analysis (PCA) also showed that the two groups (week 9 vs. $\geq$ 27) could be separated on the basis of a gene expression analysis limited to members of the proteolytic system (Fig. 5.8).

Together, we found metastatic cells in lung and bone marrow at a time when mammary glands displayed histologically atypical hyperplasia only and no invasive carcinoma. At that stage, electron microscopy revealed microinvasion of the basement of stem-like cells. Disseminated cells in bone marrow expressed distinct stem/progenitor cell markers and eighty of these cells - injected into wild-type siblings sufficed to induce massive carcinosis of the recipients' bone marrow. Since neither the origin nor the tumourigenicity of the early disseminated cancer cells can be easily dismissed, the concept of metastasis as being a late event in oncogenesis may be in need of reconsideration.

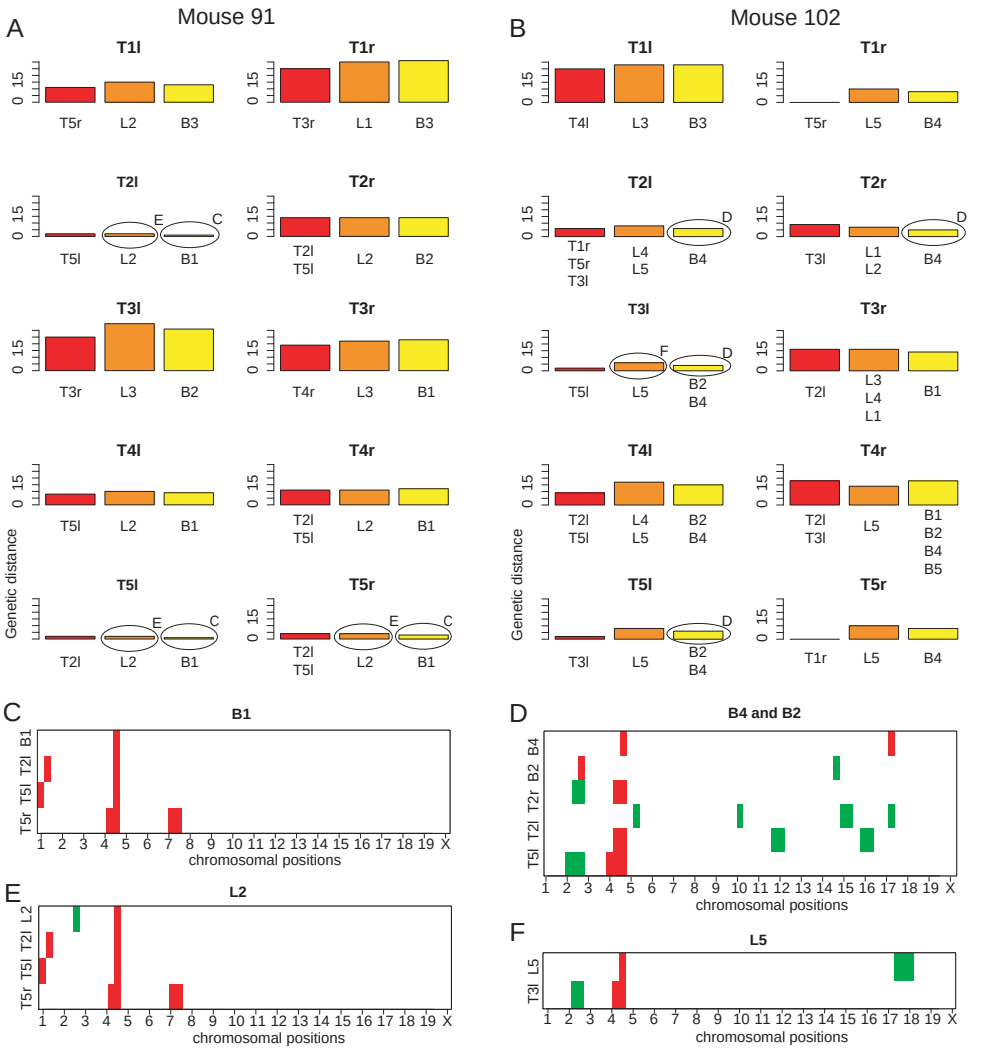


Figure 5.1: Genomic relationship between primary tumours (T) and disseminated tumour cells from bone marrow (B) and lung metastases (L). (A and B) For each primary tumour of mouse 91 (A) and 102 (B) the genetic distance to the closest related T, L and B is shown. (C-F) The genomic profile of all B and L displaying a genetic distance of less than 7 from a primary tumour are shown in detail, together with the respective primary tumours (the selected samples are indicated by ellipsoids in panel A and B). The genetic distance was calculated as the Manhattan distance of the CGH profile values which were encoded as -1 for a deleted region, 0 as balanced region and +1 for an amplified region.

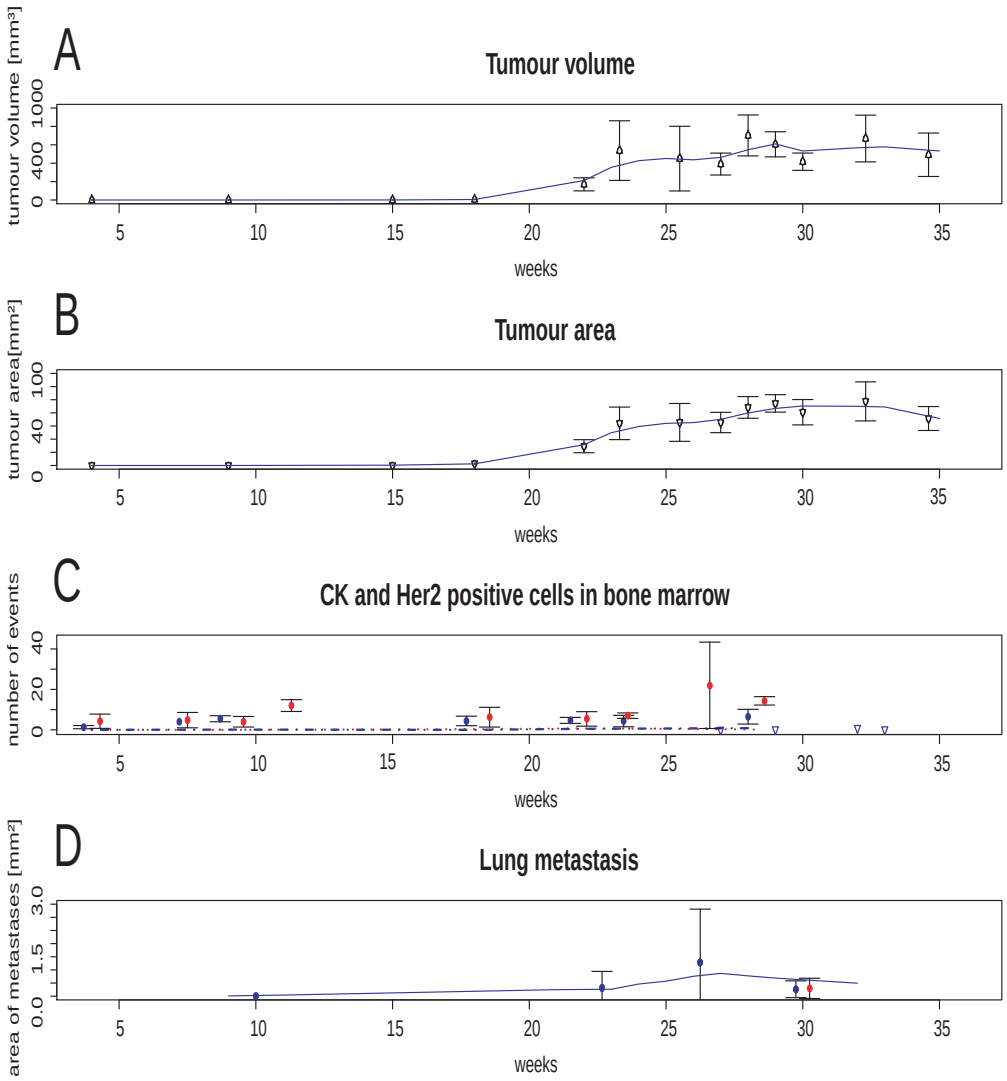


Figure 5.2: Progression of local and systemic disease over time. A/B) Increase of tumour volume and tumour area (triangles indicate mean values, whiskers 95% confidence interval with solid line between triangle indicating best fitted curve). C) Number of cytokeratin+ (red dots) and HER-2+ (blue dots) cells in bone marrow (whiskers indicate 95% confidence interval, dashed red or blue line depicts results from non-transgenic control mice for cytokeratin or HER-2, respectively). D) Average area of lung metastases (note logarithmic scale) in an individual mouse (blue indicates average size of non-operated mice at various time points, red the average size of operated animals at 10-15 weeks after surgery, whiskers indicate 95% confidence interval).

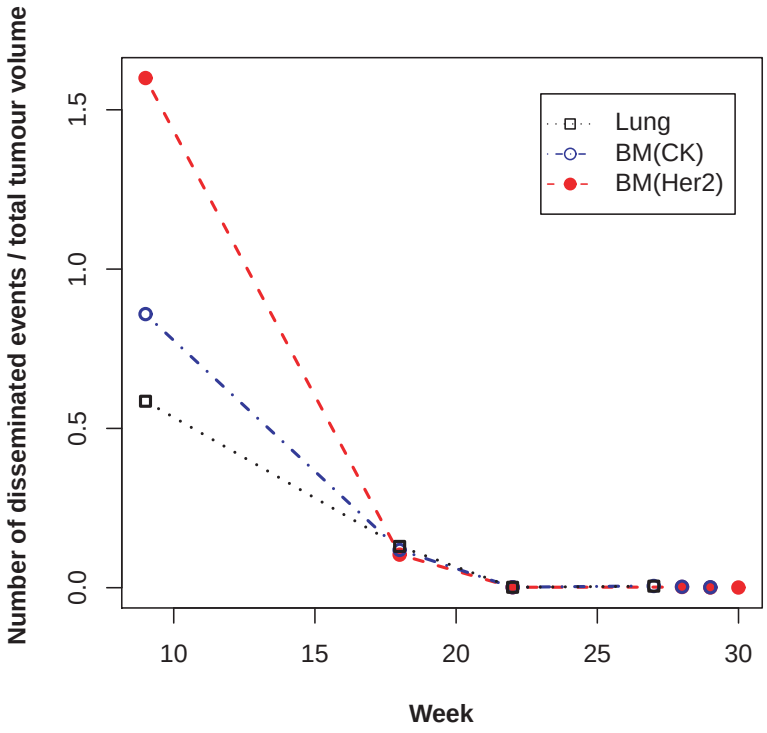


Figure 5.3: Primary tumour growth and tumour cell dissemination to lung and bone marrow. Cancer cell dissemination to lung and bone marrow (BM) in the course of time. The number of dissemination events (single cells plus clusters) was divided by the total tumour volume in mm^3. Cancer cells were identified by anti-HER-2 staining in lung sections, and by anti-HER-2 and anti-CK staining in bone marrow.

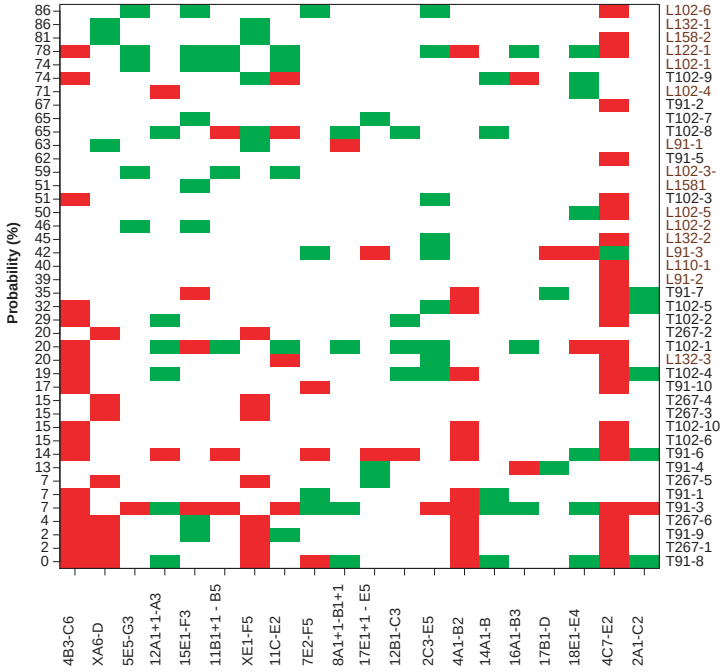


Figure 5.4: Results from the classifier trained to distinguish between primary tumours (T) and lung metastases (L). The samples (rows) were sorted according to probability to be classified as "lung metastasis". Chromosomal regions (columns) are ranked according to the information they provide to the classification result (green, chromosomal gain; red, chromosomal loss) and the best 20 chromosomal regions are shown only.

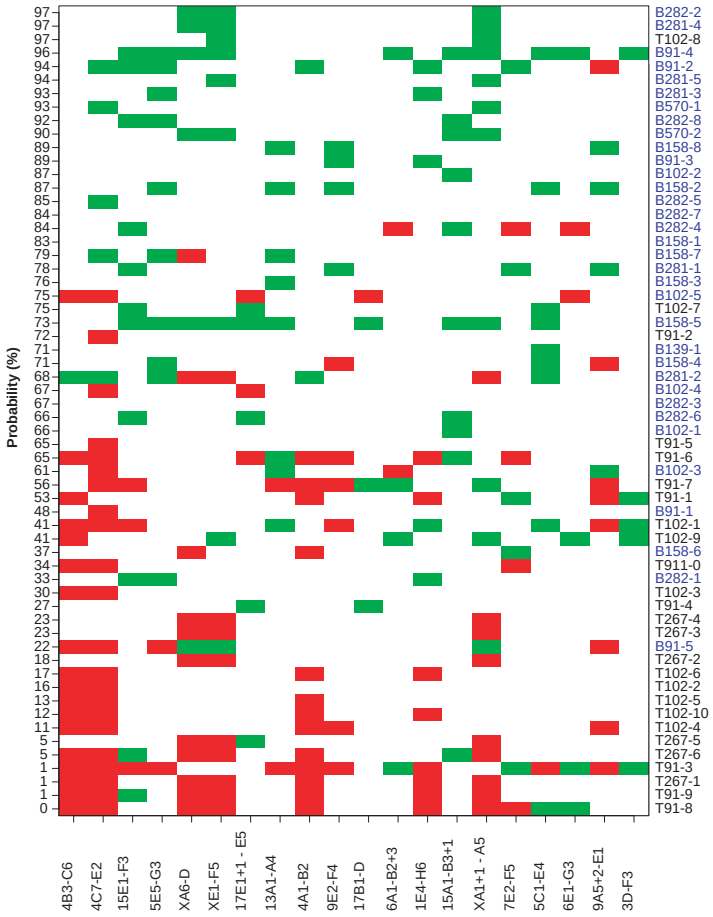


Figure 5.5: Results from the classifier trained to distinguish between primary tumours (T) and bone marrow derived tumour cells (B). The samples (rows) were sorted according to probability to be classified as "bone marrow derived cell". Chromosomal regions (columns) are ranked according to the information they provide to the classification result (green, chromosomal gain; red, chromosomal loss) and the best 20 chromosomal regions are shown only.

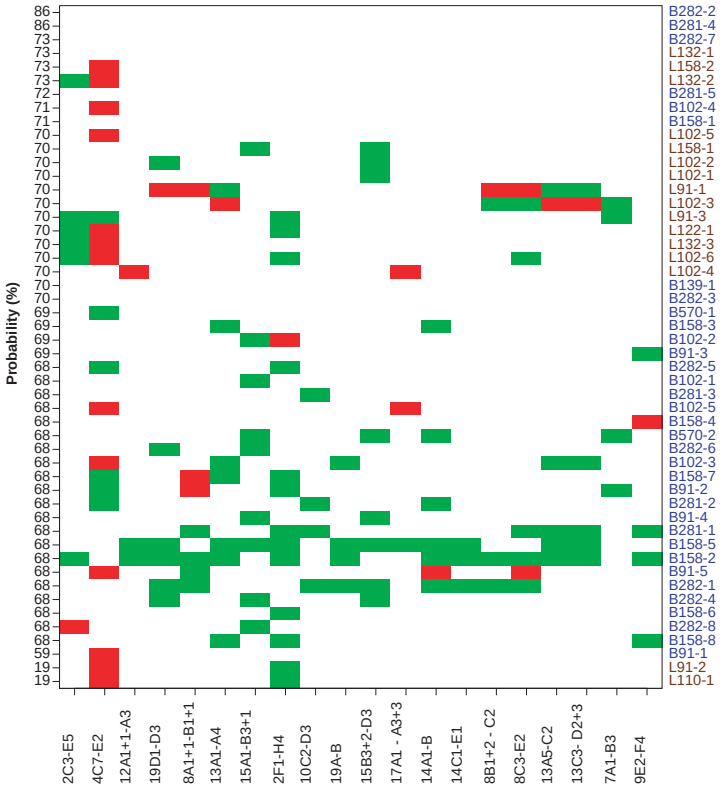


Figure 5.6: Results from the classifier trained to distinguish between bone marrow derived tumour cells (B) and lung metastases (L). The samples (rows) were sorted according to probability to be classified as "lung metastasis". Chromosomal regions (columns) are ranked according to the information they provide to the classification result (green, chromosomal gain; red, chromosomal loss) and the best 20 chromosomal regions are shown only.

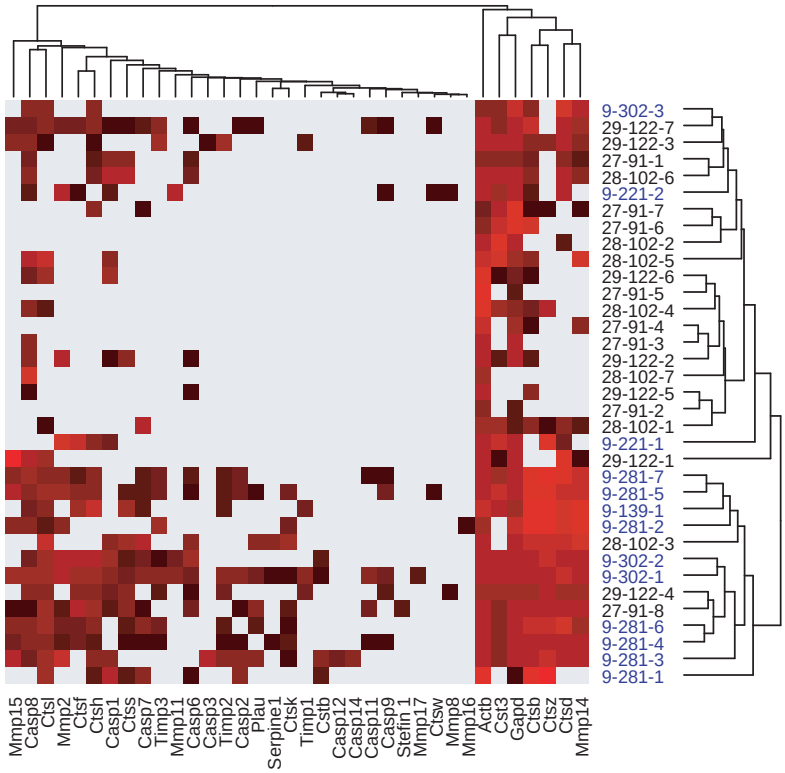


Figure 5.7: Activation of the proteolytic system during atypical hyperplasia. Cluster analysis of gene expression data. Samples from young transgenic mice (rows) could be separated on the basis of expression of molecules of the proteolytic system (columns). Sample identifiers consist of age in weeks, the number of the mouse from which the sample was taken and the sample number; individual samples from one animal were isolated from different mammary glands.

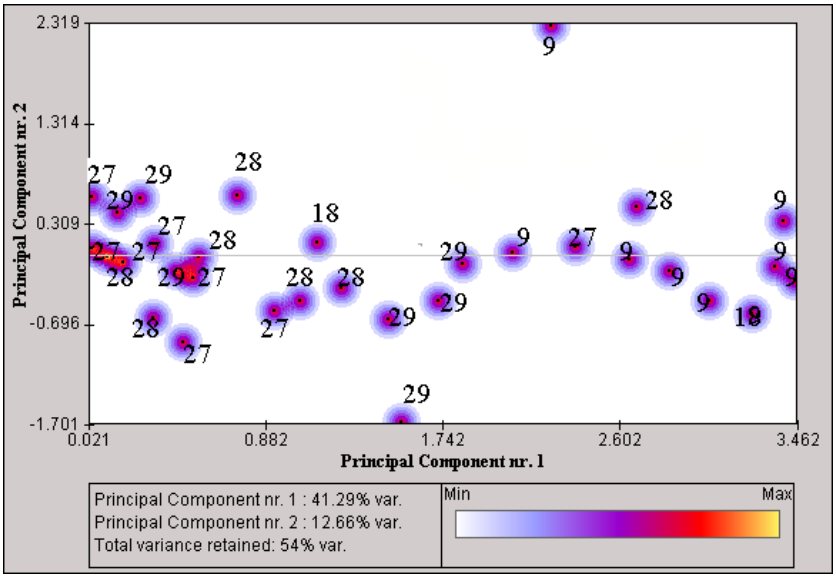


Figure 5.8: Principal component analysis of gene expression data. Samples from young animals (week 9, here shown as "9") and old animals (week 27,28, 29, shown as "27', "28", "29") could be separated on the basis of expression of molecules of the proteolytic system. Note that only two samples from week 27 and 28 are found between samples from week 9.

5.2 Matrix-CGH

5.2.1 Genomic profiling of ductal and lobular breast cancer

This matrix-CGH study was published in [SRS+06]. The profiling was done by Daniel Stange under the supervision of Bernhard Radlwimmer at the laboratory of Prof. Peter Lichter, DKFZ Heidelberg. I analysed the difference between ductal and lobular profiles as well as correlations with other histopathological variables by means of machine learning and classical statistics.

Introduction

Invasive ductal carcinoma (IDC) and invasive lobular carcinoma (ILC) represent the major histological subtypes of invasive breast cancer. They differ with regard to presentation, metastatic spread and epidemiological features. However, the molecular basis for differences in phenotype and clinical behaviour between ILC and ICD are not yet understood. To elucidate the genetic basis of these differences, we analysed copy number imbalances that differentiate the histological subtypes.

Methods

Genomic profiling

High-resolution genomic profiling of 40 invasive breast cancers using matrix-comparative genomic hybridisation with an average resolution of 0.5 Mega base pairs was conducted on bacterial artificial chromosome microarrays.

Preprocessing

Fluorescence intensities of all spots were filtered (intensity/local background >3; mean/median intensity <1.3; standard deviation of genomic fragment log ratios <0.25) and normalised block-wise according to Loess [BIAS03]. Chromosomal breakpoints delimiting regions were then detected by GLAD (Gain and Loss Analysis of DNA), a method developed by Hupé et al. [HST+04] based on the Adaptive Weight Smoothing (AWS) procedure. The parameters of GLAD were adjusted through several hybridisations of normal-proband-DNA against pool-DNA from five normal probands as negative and cell line experiments with well known genomic aberrations as positive controls. The

thresholds differentiating balanced and altered regions were 1.12 for gains and 0.88 for losses and were set in such a way that no false positive region was found in the control hybridisations.

Significance Analysis of Microarrays (SAM)

Significance Analysis of Microarrays (SAM) was performed for discretised copy number ratios using an implementation of SAM for categorical variables (kindly provided by Holger Schwender) with a false discovery rate of $< 5\%$. This procedure differs from the original SAM in the use of Chi-squared instead of t-test statistics [Sch04]. All copy number values were discretised and encoded as -1 for a deletion (ratio <0.88), 0 for a balanced genomic fragment and +1 for a gain (ratio >1.12). SAM was performed on all genomic fragments which showed aberrations in at least 20% of all cases. This preselection of genomic fragments was done independently from the class label and a simulation showed that the number of false positive genomic fragments is still controlled.

Machine learning analysis

The support vector machine (SVM) implementation libsvm was used as the classifier in a leave-one-out cross-validation (LOO CV) design with a grid parameter search within the LOO CV. The SVM (RBF-kernel) was used on discretised copy number ratios. The threshold differentiating balanced from aberrated genomic fragments was automatically chosen for each run inside the cross-validation (by maximally selected statistics) to avoid overfitting. Inside the cross-validation genomic fragments were used that showed aberrations for at least 10% of all cases. The most important separating chromosomal regions were calculated from the trained SVM classifier according to the absolute value of each component of the hyperplane direction vector [GWBV02].

Unsupervised hierarchical clustering and classical statistics

The clustering of all ILC and IDC was done for the Manhattan distance and Ward's linkage with discretised copy-number ratios. Associations of histopathological parameters and number of aberrations with the clustering were tested by Fisher's exact test and Kruskal-Wallis test. An association of each clustering subgroup with histopathological parameters and number of aberrations was tested by using an exact Wilcoxon signed rank test (corrected by Hochberg). All analyses were done in the open source statistical environment R, version 1.91 (`http://www.r-project.org`).

Results and discussion

To identify regions that were important for the discrimination of IDC and ILC by independent and rigorous biostatistical methods, I did both an SVM and a SAM analysis (Fig. 5.9).

The 128 top-ranked fragments selected by SVM map to chromosomal regions on 1q and 16p, identifying them as the most significant discriminators of IDC and ILC. An optimal classification accuracy of 65% was achieved using a classifier consisting of 733 genomic fragments.

An implementation of SAM, specifically adapted to the analysis of DNA copy number data, was used to identify the fragments that were imbalanced with significantly different frequencies in the histological subtypes. 116 genomic fragments were identified that cluster in two regions, one on chromosome 1q24.2-q31.3 and the other on 16p11.2.

These regions were further narrowed down to subregions 1q24.2-25.1, 1q25.3-q31.3, and 16p11.2. Located within the candidate gains on 1q are two genes, FMO2 and PTGS2, known to be overexpressed in ILC relative to invasive ductal carcinoma. Assessment of four candidate genes on 16p11.2 by real-time quantitative PCR revealed significant overexpression of FUS and ITGAX in ILC with 16p copy number gain.

Unsupervised hierarchical cluster analysis identified three molecular subgroups that are characterised by different aberration patterns, in particular concerning gain of MYC (8q24) and the identified candidate regions on 1q24.2-25.1, 1q25.3-q31.3, and 16p11.2. These genetic subgroups differed with regard to histology, tumour grading, frequency of alterations and oestrogen receptor expression.

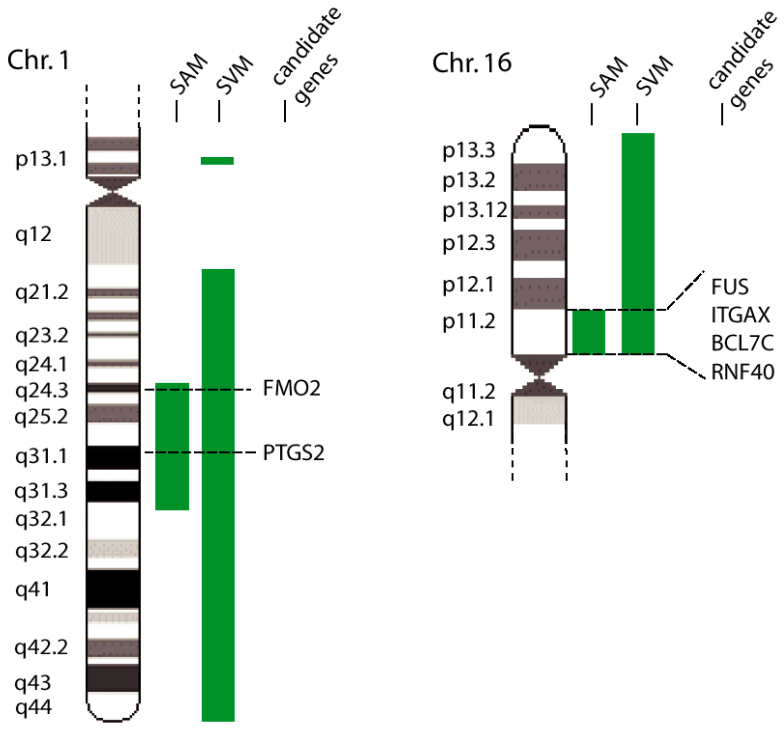


Figure 5.9: Discriminating regions on 1q and 16p identified by different analysis methods. The different regions identified by SVM and SAM were compared with matrix-CGH profiles and plotted against the ideograms (schematic chromosomal map) of chromosomes 1 and 16. The localisation of candidate genes is indicated. Figure taken from [SRS+06].

5.2.2 Genomic profiling of dedifferentiated and pleomorphic liposarcoma

This section is based on [FSW+02]. The genomic profiling was performed by Björn Fritz at the laboratory of Peter Lichter. My task was the statistical and bioinformatorical analysis of the genomic profiles.

Introduction

Liposarcomas represent the most common soft tissue tumours in adults and account for 20% of all mesenchymal malignancies. The tumours are characterised by a high morphological diversity. Five morphological liposarcoma subtypes can be distinguished:

- pleomorphic liposarcoma (PL)
- dedifferentiated liposarcoma (DL)
- myxoid liposarcoma
- round cell liposarcomas and
- well differentiated liposarcomas.

Sixteen dedifferentiated and pleomorphic liposarcomas were analysed by comparative genomic hybridisation (CGH) to genomic microarrays (matrix-CGH). The low number of samples resulted from the fact that liposarcomas are rare tumours and the developmental status of the matrix-CGH method at the time of the hybridisations (2000).

Methods

Genomic profiling

The genomic profiles of this study were based on 228 genomic fragments arrayed on a matrix CGH-chip. The matrix-CGH chip was built from a genome-wide array and additional genomic fragments from the region of interest. 116 genomic fragments were approximately equidistantly arrayed whereas 112 additional genomic regions were located in a region of interest (chromosome 12q).

Cluster analysis

I applied the clustering algorithm Two-Step in an implementation of Clementine (version 4) with a predefined number of four clusters. The principal component analysis was done using J-Express 2.01d.

Machine learning

Predictive genes were revealed using an implementation of the decision tree algorithm C5.0 in Clementine and the support vector machine SVM light. A support vector machine was used for tumour type assignment of the two initially unclassified samples PL39 and DL48. The class assignment probabilities of these unclassified cases were calculated from the distance to the hyperplane using Bayes's rule.

Results and discussion

Matrix-CGH revealed copy number gains of numerous oncogenes, i.e., CCND1, MDM2, GLI, CDK4, MYB, ESR1 and AIB1, several of which correlate to a high level of transcripts from the respective gene. Self-organising maps ([Koh95b]) and a principal component analysis uncovered a clear separation of both tumour subtypes (Fig. 5.11 and Fig. 5.12).

A classification analysis using the decision tree algorithm C5.0 and an SVM also showed a separation of both classes (leave-one-out accuracy 100%). Moreover, the prediction of the tumour type of two initially unclassified samples revealed the experimentally validated class assignments and a low error probability of $\leq 5\%$.

Copy number changes of oncogenes and tumour suppressor genes in the eight DL and eight PL analysed are indicated in Fig. 5.10. High-level amplifications (intensity ratio 2) were detected for CCND1, CCND2, MYB, MDM2, GLI and CDK4. The highest ratio values were observed for MDM2, GLI and CDK4 localised on chromosomal subregion 12q13–q15 and were present in all DL tumours.

The best discriminators between pleomorphic and dedifferentiated were derived from DNA fragments mapping on chromosome 12q including CDK4, MDM2, and GLI and RP11-1143G9. Amplification of these clones together with the characteristic amplicon structure found in DL not only allows a clear discrimination from PL but strongly argues for different pathogenic mechanisms leading to these liposarcoma subtypes.

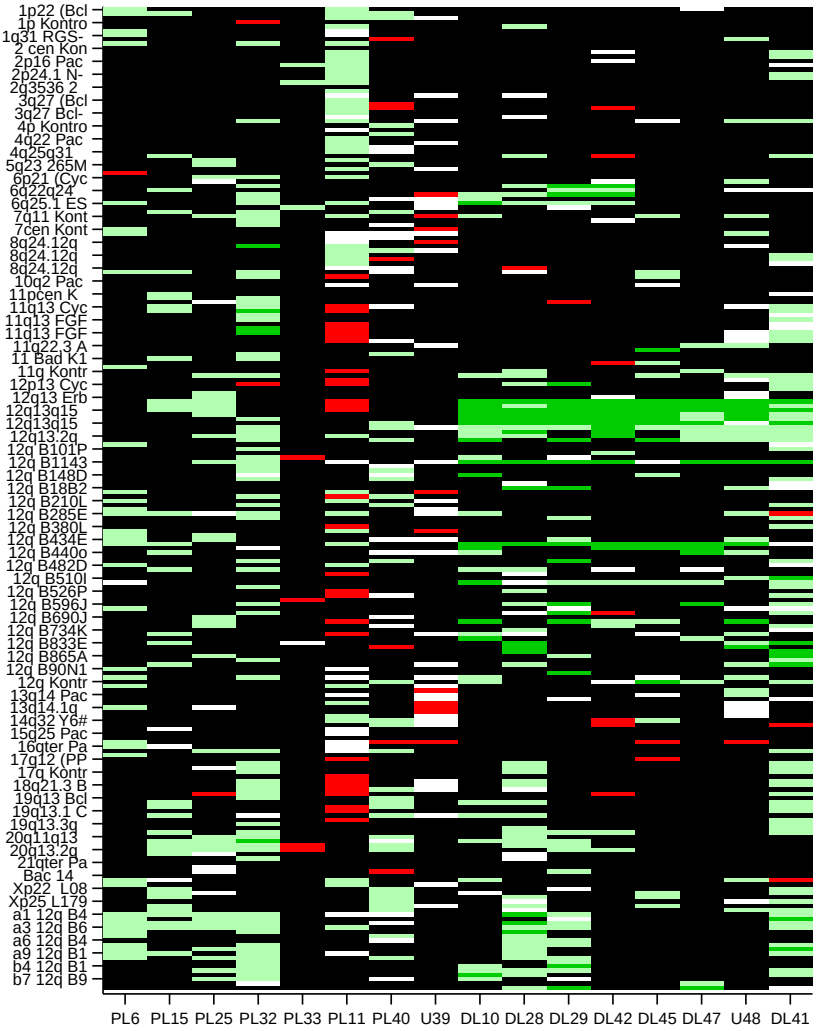


Figure 5.10: Liposarcoma dataset. Each column denotes a case and each row a genomic fragment. Dark green, high-level amplification ($\log_2$ 1); light green, low-level amplification ($\log_2$ 0.32); red, deletion ($\log_2$ 0.41); black, balanced ($\log_2$ 0); white, unknown.

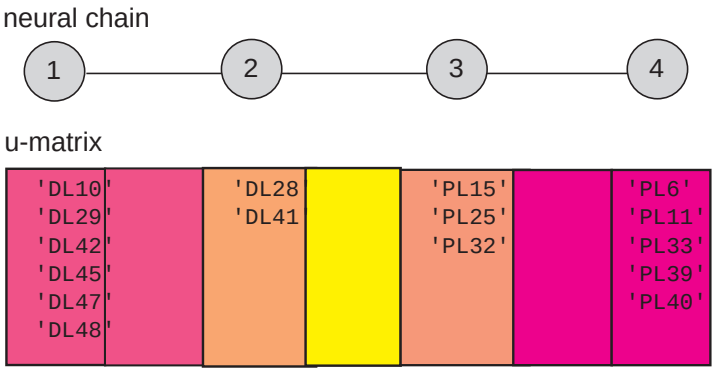


Figure 5.11: Self-organising map of the liposarcoma dataset. U-matrix representation of the tumour samples in a self-organizing map with four neurons. Samples from the dedifferentiated liposarcoma (DL) were assigned to neurons 1 and 2 and the samples from the pleomorphic liposarcoma (PL) to neurons 3 and 4. The colours (or grey values) visualise the distance between the neurons in input space: magenta (dark gray) corresponds to the smallest and yellow (light gray) to the largest distance.

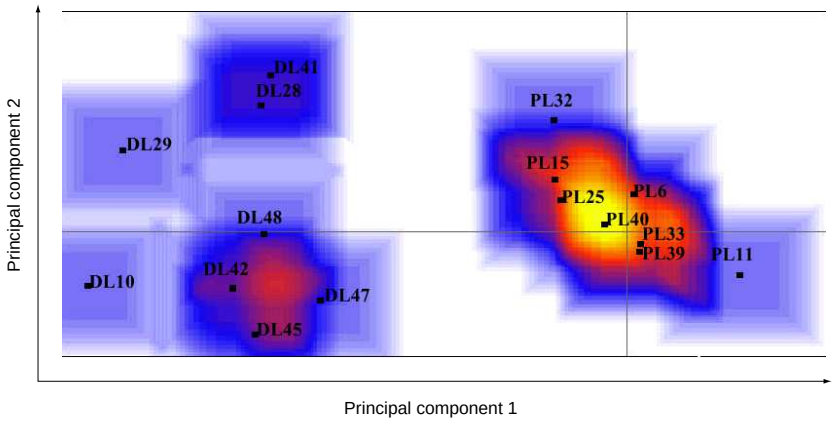


Figure 5.12: Unsupervised analysis of eight PL and eight DL cases using matrix-CGH data and the principal component algorithm (see text). DL and PL tumours are clearly separated in the first main component as indicated by their different plane position.

5.2.3 Understanding the classication of liposarcoma tumours with a support vector machine

Introduction

Here, I present an application of the case-based analysis of classification results (section 4.4.1) on the liposarcoma data set introduced before. This data set was exemplarily chosen due to the separability of both classes (100% classification accuracy).

The underlying data set was kindly provided by Björn Fritz. An earlier version of this section was part of a publication at the German Conference on Bioinformatics 2003 [SMF+03]. Parts of the calculations were performed by Jasmin Müller in her diploma thesis which I supervised.

Methods

204 of the 226 DNA fragments with validated measurements for at least 13 tumour probes were included. SVM-LIGHT was used as an implementation of a support vector machine [Joa99]. The SVM was trained on 15 cases while the remaining case was classified and analysed for explanatory features (genomic fragments). Due to the limited amount of tumour probes, I applied an SVM with a linear kernel.

The case-based analysis framework is explained in section 4.4.1.

Results and discussion

The case-based feature ranking revealed the importance of genomic fragments on chromosome 12q for the discrimination between pleomorphic and dedifferentiated liposarcoma (Table 5.2). Using the replacement strategy, the number of genomic fragments (out of 204) needed to explain the classification outcome for each tumour probe was estimated at between one and five .

As an example, case DL48 (dedifferentiated liposarcoma) is classified as dedifferentiated due to the DNA fragments 12q B438N16 and 12q B1143G9 (Fig. 5.13). The normalised copy number ratio for 12q B438N16 is 3.04 for case DL48, 0.11 for the average of all pleomorphic liposarcomas (PL) and 2.87 for the average of the remaining dedifferentiated liposarcomas (DL). The normalised copy number ratio for 12q B1143G9 is 2.55 for case DL48, 0.19

for the average of all pleomorphic liposarcomas (PL) and 2.62 for the average of the remaining dedifferentiated liposarcomas (DL).

A modification of these two feature values for the dedifferentiated tumour probe DL 48 to the mean values of pleomorphic tumour probes lead to the prediction pleomorphic. In detail, a support vector machine predicts the label dedifferentiated based on the feature vector of case DL48. Next, the feature vector of case DL 48 is modified such that the feature 12q B438N16 becomes 0.11 (average of all pleomorphic liposarcomas) and the feature 12q B1143G9 becomes 0.19 (average of all pleomorphic liposarcomas). All other 202 features remain unchanged. Finally, the support vector machine predicts the label pleomorphic based on the changed feature vector of case DL48. That means the modification of two out of 204 feature values to average values of the other class is sufficient to change the classification outcome.

Likewise, a change of selected feature values for a pleomorphic tumour probe shifted the outcome to dedifferentiated.

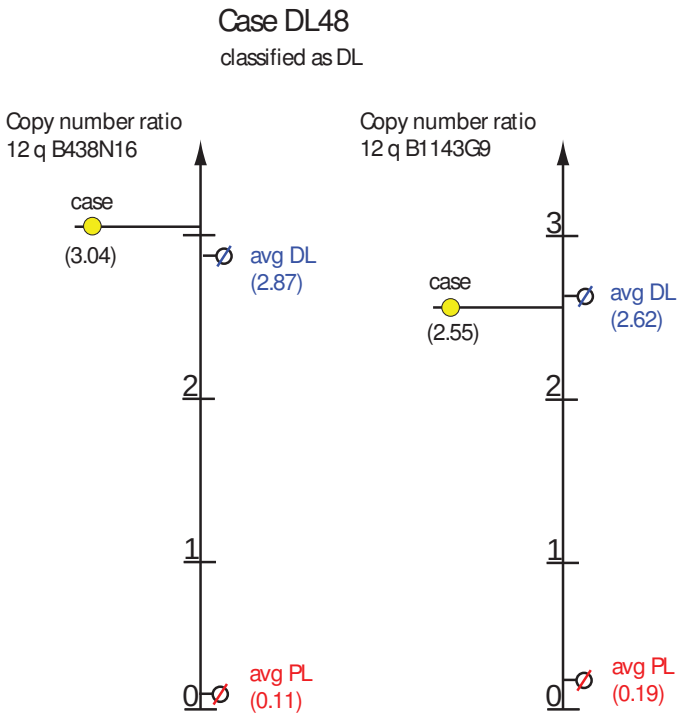


Figure 5.13: Explaining the classification of liposarcomas. Two chromosomal regions were identified for the classification of case DL48: 12q B438N16 and 12q B1143G9. The value of both features from case DL48 is characteristic for the average (avg) value of dedifferentiated liposarcomas (DL).

The results clearly indicate the importance of aberrations on chromosome 12q for the distinction between pleomorphic and dedifferentiated liposarcoma. This is consistent with former results [FSW+02].

Case	Important DNA fragments for the explanation of this case
PL6	12q B438N16; 12q B1143G9
PL11	12q B438N16; 12q13-q15 CDK4; 12q13 MDM2; 12q B1007B5; 12q Control G1749
PL15	12q B438N16; 12q B1143G9
PL25	12q B438N16; 12q B1143G9
PL32	12q B438N16; 12q B1143G9
PL33	12q B438N16; 12q B1143G9
PL40	12q B438N16; 12q B1143G9; 12q13-q15 CDK4
PL39	12q B438N16; 12q13-q15 CDK4
DL10	12q B438N16; 12q B1143G9; 12q13-q15 CDK4
DL28	12q B438N16; 12q B1143G9
DL29	12q B438N16; 12q B1143G9; 12q13-q15 CDK4; 12q13 MDM2
DL41	12q B1143G9
DL42	12q B438N16; 12q B1143G9
DL45	12q B438N16; 12q13 MDM2
DL47	12q B438N16; 12q B1143G9
DL48	12q B438N16; 12q B1143G9

Table 5.2: Features explaining the classification of a tumour probe. The case identifiers refer to [FSW+02].

5.2.4 Genomic profiling of early breast cancer

This section covers an application of matrix-CGH drafted in [WSS+08]. A customised chip from Donna Albertson's laboratory was used and the hybridisations were performed by Rick Segraves. The preparation of the tumour samples and the biological assessment was done by Ute Wölfle. I was responsible for the annotation of the genomic fragments and the statistical and bioinformatorical analysis.

Introduction

Genomic profiles of 21 breast cancer patients were analysed using matrix-CGH to identify genomic aberrations associated with bone marrow (BM) micrometastasis or lymph node (LN) metastasis. A cDNA analysis of a sub set of these tumours previously revealed a specific gene expression signature associated with bone marrow micrometastasis [WCS+03].

Tumours were derived from 16 patients without lymph node (LN) metastases (pN0; LN-) and five patients with nodal metastases (pN1; LN+). Eight of the 16 pN0-patients exhibited tumour cells in their bone marrow (BM+), as revealed by immunostaining with an anti-cytokeratin antibody.

Methods

Hybridisation

The chip, used in this matrix-CGH analysis, contained 2464 genomic fragments, providing an average resolution of 1.4 Mb over the genome. All genomic fragments were spotted in triplicate (3 measurements per genomic region). Female tumour DNA was hybridised with male reference DNA so that on the x-chromosome (numbered as chromosome 23) a DNA gain and on the y-chromosome (numbered as chromosome 24) a DNA loss could clearly be detected and demonstrated the reliability of the array CGH results. For further analyses the values of the y-chromosome were omitted.

Data analysis

I updated the positions of the genomic fragments, originally assigned in August 2001, in April 2005 (UCSC database, www.genome.ucsc.edu). Ratios of genomic profiles for which only one of the triplicate remained after the quality check were excluded from further analysis.

Smoothing

Smoothing and chromosomal breakpoint identification was conducted using GLAD (Gain and Loss Analysis of DNA), a method developed by Hupé et al. [HST+04] based on the Adaptive Weight Smoothing (AWS) procedure. The threshold of the log_2 transformed fluorescence ratios differentiating balanced from altered regions were 0.3 for DNA gains and -0.3 for DNA losses according to previous experience.

Classification analysis

The support vector machine (SVM) implementation libsvm [CL01] was used as classifier in a leave-one-out cross-validation (LOO CV) design with a grid parameter search within the LOO CV.

Statistical and cluster analysis

Significance analysis of Microarrays (SAM) was performed on continuous and discretised copy number ratios. The implementation of SAM for categorical variables [Sch04] used for discretised copy number ratios differs from the original SAM [TTC01] in the use of the Chi-squared instead of t-test statistics. Differences in the number of gains and losses between all subtypes were evaluated using an exact Wilcoxon rank test. The distribution of genomic patterns to hematogenous or lymphogenous dissemination was visualised by hierarchical clustering (Manhattan distance, Ward linkage) including all tumour samples.

Simulation study

A simulation study was performed to check the power of our approach given the low number of cases. High level amplifications (log_2-ratio 0.7) and low level amplifications (log_2-ratio 0.3 and 0.4) were inserted by adding 0.7, 0.3 and 0.4 at randomly chosen loci in 6 of 8 (randomly chosen) BM-positive cases. Amplifications were simulated with a length between two and seven genomic fragments. Missing values were not altered by the simulation. Finally, I could detect distinct high level amplifications (log_2-ratio 0.7, length 4 clones) and large low level amplifications (log_2-ratio 0.4, length 7 clones) with my workflow (smoothing, classification, and statistical analysis). All analyses were performed using the statistical language R (www.r-project.org), version 2.1.

Results and discussion

In total, all 21 breast tumours analysed showed diverse chromosomal aberrations (Fig. 5.14). The mean number of DNA alterations per case was 225 with on average 102 gains and 123 losses. The most frequent DNA gains were observed at 1q32 (in 61% of the samples), whereas the most prominent losses were observed at 16q24 (in 35% of the samples). No statistically significant differences in the array-CGH-patterns between patients with or without BM or LN dissemination could be revealed. Furthermore, the classification accuracy was always below 50%.

The assignments to the tumour class BM- were determined by an analysis of 1 million bone marrow cells, aspirated from both sides of the upper iliac crest. The specificity of this method is below 100 % due to the fact that single disseminated cells are rare. That means that some patients were labeled as BM- even if their bone marrow homed (undetected) single disseminated cells.

Our results indicate that early metastatic spread in breast cancer might not be closely associated with a clear pattern of cytogenetic aberrations in the primary tumour. However, to exclude an underlying genomic pattern in relation to early metastasis the sample size of this study needs to be enlarged and a higher array resolution could be useful. A simulation study revealed that in this setting only distinct high level DNA gains and losses and large single DNA losses / gains would have been found as statistically significant.

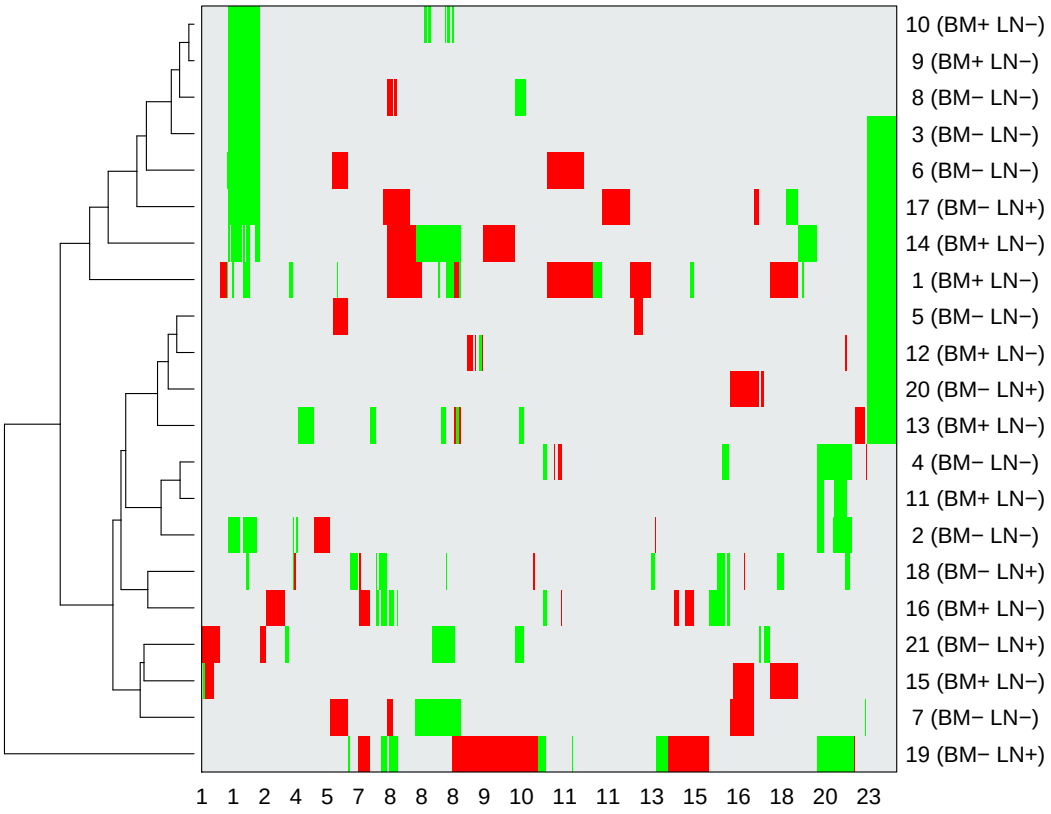


Figure 5.14: Cluster analysis including all breast cancer samples. Each row denotes a case and each column a genomic fragment. A hematogenous (BM+/BM-) or lymphogenous (LK+/LK-) dissemination pattern could not be identified. Green represents DNA gain, red DNA loss and white balanced DNA content. Chromosomal aberrations were classified as DNA gain with a normalised log_2 transformed fluorescence ratio >0.3 and as loss with a ratio <-0.3. The chromosomal positions of the genomic fragments are depicted on x axis.

5.3 Loss of heterozygosity

5.3.1 Genomic analysis of single cytokeratin-positive cells from bone marrow in breast cancer

Here, I describe a study published in [SMH+05]. The LOH analysis was performed in the laboratory of Christoph Klein, University of Munich. I contributed with a machine learning and statistical analysis.

Introduction

Recently, Christoph Klein and his co-workers found that 57% of disseminated cells isolated from the bone marrow of breast cancer patients without clinically evident metastasis do not display chromosomal aberrations (as defined by comparative genomic hybridization)[SKRD+03]. However, the matched primary tumours of the patients regularly harboured multiple chromosomal alterations. Thus, we reasoned that karyotypically normal cytokeratin-positive cells in bone marrow may represent genetically very early stages of breast cancer development — if they are indeed tumour cells.

To prove this hypothesis, we evaluated if cytokeratin-positive cells displayed DNA damage when analysed for LOH at higher resolution using 27 microsatellite and two restriction fragment length polymorphism markers. We analysed single disseminated tumour cells with normal karyotypes for chromosomal aberrations, subchromosomal allelic losses and gene amplifications.

All available disseminated cells from the breast cancer patients were divided into the following groups (M0: without clinically evident metastasis; M1: with metastasis):

- group A: 37 cytokeratin-positive cells, M0 patients, normal CGH profile
- group B: 15 cytokeratin-positive cells, M0 patients, aberrated CGH profile
- group C: 45 cytokeratin-positive cells, M1 patients, aberrated CGH profile

and compared to the following control cells

- control group 1: 21 blood cells from healthy controls

- control group 2: 52 cytokeratin-negative cells from bone-marrow of patients with malignant epithelial cancer.

Methods

Statistical analysis

I compared the rate of LOH of all subgroups using a two-sided exact Wilcoxon signed rank test in the software package R. To exclude a patient effect, a simulation study by randomly choosing one cell per patient was performed. All markers were compared using Fisher's exact test. Noninformative cells and homozygous losses were treated as missing values. The false discovery rate (FDR)-controlling procedure by Benjamini and Hochberg was used to account for multiple testing. Adjacent markers were combined in a sliding window to enhance the reliability of the result. All measurements of three adjacent markers were merged and compared by Fisher's exact test.

Machine learning

The decision tree algorithm C5.0 was applied as classifier using a 10-fold crossvalidation (CV) and a balanced design. A decision tree was used to deal with the fact that approx. 50 % of all measurements were non-informative and therefore missing. Imputation as a prerequisite for the use of an SVM would have introduced a large bias. The classification accuracy was estimated from 10 runs (10×10 CV). Noninformative markers and homozygous losses were encoded as missing values. All calculations were performed with the software package Clementine (http://www.spss.com/clementine/), version 8.5. The standard errors for the accuracy values and the classification probabilities were calculated as described by SPSS. For the control group 1, group C and group B cells, the specificity was estimated from class assignments with a classification accuracy >60%.

Results

We found that cells from control group 1 displayed significantly fewer allelic losses than control group 2 (2.9% [95% confidence interval (CI), 0.1%–5.7%] versus 9.8% [95% CI, 6.8%–12.8%]; p = 0.006, exact Wilcoxon rank test). Comparing each marker separately, we found no significant differences for any marker between control group 1 and control group 2, suggesting random DNA loss throughout the genome.

On average, cells from groups A, B, and C displayed 37.3% (95% CI, 32.8%-41.8%), 48.7% (95% CI, 36.5%–60.8%) and 48.2% (95% CI, 41.7%–54.8%) loss of informative markers, respectively. For statistical comparison of DNA losses, we used only the control cells from the age-matched group (control group 2). All three groups (A, B and C) showed a significantly ($p < 0.001$) higher percentage of LOH than control group 2 cells when evaluated by exact Wilcoxon rank test (Fig. 5.15). Cytokeratin-positive cells that harboured chromosomal abnormalities (groups B and C) displayed a slightly higher rate of DNA loss than cytokeratin-positive cells with normal karyotypes (group A). The significantly higher number of LOH in cytokeratin-positive cells without chromosomal imbalances (group A) compared to age-matched, bone marrow-derived cytokeratin-negative cells suggests that cytokeratin antibodies identify a subpopulation within the bone marrow that displays significant genetic instability.

We therefore attempted to define criteria by which an individual cytokeratin-positive cell with a normal CGH profile can be identified as a tumour cell and constructed a classifier that was trained to differentiate between group 2 control cells and group A cells. Best classification was obtained with two markers only, D16S485 and the microsatellite marker that maps within the beta-catenin gene. Control cells from group 2 could be separated from group A cells with an accuracy of 74% (the 95% CI being 64%–84%; Fig. 5.16). Generally, this classifier can be used to select breast tumour cells or control cells on the basis of LOH markers. We tested this approach and were able to correctly identify control cells from group 1 and tumour cells from group B/C cells with a high specificity (88% control group 1, 90% group B/C).

In conclusion, our finding of disseminated breast cancer cells in bone marrow that show less progressed genomic changes than preinvasive primary lesions provides a mechanism to uncover cancer-initiating and -promoting genetic or epigenetic alterations.

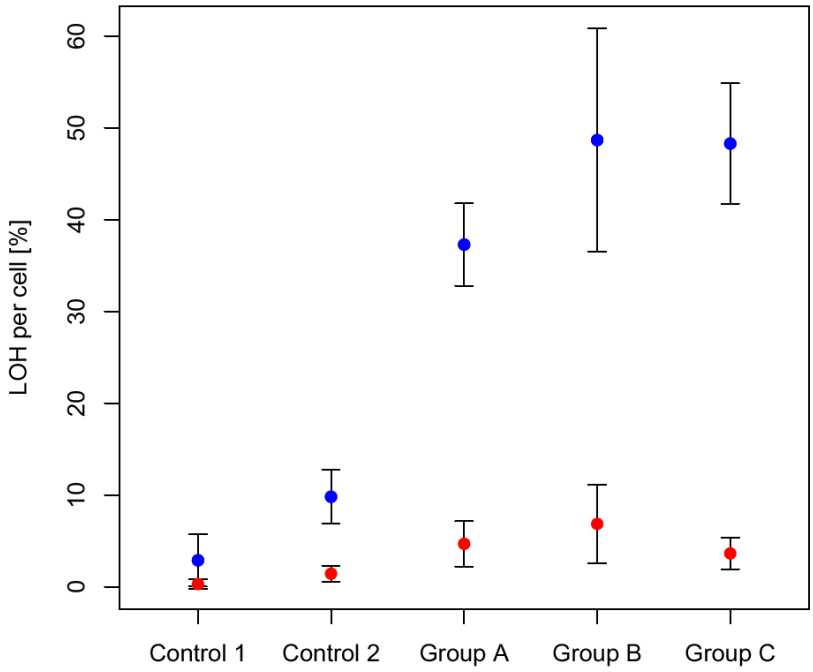


Figure 5.15: Comparison of loss of heterozygosity of all case and control groups. Allelic loss per cell of all tested markers evaluated for control groups 1 and 2 and cytokeratin-positive cells from groups A–C (groups A and B were from M0 stage patients, group C was from M1 stage patients, and groups B and C were from patients with aberrant CGH profiles). Blue dots indicate heterozygous loss, and red dots indicate homozygous deletions. Error bars represent 95% confidence intervals.

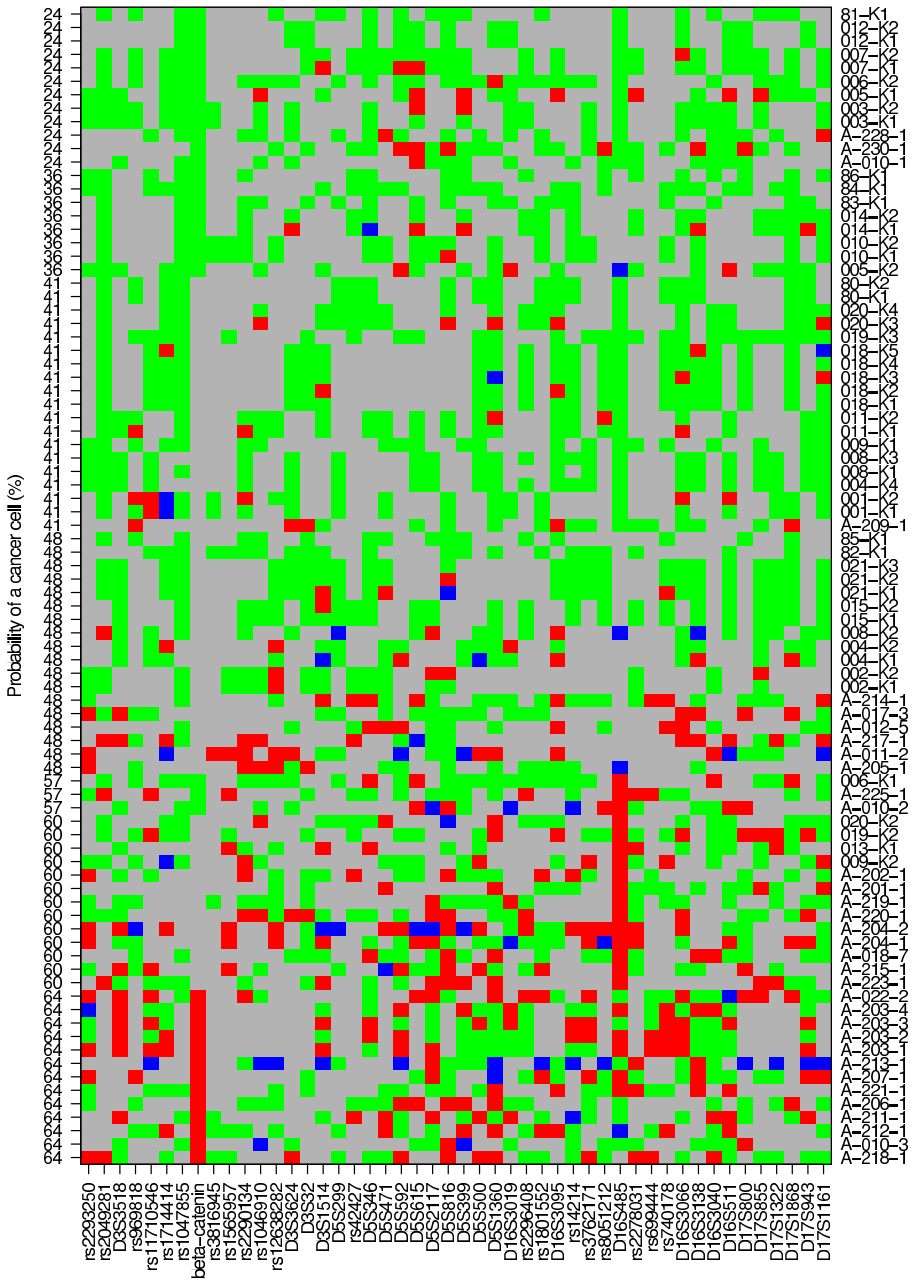


Figure 5.16: A classifier was trained to distinguish between control cells and cytokeratin-positive cells without CGH aberrations. Each column denotes a genomic marker and each row a cell. All markers (shown at the bottom of the figure) are ordered according to their chromosomal location on chromosomes 3, 5, 16, and 17. Cell identifiers are given on the right side (with letter A indicating cells from group A and letter K indicating control group 2 cells). Cells are ordered according to their probability of being classified as cells from group A. Green colour indicates informative markers, red indicates heterozygous loss and blue indicates homozygous loss.

Chapter 6

General discussion and outlook

6.1 Discussion

In this thesis, I considered machine learning and its application to genomic profiles. I studied different supervised machine learning methods and suggested a special loss function for a support vector machine and survival data.

Furthermore, I developed a workflow for machine learning of genomic profiles. This workflow starts with preprocessing, feature selection and discretisation of genomic profiles, includes strategies to deal with missing values and provides a multi-resolutional analysis. Then, training and analysis of a classifier is performed.

Additionally, I propose a case-based analysis of the classification results. This case-based analysis method can be used as an explanation scheme of a decision support system and seems to be indispensable for most clinical applications of a classification system.

My main contributions to the field of computer science apart from the workflow itself (chapter 4) are:

- Loss function for support vector machines and survival data (section 2.4.3)
- Special preprocessing method for classical CGH profiles (section 4.2.1; published in [STJE05])
- Case-based analysis of the result of classification algorithms (section 4.4; published in [SMF+03]).

I applied the workflow or parts of it to five different genomic data sets. The methods used were not restricted to classification algorithms or supervised machine learning. The toolbox of methods needed to answer the questions raised by biologists also included unsupervised machine learning and classical statistics.

Finally, the following questions have been analysed using these methods:

- Can sub groups of histological defined cancer types be determined?
  - Principal component analysis
  - Self organising maps
  - Clustering
- Are genomic aberrations associated with certain clinical variables, such as age, gender and tumour size?
  - Association rules
  - Classical statistics
- Are different tumour types distinguishable?
  Which aberrations (signatures) distinguish one tumour type from another?
  - Decision trees
  - Support vector machines

I also contributed to the field of cancer research as a co-author of publications that revealed new knowledge about

- Early metastasis of breast cancer [HGS+08]
- Ductal and lobular breast cancer [SRS+06]
- Liposarcoma [FSW+02]
- Early breast cancer (drafted in [WSS+08])
- Loss of heterozygosity analysis of single cells [SMH+05]

Cancer can be caused by, and often correlates with, a combination of genomic alterations. In this thesis, I contributed to the detection of new tumour markers, namely FUS, ITGAX [SRS+06], and CCND1, MDM2, GLI, CDK4, MYB, ESR1 and AIB1 [FSW+02].

To summarise, this work demonstrates that machine learning methods can improve our understanding of genomic aberrations and may help to improve the delivery of therapies to cancer patients.

6.1.1 Customising the learning algorithm

Generally speaking, the following strategies are possible to customise the support vector machine learning algorithm for a biological application:

1 Data preprocessing

2 Incorporation of previous knowledge as support vectors

3 Adaptation of the kernel function

4 Adaptation of the loss function

5 Domain specific interpretation of the results

Adequate data preprocessing for the domain of genomic profiles is discussed in chapter 4. The incorporation of previous knowledge by means of artificial support vectors was suggested by Schölkopf [Sch97]. I did not use this technique because no operable previous knowledge was available in this domain. The customisation of the kernel function (e.g. [RSS05]) should be considered in future studies. Specifically, a customised string kernel could be used to analyse genomic profiles with a one base pair resolution (see section 6.2.3). So far, I have transformed the data using appropriate data preprocessing in such a way that it could be learned by a usual kernel. A specific loss function for survival data is proposed in section 2.4.3. The interpretation of the results of the learning algorithm is discussed in section 4.4.

6.1.2 Workflow

To the best of my knowledge, no other machine learning workflow of genomic profiles has so far been published. Nevertheless, workflows and software packages for analysing genomic profiles (often array CGH data) are available (e.g. [LHN+06, WMZKM04, KNL+05]) but cover the analysis from the raw data to the assignment of losses and gains only.

I validated the proposed workflow using different strategies. First of all, null models (using permuted class labels) were analysed and did not reveal false positive results (given the data sets described in chapter 5). Furthermore,

the aberrations of a simulated data set could be detected (see section 5.2.4). Finally, the usefulness of my approach was demonstrated by analysing genomic profiles from five different biological data sets. Here, the results revealed by machine learning were consistent with previous biological knowledge and statistical tests (chapter 5).

The algorithm choice at each part of the workflow was directed by the literature and the biology of genomic profiles. A full search through all options at each part of the workflow was not performed, as the search space would have been too large (3 univariate preprocessing options $\times$ 5 multi-resolutional preprocessing options $\times$ 5 missing value options $\times$ 2 discretisation options $\times$ 4 classification algorithms = 600 multiple tests).

Although the model selection approach depends on a distributed computing infrastructure, it is still time consuming. Therefore, other model selection approaches should be chosen in the future.

6.1.3 Supervised, and unsupervised machine learning and classical statistics

Remarkably, classical statistics and supervised machine learning methods identified similar discriminating genomic regions (e.g., Fig. 5.9). However, both analysis methods complement each other. Classical statistics reveals a probability that two tumour types are different whereas machine learning provides a rule regarding the class assignment of unclassified (future) cases.

The choice of the classification algorithm seems to be not as important as long as the features are appropriately preprocessed. However, successful classifiers share a similar background, as this is the the case for support vector machines and boosting. Moreover, the observed non-linearity in genomic profiles is low due to the high dimensionality.

In addition, machine learning returns a ranked list of discriminating regions and it is sometimes difficult to find the correct number of discriminating regions. Even the comparison with a null model (permuted class labels) is not always sufficient. Classical statistics seems to be the more reliable approach here.

Association rules detect co-occurrences of events. A classic example is the following: ”IF a customer buys nappies THEN he will also buy beer”. When applied to genomic data sets, association rules provide thousands of rules that have to be preselected by parsing. Moreover, association rules prefer rules

that include frequent aberrations. Yet, frequent associations are often shared by different tumour subtypes and therefore are not discriminative (and not of interest).

All unsupervised machine learning methods (clustering, principal component analysis, self-organising maps) also revealed similar results. For the liposarcoma data set, unsupervised machine learning uncovered the correct class separation.

A noteworthy difference between supervised and unsupervised machine learning methods is that an unsupervised method returns a hypothesis about the structure of a data set only. Such a hypothesis has always to be proven / retracted using classical statistics (and unseen experimental data). The co-occurrence of cases in different subtrees of a hierarchical clustering is never proof that these cases belong to different (cancer) subtypes. Finally, unsupervised methods should be used for an exploration or visualisation of the data set.

A second problem relates to the stability of a hypothesis uncovered by unsupervised methods. Such a hypothesis is often very unstable and changes if a few cases from the same population are included or excluded. Methods to measure the stability of a cluster, after bootstrapping or adding some noise, exist but they are not routinely used. In this thesis (section 5.1.1), I applied a method suggested by Richard Simon [MRF+02].

6.1.4 Experimental setting and validation

So far, microarray experiments have identified thousands of candidate genes and genomic fragments. However, the statistical validity of these candidates remains often questionable. In addition, the overlap between candidates of the same tumour type revealed by different research groups or different experimental platforms is also quite low [EDZD06].

Strikingly, I often observed that classical markers (like lymph node status and tumour size) are at least as good as markers derived from genomic and gene expression profiles. New diagnostic markers may be successfully identified only if the candidates are carefully chosen (according to previous knowledge) or if the power of the underlying statistical experiment is sufficient.

The number of samples is important to draw scientific conclusions from a machine learning model. The confidence interval of the classification accuracy depends on the size of the training data set. However, even in high-

dimensional data sets large effects can be detected using a small number of cases (this was the case for the liposarcoma-dataset) whereas small effects require a large number of cases. Similarly, a small number of DNA probes from different organisms is sufficient to determine differences between different species even though this data set is very high-dimensional (approx. 3 billion base pairs). Celera Genomics used just five individuals for revealing the sequence of the human genome.

Subsequently, it would be desirable if more experimental biologists and physicians would consult a statistician or mathematical biologist at the design phase of an experiment to estimate the power or minimal number of samples to prove or falsify a hypothesis or to successfully apply a machine learning algorithm.

Furthermore, hypotheses, which were selected by machine learning, have to be validated using new samples. Resampling techniques like cross-validation can never substitute an experimental validation. Finally, it remains to be proven that a new marker outperforms current diagnostic schemes in a clinical setting.

6.1.5 Explanation scheme

The proposed explanation scheme, including a qualitative explanation, a competence score and a reliability score could be part of a decision support system based on a support vector machine classifier. Fig. 6.1 visualises the user view of such a system. The system could enable physicians to deal with complex genomic and gene expression profiles, even if they are not experts in molecular genetics or machine learning.

At the moment, the qualitative explanation is limited to two-class-problems, whereas the competence and reliability estimations are able to deal with multi-class probabilities.

As far as I am aware, another approach of a decision support system with an explanation component based on a support vector machine has not yet been proposed. Morik et al integrated an SVM in a decision support system based on first-order-logic [MIB$^+$00]. However, a case-based explanation of the SVM classifier is not provided.

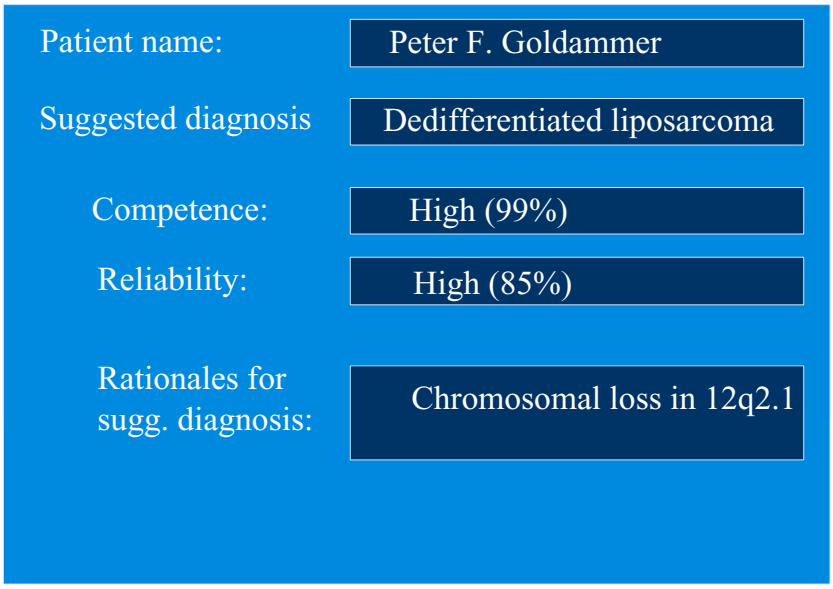


Figure 6.1: Sketch of a user interface for a decision support system.

6.2 Future directions of research

Based on my research experience, the following directions for future research at the intersection of computer science and biology, medicine and statistics can be identified:

- Future research projects in bioinformatics

  Future studies will be based on a larger number of cases and will therefore enable the detection of small genomic aberrations that correlate and may contribute to an increased disease risk in a population. Section 6.2.2 deals with disease association studies of genomic profiles.

  An important paradigm shift in biology refers to modelling rather than a descriptive analysis. This is also the case for genomic profiles. Section 6.2.1 provides ideas for modelling genomic aberrations.

  New sequencing technologies like 454 and Illumina/Solexa will enable the sequencing of individual genomes and transcriptomes [WMW+08] and will reveal genomic profiles at a one base pair resolution. The challenges of analysing these data sets are discussed in section 6.2.3.

- Future research projects in computer science / statistical learning

  The increasing amount of data in classification studies raises the need

for classifiers with a low time complexity. This problem is covered in section 6.2.4.

An unsolved problem relates to the (optimal) choice of a classification algorithm for a given task. This issue is discussed in section 6.2.5.

- Future research projects in medical informatics

  To improve cancer diagnosis and management, the knowledge gained in cancer research has to be transferred to the point of care. A useful tool for the deployment of this knowledge would be a decision support system. Section 6.2.6 discusses applications of classifiers as decision support systems in healthcare and the vision of personalised medicine.

6.2.1 Understanding cancer by modelling genomic aberrations

Genomic and gene expression profiles improve our understanding of cancer. However, the state-of-the-art outcome of such a study is a signature that depicts genomic aberrations for a specific (sub) type of cancer at a specific time point. Most of the studies reveal a descriptive view of pathological processes only. A more comprehensive view and a mechanistic understanding of tumorigenesis could emerge from modelling.

One example of a successfully applied model of genomic profiles is evolutionary trees (e.g., [DJK+99, BRK+05, BSB+04]), revealing the order of genomic aberrations. Each node of such a tree (Fig. 6.2) depicts a mutation and each edge labels a conditional probability for this mutation. However, given the aforementioned data sets of limited size and high dimensionality, existing algorithms failed to discover biologically relevant knowledge.

For gene expression profiles, the modelling of transcriptional networks is an important research topic [SFK+05]. An immense number of publications have emerged in the field of learning signalling networks in recent years. Yet, most of the algorithms are restricted to specific experimental designs or a large sample number (e.g., [MBS05, Fri04, Wag01, XVDL05]). Finally, the biological relevance of most of the existing models is limited.

The most often used graphical network models belong to the class of Bayesian networks [BK02, Jor98]. Bayesian networks are restricted to acyclic graphs but the biology (e.g. of intrinsic gene buffering) involves feedback loops. Therefore, the reconstruction of cyclic graphs will be a challenge and will

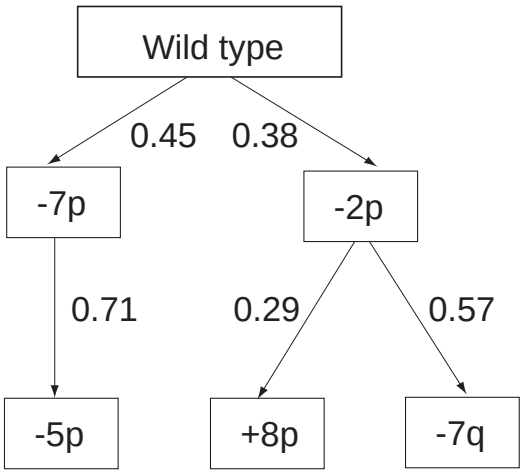


Figure 6.2: Schematic example of an oncotree. Each node depicts a mutation and each edge labels a conditional probability for this mutation.

imply the use of Dynamic Bayesian Networks.

Another approach emerges from the field of control theory [CD02, SSS+04] and seems to be suitable to model feedback loops. Additionally, recursive artificial neural networks could be trained with stimuli influencing cell death, morphogenesis and differentiation. Jaeger et al. proposed in 2004 an innovative method to learn such a network [JH04].

6.2.2 Large scale disease association studies

Recently, it has been found that the copy number ratios of a normal human population are diverse [RIF+06].

The overall aim of disease association studies is to analyse copy number variations in humans and their correlation to human diseases (e.g., diabetes). Here, the challenge is to deal with the high dimensionality of the data (approx. 1 million polymorphisms). On the other hand, samples are limited both by their availability and the cost of hybridisations.

A univariate analysis would reveal all genomic regions (given a minimal frequency and length) that show a significant association with the disease. This analysis should be corrected by using the false discovery rate.

One possible solution, to deal with the high dimensionality, would be a two-step analysis. Step one is used to generate a hypothesis which is proven or retracted in step two. Step one involves the study of a limited number of

cases and all polymorphisms. A univariate analysis could then lead to a first hypothesis. Finally, a chosen subset of hypotheses should then be analysed in step two on new cases.

A multivariate analysis should be based on the logistic regression. However, other methods may contribute to additional knowledge about associations.

6.2.3 Analysing genomic profiles at one base pair resolution

Recently, new sequencing technologies like 454 and Solexa have emerged that reduce the costs and improve the throughput by a factor of at least one hundred [Ben06, Car07]. I was involved in analysing and modelling of an Eucaryotic transcriptome using such a technique [WMW+08].

The genomic profiles revealed by sequencing have a one base pair resolution and a low-signal-to-noise ratio. In comparison, the resolution of the studies I analysed was approx. 1000 base pairs. There is hope that future sequencing approaches will provide an individual genomic profile (copy number variation and structural re-arrangements) at a cost of 1000 dollars (within the next 10 years or so).

An unsolved problem is that sequence reads are small (approx. 35 base pairs) and it is currently not possible to map all of them uniquely to a reference genome (especially repeat regions). For the Human genome, it is estimated that only 80% of the genome is mappable [Ben06].

From a computational point of view, sequencing produces huge amounts of data (in the order of TBytes) and requires a massive parallel analysis of the resulting data set. Interestingly, a genomic profile at a one-base pair resolution could be interpreted as a string (based of the four letters A,T,G,U) and analysed using a string kernel of a support vector machine.

6.2.4 Large scale classification problems

The time complexity of a classifier is an issue, especially for future applications like large-scale meta analyses or large-scale association studies.

Therefore I initiated, together with Stefan Hezel, a project to speed up an SVM classifier using special hardware (FPGA, Field Programmable Gate Arrays). In cooperation with the Department of Technical Informatics at

the University of Mannheim, a hardware-based support of a SVM-classifier was designed by Dirk Fuchs [Fuc05].

Another possibility is the use of classifiers with a better training time complexity like Sparse Grids ([GGT01] and [GG02]) which scales linearly with the number of samples.

To narrow down the computational complexity of the SVM learning algorithm, a subset of cases in the training set could be used as base kernels [TS01].

6.2.5 General theory of classifier choice

Currently, only "fragments of a computational theory of learning" exist [Mit97] and these fragments are often based on unrealistic assumptions like noise-free training data. Examples are the statistical learning theory by Vapnik [Vap95] and the theory of probably approximately correct (PAC) learning [Mit97, Hau90].

What seems to be missing is a general rule or general theory of selecting an appropriate classification algorithm for a given classification problem. So far, the choice of the classifier seems to be directed by personal experience and benchmark studies using data repositories only.

However, it would be desirable to base the choice of a classifier on the structure of the decision problem and on a theory instead of a rule of thumb. So the question emerges:

Which classifier seems to be successfully applicable considering

- the size (number of cases),
- the number of classes,
- dimensionality (number of features),
- univariate distribution (in particular discrete or continuous),
- multivariate distribution (correlation structure) and
- previous knowledge of a learning problem?

6.2.6 Personalised medicine

Gene expression and genomic profiles can be used as diagnostic markers and it has been shown that such profiles can outperform current diagnostic and prognostic classification schemes. New diagnostic markers like genomic polymorphisms have the potential to revolutionise medical diagnostics and will finally lead to a personalised delivery of healthcare [MSMLC02]. This includes tailoring the drug dosage and timing to individual patients, the assessment of personal risk factors (e.g. smoking or sun) and the administration of antibodies/inhibitors for proteins with an increased expression or activation.

Furthermore, kits for analysing gene expression and genomic profiles ("lab on a chip") or sequencing will enter the clinical practice. Yet, the integration of gene expression and genomic profiles in the diagnostic process remains a challenge. One possible solution is to use the knowledge generated by bioinformatics methods in clinical decision support systems. Such systems should enable physicians to analyse and interpret such complex profiles - even if they are not experts in molecular genetics - as easily as analysing and interpreting current laboratory results.

List of Figures

List of Tables

Abbreviations

Abbreviation	Description
BM	Bone marrow
CGH	Comparative genomic hybridisation
CI	Confidence interval
CK	Cytokeratin
CV	Cross validation
DL	Dedifferentiated liposarcoma (cancer subtype)
DNA	Deoxyribonucleic acid
FDR	False discovery rate
LN	Lymph node
LOH	Loss of heterozygosity
LOO-CV	Leave-one-out cross validation
M0	Cancer patient without clinically manifest metastases
M1	Cancer patient with clinically manifest metastases
PCA	Principal component analysis
PCR	Polymerase chain reaction (molecular biological technique for replicating DNA)
PL	Pleiomorphic liposarcoma (cancer subtype)
R	Statistical software package
RBF	Radial basis function
SAM	Significance analysis of microarrays
SVM	Support vector machine

Notation

Symbol	Description
i, j	counter
x	scalar
z^*	complex conjugate of $z \in \mathbb{C}$
$\boldsymbol{x}$	vector
$\boldsymbol{x}^t$	transposed vector $\boldsymbol{x}$
$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of non-negative real numbers
$\mathbb{S}$	training data
$M = \lvert\mathbb{S}\rvert$	number of cases
$\mathbb{X}$	input space
$N = \dim(\mathbb{X})$	dimensionality of the input space
$\boldsymbol{x_i}$	input or feature vector
$\mathbb{Y}$	space of the target variable
y_i	target variable, class label of an input vector $\boldsymbol{x}_i$
$K = \lvert\mathbb{Y}\rvert$	number of classes of the target variable
$\mathbb{F}$	feature space
$\Psi(\boldsymbol{x})$	mapping function from the input to the feature space
$\mathbb{A}$	set (list) of all feature names
$f(\boldsymbol{x})$	decision function
$\mathfrak{F}$	function space
$\boldsymbol{\omega}$	hyperplane direction vector
ρ	margin
r	radius of the smallest sphere around the training data $\mathbb{S}$
sign	sign function
$\arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$	argument of the minimum, returns $\boldsymbol{x}_*$ that minimises $f(\boldsymbol{x}_*)$
$P(A)$	probability of an event A
$p(\boldsymbol{x}, y)$	density of a probability distribution
$R(f)$	expected risk of a function f
$R_{emp}(f, \mathbb{S})$	empirical risk of a function f on training data $\mathbb{S}$

Symbol	Description
$R_{reg}(f, \mathbb{S})$	regularised risk of a function f on training data $\mathbb{S}$
$R_{cross}(f, \mathbb{S})$	risk of a function f on training data $\mathbb{S}$ estimated by cross-validation
$R_{boot}(f, \mathbb{S})$	empirical risk of a function f on training data $\mathbb{S}$ estimated by bootstrap
$\mathcal{L}$	loss function
$\|\boldsymbol{x}\|$	absolute value of $\boldsymbol{x}$
$\|\mathbb{S}\|$	number of elements in the set $\mathbb{S}$
$\|.\|_2$	L2 norm
$\|.\|_1$	L1 norm
ϵ	accuracy parameter
δ	confidence parameter
θ	Vapnik-Chervonenkis dimension (VC-dimension)
Φ	regularisation operator
ξ_i, ξ_i^*	slack variables
$\boldsymbol{s}_i$	support vector
C	regularisation parameter
$cens$	censor vector, each element is 0 for a censored observation and 1 otherwise
d	degree of the polynomial kernel
γ	parameter of the polynomial kernel
c	parameter of the polynomial kernel
$\boldsymbol{\alpha}$	weight vector of the decision function and coefficients of the support vectors
b	bias of the decision function
$L(x)$	description length (in bits)
k	number of partitions used in cross-validation
$L_i(x)$	Boolean expression of binary feature vectors
$K(\boldsymbol{x}_1, \boldsymbol{x}_2)$	kernel, scalar product in feature space
$\mathbb{T}$	working set of a decision tree algorithm
$\langle . \rangle$	scalar product in Euclidean space
$\langle . \rangle_{\mathfrak{H}}$	scalar product in Hilbert space
$x_1 == x_2$	comparison of equality
$x_1 \neq x_2$	comparison of inequality
$x_1 := x_2$	assignment, x_1 becomes x_2
$O(.)$	computational complexity of an algorithm
$\psi_{u,s}(x)$	wavelet
u	translation parameter of a wavelet function
s	scaling parameter of a wavelet function

Bibliography

[ABR+05] D. Anguita, A. Boni, S. Ridella, F. Rivieccio, and D. Sterpi. Theoretical and practical model selection methods for support vector classifiers. In L. Wang, editor, *Support vector machines: Theory and applications*. Springer, Berlin, 2005.

[BC01] M. Baudis and M. L. Cleary. Progenetix.net: An online repository for molecular cytogenetic aberration data. *Bioinformatics*, 17(12):1228–1229, 2001.

[BD05] N. Barakat and J. Diederich. Eclectic rule-extraction from support vector machines. *International Journal of Computational Intelligence*, 2(1):59–62, 2005.

[Ben06] D. R. Bentley. Whole-genome re-sequencing. *Current Opinions in Genetics and Development*, (16):545–552, 2006.

[BGP03] A. Balmain, J. Gray, and B. Ponder. The genetics and genomics of cancer. *Nat Genet*, 33 Suppl:238–244, 2003.

[BHDM+95] M. Bentz, K. Huck, S. Du Manoir, S. Joos, C. A. Werner, K. Fischer, H. Dohner, and P. Lichter. Comparative genomic hybridization in chronic b-cell leukemias shows a high incidence of chromosomal gains and losses. *Blood*, 85(12):3610–3618, 1995.

[BIAS03] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*, (19):185–193, 2003.

[Bis94] C. Bishop. Novelty detection and neural network validation. In *IEE Proceedings - Vision, Image and Signal processing*, pages 217–222, 1994.

[Bis95] C. M. Bishop. *Neural networks for pattern recognition*. Oxford Press, 1995.

[BK02] C. Borgelt and R. Kruse. *Graphical models, methods for data analysis and mining*. Wiley, Chichester, England, 2002.

[BLS06] S. A. Barnes, S. R. Lindborg, and Jr. Seaman, J. W. Multiple imputation techniques in small sample clinical trials. *Stat Med*, 25(2):233–245, 2006.

[BND04] U. M. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[BPM+00] S. Braun, K. Pantel, P. Muller, W. Janni, F. Hepp, C. R. Kentenich, S. Gastroph, A. Wischnik, T. Dimpfl, G. Kindermann, G. Riethmuller, and G. Schlimok. Cytokeratin-positive cells in the bone marrow and survival of patients with stage I, II, or III breast cancer. *N Engl J Med*, 342(8):525–533, 2000.

[BPS+98] M. Bentz, A. Plesch, S. Stilgenbauer, H. Dohner, and P. Lichter. Minimal sizes of deletions detected by comparative genomic hybridization. *Genes Chromosomes Cancer*, 21(2):172–175, 1998.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[BRK+05] N. Beerenwinkel, J. Rahnenfuhrer, R. Kaiser, D. Hoffmann, J. Selbig, and T. Lengauer. Mtreemix: A software package for learning and using mixture models of mutagenetic trees. *Bioinformatics*, 21(9):2106–2107, 2005.

[BSB+04] S. Bulashevska, O. Szakacs, B. Brors, R. Eils, and G. Kovacs. Pathways of urothelial cancer progression suggested by bayesian network analysis of allelotyping data. *Int J Cancer*, 110(6):850–856, 2004.

[BWD+96] M. Bentz, C. A. Werner, H. Dohner, S. Joos, T. F. Barth, R. Siebert, M. Schroder, S. Stilgenbauer, K. Fischer, P. Moller, and P. Lichter. High incidence of chromosomal imbalances and gene amplifications in the classical follicular variant of follicle center lymphoma. *Blood*, 88(4):1437–1444, 1996.

[BZL+04] C. Brennan, Y. Zhang, C. Leo, B. Feng, C. Cauwels, A. J. Aguirre, M. Kim, A. Protopopov, and L. Chin. High-resolution global profiling of genomic alterations with long oligonucleotide microarray. *Cancer Res*, 64(14):4744–4748, 2004.

[Car07] N.P. Carter. Methods and strategies for analyzing copy number variation using dna microarrays. *Nat Genet*, 39 Suppl 1(7s):S16–S21, 2007.

[CBB02] R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. *Neural Comput*, 14(5):1105–1114, 2002.

[CD02] M. E. Csete and J. C. Doyle. Reverse engineering of biological complexity. *Science*, 295(5560):1664–1669, 2002.

[CL01] C.-C. Chang and C.-J. Lin. Libsvm : A library for support vector machines, 2001.

[CLB+02] B. Coiffier, E. Lepage, J. Briere, R. Herbrecht, H. Tilly, R. Bouabdallah, P. Morel, E. Van Den Neste, G. Salles, P. Gaulard, F. Reyes, P. Lederlin, and C. Gisselbrecht. Chop chemotherapy plus rituximab compared with chop alone in elderly patients with diffuse large-b-cell lymphoma. *N Engl J Med*, 346(4):235–242, 2002.

[CM01] G. Carenini and J. D. Moore. An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *IJCAI*, pages 1307–1314, Seattle, Washington, USA, 2001.

[CM04] V. Cherkassky and Y. Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Netw*, 17(1):113–126, 2004.

[CST00] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, 2000.

[CVBM02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

[DB95] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[DH96] J. Doonan and T. Hunt. Cell cycle. why don't plants get cancer? *Nature*, 380(6574):481–482, 1996.

[Dij04] A. Dijksterhuis. Think different: The merits of unconscious thought in preference development and decision making. *J Pers Soc Psychol*, 87(5):586–598, 2004.

[Din55] G. P. Dinneen. Programming pattern recognition. In *Western Joint Computer Conference*, volume 94-100, New York: Institute of Radio Engineers, 1955.

[DJK+99] R. Desper, F. Jiang, O.-P. Kallioniemi, H. Moch, Papadimitriou, and A.A. C.H., S. Inferring tree models for oncogenesis from comparative genome hybridization data. *J. Comp. Biol.*, 6:37–51, 1999.

[DMSJ+93] S. Du Manoir, M. R. Speicher, S. Joos, E. Schrock, S. Popp, H. Dohner, G. Kovacs, M. Robert-Nicoud, P. Lichter, and T. Cremer. Detection of complete and partial chromosome gains and losses by comparative genomic in situ hybridization. *Hum Genet*, 90(6):590–610, 1993.

[EDZD06] L. Ein-Dor, O. Zuk, and E. Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *PNAS*, 103(15):5923–2928, 2006.

[EPP00] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

[Fri84] J. H. Friedman. A variable span scatterplot smoother. Technical Report 5, Laboratory for Computational Statistics, 1984.

[Fri04] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[FS96] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *International Conference on Machine Learning*, pages 148–156, Bari, Italy, 1996. Morgan Kaufmann.

[FSW+02] B. Fritz, F. Schubert, G. Wrobel, C. Schwaenen, S. Wessendorf, M. Nessling, C. Korz, R. J. Rieker, K. Montgomery, R. Kucherlapati, G. Mechtersheimer, R. Eils, S. Joos, and P. Lichter. Microarray-based copy number and expression

profiling in dedifferentiated and pleomorphic liposarcoma. *Cancer Res*, 62(11):2993–2998, 2002.

[Fuc05] D. Fuchs. Entwicklung eines FPGA-basierten Support-Vektor Maschinen Klassifikators. Master's thesis, (Diplomarbeit), Universität Mannheim, 2005.

[FZ05] H. Fröhlich and A. Zell. Efficient parameter selection for support vector machines in classification and regression via model-based global optimization. In *Int. Joint Conf. on Neural Networks (IJCNN)*, pages 1431 – 1438, 2005.

[GG02] J. Garcke and M. Griebel. Classification with sparse grids using simplicial basis functions. *Intelligent Data Analysis*, 6:483 – 502, 2002.

[GGT01] J. Garcke, M. Griebel, and M. Thess. Data mining with sparse grids. *Computing*, 67:225–253, 2001.

[Gir98] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Comput*, 10(6):1455–1480, 1998.

[GTG99] G. Gigerenzer, P. M. Todd, and A. R. Group. *Simple heuristics that make us smart*. Oxford University Press, Oxford, 1999.

[GWBV02] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[GWHMR+04] A. M. Gruszka-Westwood, S. W. Horsley, A. Martinez-Ramirez, C. J. Harrison, H. Kempski, A. V. Moorman, F. M. Ross, M. Griffiths, M. F. Greaves, and L. Kearney. Comparative expressed sequence hybridization studies of high-hyperdiploid childhood acute lymphoblastic leukemia. *Genes Chromosomes Cancer*, 41(3):191–202, 2004.

[Has92] S. Hashem. Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions. In *IJCNN*, Baltimore, Maryland, USA, 1992. IEEE Press.

[Hau90] D. Haussler. Probably approximately correct learning. In *National Conference on Artificial Intelligence*, pages 1101–1108, Boston, MA, 1990.

[HG03] B. Hammer and K. Gersmann. A note on the universal approximation capability of support vector machines. *Neural Process. Lett.*, 17(1):43–53, 2003.

[HGS+08] Y. Hüsemann, J. Geigl, F. Schubert, M. Meyer, E. Burghart, G. Forni, R. Eils, P. Musiani, G. Riethmüller, and C. Klein. Metastasis is initiated prior to mammary tumour formation in HER-2 transgenic mice. *Cancer Cell*, 13(1):58–68, 2008.

[HL99] D. W. Hosmer and S. Lemeshow. *Applied survival analysis: Regression modeling of time to event data*. Wiley-Interscience, New York, 1999.

[HM04] W. Härdle and R. Moro. Talk: Survival analysis with support vector machines., 2004.

[Hor04] G. N. Hortobagyi. Opportunities and challenges in the development of targeted therapies. *Semin Oncol*, 31(1 Suppl 3):21–27, 2004.

[HSG+05] L. Hsu, S. G. Self, D. Grove, T. Randolph, K. Wang, J. J. Delrow, L. Loo, and P. Porter. Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics*, 6(2):211–226, 2005.

[HSS05] D. Hush, C. Scovel, and I. Steinwart. Polynomial time algorithms for computing approximate SVM solutions with guaranteed accuracy. Technical Report LA-UR-05-7738, Los Alamos National Laboratory, 2005.

[HST+04] P. Hupe, N. Stransky, J. P. Thiery, F. Radvanyi, and E. Barillot. Analysis of array CGH data: From signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20(18):3413–3422, 2004.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, New York, Berlin, Heidelberg, 2001.

[HW00] D. Hanahan and R. A. Weinberg. The hallmarks of cancer. *Cell*, 100(1):57–70, 2000.

[HWL06] T.-K. Huang, R. C. Weng, and C.-J. Lin. Generalized bradley-terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7:85–115, 2006.

[HZ03] P. Houshmand and A. Zlotnik. Targeting tumor cells. *Curr Opin Cell Biol*, 15(5):640–644, 2003.

[HZET07] P.A. Habas, J.M. Zurada, A.S. Elmaghraby, and G.D. Tourassi. Reliability analysis framework for computer-assisted medical decision systems. *Med. Phys.*, 34(2):763–772, 2007.

[IL04] S. Iqbal and H. J. Lenz. Angiogenesis inhibitors in the treatment of colorectal cancer. *Semin Oncol*, 31(6 Suppl 17):10–16, 2004.

[IS96] A. Ittner and M. Schlosser. Non-linear decision trees. In L. Saitta, editor, *Proceedings of the 13th International Conference of Machine Learning (ICML'96)*, pages 252–257, Bari, Italy, 1996. Morgan Kaufmann Publishers.

[Itt05] C. Ittrich. Normalization for two-channel microarray data. *Methods Inf Med*, 44(3):418–422, 2005.

[JH04] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[JMM+04] K. Jong, E. Marchiori, G. Meijer, A. V. Vaart, and B. Ylstra. Breakpoint identification and smoothing of array comparative genomic hybridization data. *Bioinformatics*, 20(18):3636–3637, 2004.

[Joa99] T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning*. MIT-Press, Cambridge, MA, USA, 1999.

[Jor98] M. I. Jordan, editor. *Learning in graphical models*. Kluwer Academic Pubishers, Dordecht, Netherlands, 1998.

[JWWO05] R. Jornsten, H. Y. Wang, W. J. Welsh, and M. Ouyang. DNA microarray data imputation and significance analysis of differential expression. *Bioinformatics*, 21(22):4155–4161, 2005.

[KGM+99] M. Kirchhoff, T. Gerdes, J. Maahr, H. Rose, M. Bentz, H. Dohner, and C. Lundsteen. Deletions below 10 megabasepairs are detected in comparative genomic hybridization by standard reference intervals. *Genes Chromosomes Cancer*, 25(4):410–413, 1999.

[KGV83] S. Kirkpatrick, Jr. Gelatt, C. D., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KKP+94] O. P. Kallioniemi, A. Kallioniemi, J. Piper, J. Isola, F. M. Waldman, J. W. Gray, and D. Pinkel. Optimizing comparative genomic hybridization for analysis of DNA sequence copy number changes in solid tumors. *Genes Chromosomes Cancer*, 10(4):231–243, 1994.

[KKS+92] A. Kallioniemi, O. P. Kallioniemi, D. Sudar, D. Rutovitz, J. W. Gray, F. Waldman, and D. Pinkel. Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science*, 258(5083):818–821, 1992.

[Kle03] C. A. Klein. The systemic progression of human cancer: A focus on the individual disseminated cancer cell - the unit of selection. *Adv Cancer Res*, 89:35–67, 2003.

[KNL+05] S. Y. Kim, S. W. Nam, S. H. Lee, W. S. Park, N. J. Yoo, J. Y. Lee, and Y. J. Chung. ArrayCyGHt: A web application for analysis and visualization of array-CGH data. *Bioinformatics*, 21(10):2554–2555, 2005.

[Koh95a] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.

[Koh95b] T. Kohonen. *Self-organizing maps*. Springer, Berlin, 1995.

[Kol92] J. L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1):3–34, 1992.

[KR05] C. Kooperberg and I. Ruczinski. Identifying interacting snps using monte carlo logic regression. *Genet Epidemiol*, 28(2):157–170, 2005.

[KRLH01] C. Kooperberg, I. Ruczinski, M. L. Leblanc, and L. Hsu. Sequence analysis using logic regression. *Genet Epidemiol*, 21 Suppl 1:S626–631, 2001.

[KSKS+99] C. A. Klein, O. Schmidt-Kittler, J. A. Schardt, K. Pantel, M. R. Speicher, and G. Riethmuller. Comparative genomic hybridization, loss of heterozygosity, and dna sequence analysis of single cells. *Proc Natl Acad Sci U S A*, 96(8):4494–4499, 1999.

[Lew00] B. Lewin. *Genes VII*. Oxford University Press, New York, 2000.

[LHN+06] S. Liva, P. Hupe, P. Neuvial, I. Brito, E. Viara, P. La Rosa, and E. Barillot. Capweb: A bioinformatics CGH array analysis platform. *Nucleic Acids Res*, 34(Web Server issue):W477–481, 2006.

[Lis02] P. J. Lisboa. A review of evidence of health benefit from artificial neural networks in medical intervention. *Neural Netw*, 15(1):11–39, 2002.

[LJBL00] P. Lichter, S. Joos, M. Bentz, and S. Lampel. Comparative genomic hybridization: Uses and limitations. *Semin Hematol*, 37(4):348–357, 2000.

[LJKP05] W. R. Lai, M. D. Johnson, R. Kucherlapati, and P. J. Park. Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, 21(19):3763–3770, 2005.

[LKV98] C. Lengauer, K. W. Kinzler, and B. Vogelstein. Genetic instabilities in human cancers. *Nature*, 396(6712):643–649, 1998.

[LW93] H.-T. L.-J. Lin and R. C. Weng. A note on Platt's probabilistic outputs for support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 1993.

[LWC+01] Y. J. Lu, D. Williamson, J. Clark, R. Wang, N. Tiffin, L. Skelton, T. Gordon, R. Williams, B. Allan, A. Jackman, C. Cooper, K. Pritchard-Jones, and J. Shipley. Comparative expressed sequence hybridization to chromosomes for tumor classification and identification of genomic regions of differential gene expression. *Proc Natl Acad Sci U S A*, 98(16):9197–9202, 2001.

[LWW+03] Y. J. Lu, D. Williamson, R. Wang, B. Summersgill, S. Rodriguez, S. Rogers, K. Pritchard-Jones, C. Campbell, and

J. Shipley. Expression profiling targeting chromosomes for tumor classification and prediction of clinical behavior. *Genes Chromosomes Cancer*, 38(3):207–214, 2003.

[Mac03] D. Mackay. *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge, 2003.

[Mal99] S. Mallat. *A wavelet tour of signal processing*. Academic Press, New York, 1999.

[MBS05] F. Markowetz, J. Bloch, and R. Spang. Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics*, 2005.

[McC04] P. McCorduck. *Machines who think*. A K Peters, Natick, MA, 2004.

[MHM+01] O. Monni, E. Hyman, S. Mousses, M. Barlund, A. Kallioniemi, and O. P. Kallioniemi. From chromosomal alterations to target genes for therapy: Integrating cytogenetic and functional genomic views of the breast cancer genome. *Semin Cancer Biol*, 11(5):395–401, 2001.

[MIB+00] K. Morik, M. Imhoff, P. Brockhausen, T. Joachims, and U. Gather. Knowledge discovery and knowledge validation in intensive care. *Artif Intell Med*, 19(3):225–249, 2000.

[Mik02] S. Mika. *Kernel fisher discriminants*. PhD thesis, Technische Universität Berlin, 2002.

[Mil56] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review.*, 63:81–97, 1956.

[Mit95] F. Mitelman, editor. *ISCN 1995: An international system for human cytogenetic nomenclature (1995)*. Karger, Basel, 1995.

[Mit97] T. M. Mitchell. *Machine learning*. McGraw-Hill, Boston, Massachusetts, 1997.

[Mül02] J. Müller. Entwurf eines wissensbasierten Systems für molekulargenetische Daten auf Basis einer Support Vektor Maschine. Master's thesis, (Diplomarbeit), Universität Heidelberg, Fachhochschule Heilbronn, 2002.

[MLH03] D. Meyer, F. Leisch, and K. Hornik. The support vector machine under test. *Neurocomputing*, 55:169–186, 2003.

[MMR+01] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.

[MRF+02] L. M. McShane, M. D. Radmacher, B. Freidlin, R. Yu, M. C. Li, and R. Simon. Methods for assessing reproducibility of clustering patterns observed in analyses of microarray data. *Bioinformatics*, 18(11):1462–1469, 2002.

[MSMLC02] F. Martin-Sanchez, V. Maojo, and G. Lopez-Campos. Integrating genomics into health information systems. *Methods Inf Med*, 41(1):25–30, 2002.

[MSP05] A. M. Molinaro, R. Simon, and R. M. Pfeiffer. Prediction error estimation: A comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[NS61] A. Newell and H. A. Simon. GPS: A program that simulates human thought. In H. Billings, editor, *Lernende Automaten*, pages 109–124. Oldenbourg, München, 1961.

[OH00] R. J. Osborne and M. G. Hamshere. A genome-wide map showing common regions of loss of heterozygosity/allelic imbalance in breast cancer. *Cancer Res*, 60(14):3706–3712, 2000.

[Ore03] M. Oren. Decision making by p53: Life, death and cancer. *Cell Death Differ.*, 10(4):431–442, 2003.

[OVLW04] A. B. Olshen, E. S. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572, 2004.

[PDBSDM05] K. Pelckmans, J. De Brabanter, J. A. Suykens, and B. De Moor. Handling missing values in support vector machine classifiers. *Neural Netw*, 18(5-6):684–692, 2005.

[Pla99a] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning*. MIT-Press using sequential minimal optimization, Cambridge, MA, USA, 1999.

[Pla99b] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In D. Schuurmans A. Smola, B. Schölkopf, editor, *Advances in large margin classifiers*, pages 61–74. MIT Press, 1999.

[Pop72] K. R. Popper. *Objective knowledge*. The Clarendon Press, Oxford, 1972.

[Pop00] N. C. Popescu. Comprehensive genetic analysis of cancer cells. *J Cell Mol Med*, 4(3):151–163, 2000.

[PRM+05] T. S. Price, R. Regan, R. Mott, A. Hedman, B. Honey, R. J. Daniels, L. Smith, A. Greenfield, A. Tiganescu, V. Buckle, N. Ventress, H. Ayyub, A. Salhan, S. Pedraza-Diaz, J. Broxholme, J. Ragoussis, D. R. Higgs, J. Flint, and S. J. Knight. SW-array: A dynamic programming solution for the identification of copy-number changes in genomic DNA using array comparative genome hybridization data. *Nucleic Acids Res*, 33(11):3455–3464, 2005.

[PS91] J. Park and W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246 – 257, 1991.

[PSS+98] D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W. L. Kuo, C. Chen, Y. Zhai, S. H. Dairkee, B. M. Ljung, J. W. Gray, and D. G. Albertson. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nat Genet*, 20(2):207–211, 1998.

[Qui89] J. R. Quinlan. Unknown attribute values in induction. In A. Segre, editor, *Proceedings of the Sixth International Machine Learning Workshop*, Cornell, New York, 1989. Morgan Kaufmann.

[Qui93] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[RHPM04] M. Ruschhaupt, W. Huber, A. Poustka, and U. Mansmann. A compendium to ensure computational reproducibility in high-dimensional classification tasks. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 37, 2004.

[RIF+06] R. Redon, S. Ishikawa, K.R. Fitch, L. Feuk, G.H. Perry, T.D. Andrews, H. Fiegler, M.H. Shapero, A.R. Carson, W. Chen, E.K. Cho, S. Dallaire, J.L. Freeman, J.R. Gonzalez, M. Gratacos, J. Huang, D. Kalaitzopoulos, D. Komura, J.R. MacDonald, C.R. Marshall, R. Mei, L. Montgomery, K. Nishimura, K. Okamura, F. Shen, M.J. Somerville, J. Tchinda, A. Valsesia, C. Woodwark, F. Yang, J. Zhang, T. Zerjal, J. Zhang, L. Armengol, D.F. Conrad, X. Estivill, C. Tyler-Smith, N. P. Carter, H. Aburatani, C. Lee, K. W. Jones, S. W. Scherer, and M.E. Hurles. Global variation in copy number in the human genome. *Nature*, 444(7118):444–454, 2006.

[Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[RJB+02] R. J. Rieker, S. Joos, C. Bartsch, F. Willeke, M. Schwarzbach, M. Otano-Joos, S. Ohl, J. Hogel, T. Lehnert, P. Lichter, H. F. Otto, and G. Mechtersheimer. Distinct chromosomal imbalances in pleomorphic and in high-grade dedifferentiated liposarcomas. *Int J Cancer*, 99(1):68–73, 2002.

[RL03] C. Ruczinski, I. A. K. and M. Leblanc. Logic regression. *Journal of Computational and Graphical Statistics*, 12(3):475–511, 2003.

[RN95] S. Russell and P. Norvig. *Artificial intelligence - a modern approach*. Prentice-Hall, Upper Saddle River, New Jersey, 1995.

[RRB+01] C. Rudlowski, W. Rath, A. J. Becker, O. D. Wiestler, and R. Buttner. Trastuzumab and breast cancer. *N Engl J Med*, 345(13):997–998, 2001.

[RSM87] G. D. Rennels, E. H. Shortliffe, and P. L. Miller. Choice and explanation in medical management: A multiattribute model of artificial intelligence approaches. *Med Decis Making*, 7(1):22–31, 1987.

[RSS05] G. Rätsch, S. Sonnenburg, and B. Schölkopf. Rase: Recognition of alternatively spliced exons in C. elegans. *Bioinformatics*, 21 Suppl 1:i369–i377, 2005.

[Rät01] G. Rätsch. *Robust boosting via convex optimization: Theory and applications*. PhD thesis, Universität Potsdam, 2001.

[SAG+05] I. Scheel, M. Aldrin, I. K. Glad, R. Sorum, H. Lyng, and A. Frigessi. The influence of missing value imputation on detection of differentially expressed genes from microarray data. *Bioinformatics*, 21(23):4272–4279, 2005.

[Sal97] S. L. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1:317–327, 1997.

[Saw03] C. L. Sawyers. Opportunities and challenges in the development of kinase inhibitor therapy for cancer. *Genes Dev*, 17(24):2998–3010, 2003.

[Sch97] B. Schölkopf. *Support vector learning*. PhD thesis, Technische Universität Berlin, 1997.

[Sch04] H. Schwender. Modifying microarray analysis methods for categorical data – SAM and PAM for SNPs. In C. Weihs and W. Gaul, editors, *28th Annual Conference of the Gesellschaft für Klassifikation*, pages 370–377, University of Dortmund, 2004. Springer.

[Sel55] O. Selfridge. Pattern recognition in modern computers. In *Western Joint Computer Conference*, pages 94–97, New York: Institute of Radio Engineers, 1955.

[SFK+05] E. Segal, N. Friedman, N. Kaminski, A. Regev, and D. Koller. From signatures to models: Understanding cancer using microarrays. *Nat Genet*, 37 Suppl:S38–45, 2005.

[SG02] J. L. Schafer and J. W. Graham. Missing data: Our view of the state of the art. *Psychol Methods*, 7(2):147–177, 2002.

[Sha92] J. Shanteau. How much information does an expert use? is it relevant? *Acta Psychologica*, 81:75–86, 1992.

[Sie99] H. T. Siegelmann. *Neural networks and analog computation - beyond the turing limit*. Birkhäuser, Boston, 1999.

[SKRD+03] O. Schmidt-Kittler, T. Ragg, A. Daskalakis, M. Granzow, A. Ahr, T. J. Blankenstein, M. Kaufmann, J. Diebold, H. Arnholdt, P. Muller, J. Bischoff, D. Harich, G. Schlimok, G. Riethmuller, R. Eils, and C. A. Klein. From latent disseminated cells to overt metastasis: Genetic analysis of systemic breast cancer progression. *Proc Natl Acad Sci U S A*, 100(13):7737–7742, 2003.

[SKS+02] C. Schoch, A. Kohlmann, S. Schnittger, B. Brors, M. Dugas, S. Mergenthaler, W. Kern, W. Hiddemann, R. Eils, and T. Haferlach. Acute myeloid leukemias with reciprocal rearrangements can be distinguished by specific gene expression profiles. *Proc Natl Acad Sci U S A*, 99(15):10008–10013, 2002.

[SMF+03] F. Schubert, J. Müller, B. Fritz, P. Lichter, and R. Eils. Understanding the classication of tumors with a support vector machine: A case-based explanation scheme. In H. Mewes, D. Firshman, V. Heun, and S. Kramer, editors, *German Conference on Bioinformatics*, volume I, pages 123–127, Neuherberg/Garching near Munich, 2003. belleville Verlag Michael Farin.

[SMH+05] J. A. Schardt, M. Meyer, C. H. Hartmann, F. Schubert, O. Schmidt-Kittler, C. Fuhrmann, B. Polzer, M. Petronio, R. Eils, and C. A. Klein. Genomic analysis of single cytokeratin-positive cells from bone marrow reveals early mutational events in breast cancer. *Cancer Cell*, 8(3):227–239, 2005.

[Smo98] A. Smola. *Learning with kernels*. PhD thesis, Technische Universität Berlin, 1998.

[SNS+01] A. M. Snijders, N. Nowak, R. Segraves, S. Blackwood, N. Brown, J. Conroy, G. Hamilton, A. K. Hindle, B. Huey, K. Kimura, S. Law, K. Myambo, J. Palmer, B. Ylstra, J. P. Yue, J. W. Gray, A. N. Jain, D. Pinkel, and D. G. Albertson. Assembly of microarrays for genome-wide measurement of DNA copy number. *Nat Genet*, 29(3):263–264, 2001.

[SPST+01] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput*, 13(7):1443–1471, 2001.

[SRS+06] D. E. Stange, B. Radlwimmer, F. Schubert, F. Traub, A. Pich, G. Toedt, F. Mendrzyk, U. Lehmann, R. Eils, H. Kreipe, and P. Lichter. High-resolution genomic profiling reveals association of chromosomal aberrations on 1q and 16p with histologic and genetic subgroups of invasive breast cancer. *Clin Cancer Res*, 12(2):345–352, 2006.

[SS98] A. J. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22:211–231, 1998.

[SSM98] A. J. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, 1998.

[SSS+04] J. Stelling, U. Sauer, Z. Szallasi, F. J. Doyle, and J. Doyle. Robustness of cellular functions. *Cell*, 118(6):675–685, 2004.

[ST95] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In U. M. Fayyad and R. Uthurusamy, editors, *First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, pages 275–281, Montreal, Canada, 1995.

[STJE05] F. Schubert, B. Tausch, S. Joos, and R. Eils. CGH-profiler: Data mining based on genomic aberration profiles. *BMC Bioinformatics*, 6:188, 2005.

[STLS+97] S. Solinas-Toldo, S. Lampel, S. Stilgenbauer, J. Nickolenko, A. Benner, H. Dohner, T. Cremer, and P. Lichter. Matrix-based comparative genomic hybridization: Biochips to screen for genomic imbalances. *Genes Chromosomes Cancer*, 20(4):399–407, 1997.

[TCS+01] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[TS01] V. Tresp and A. Schwaighofer. Scalable kernel systems. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Artificial Neural Networks - ICANN 2001*, pages 285–291. Springer Verlag, 2001.

[TTC01] V. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, 98(9):5116–5121, 2001.

[TW07] R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics*, Advance Access:1–12, 2007.

[Vap95] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, New York, 1995.

[VDWPW04] V. Vanhentenrijk, C. De Wolf-Peeters, and I. Wlodarska. Comparative expressed sequence hybridization studies of hairy cell leukemia show uniform expression profile and imprint of spleen signature. *Blood*, 104(1):250–255, 2004.

[VK04] B. Vogelstein and K. W. Kinzler. Cancer genes and the pathways they control. *Nat Med*, 10(8):789–799, 2004.

[VKM+04] A. Vinayagam, R. Konig, J. Moormann, F. Schubert, R. Eils, K. H. Glatting, and S. Suhai. Applying support vector machines for gene ontology based gene function prediction. *BMC Bioinformatics*, 5:116, 2004.

[Wag01] A. Wagner. How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. *Bioinformatics*, 17(12):1183–1197, 2001.

[WBZ+05] U. Woelfle, E. Breit, K. Zafrakas, M. Otte, F. Schubert, V. Muller, J. R. Izbicki, T. Loning, and K. Pantel. Bi-specific immunomagnetic enrichment of micrometastatic tumour cell clusters from bone marrow of cancer patients. *J Immunol Methods*, 300(1-2):136–145, 2005.

[WCS+03] U. Woelfle, J. Cloos, G. Sauter, L. Riethdorf, F. Janicke, P. Van Diest, R. Brakenhoff, and K. Pantel. Molecular signature associated with bone marrow micrometastasis in human breast cancer. *Cancer Res*, 63(18):5679–5684, 2003.

[WF00] I. J. Witten and E. Frank. *Data mining: Practical machine learning tools and techniques with java implementations.* Morgan Kaufmann Publishers, San Francisco, CA, 2000.

[WF05] H. Willenbrock and J. Fridlyand. A comparison study: Applying segmentation to array cgh data for downstream analyses. *Bioinformatics*, 21(22):4084–4091, 2005.

[WKP+05] P. Wang, Y. Kim, J. Pollack, B. Narasimhan, and R. Tibshirani. A method for calling gains and losses in array cgh data. *Biostatistics*, 6(1):45–58, 2005.

[WLJF06] X. Wang, A. Li, Z. Jiang, and H. Feng. Missing value estimation for dna microarray gene expression data by support

vector regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, 7:32, 2006.

[WLS+01] S. Wessendorf, P. Lichter, C. Schwanen, B. Fritz, M. Baudis, K. Walenta, M. Kloess, H. Dohner, and M. Bentz. Potential of chromosomal and matrix-based comparative genomic hybridization for molecular diagnostics in lymphomas. *Ann Hematol*, 80(Suppl 3):B35–37, 2001.

[WMW+08] B. Wilhelm, S. Marguerat, S. Watt, F. Schubert, V. Wood, I. Goodhead, C.J. Penkett, J. Rogers, and J. Bähler. Dynamic repertoire of a eukaryotic transcriptome surveyed at single nucleotide resolution. *Accepted for publication in Nature*, 2008.

[WMZKM04] J. Wang, L. A. Meza-Zepeda, S. H. Kresse, and O. Myklebost. M-cgh: Analysing microarray-based cgh experiments. *BMC Bioinformatics*, 5(1):74, 2004.

[WSG+03] J. Wiemer, F. Schubert, M. Granzow, T. Ragg, J. Fieres, J. Mattes, and R. Eils. Informatics united: Exemplary studies combining medical informatics, neuroinformatics and bioinformatics. *Methods Inf Med*, 42(2):126–133, 2003.

[WSS+08] U. Woelfle, F. Schubert, R. Segraves, R. Eils, D. Albertson, and K. Pantel. Genome-wide screening for chromosomal imbalances in early breast cancer revealed by array-cgh of archival tissue. *submitted*, 2008.

[XVDL05] B. Xing and M. J. Van Der Laan. A causal inference approach for constructing transcriptional regulatory networks. *Bioinformatics*, 2005.

[YA01] O. T. Yildiz and E. Alpaydin. Omnivariate decision trees. *IEEE Transactions on Neural Networks*, 12(6):1539–1546, 2001.