Regularized Discriminant Analysis Incorporating Prior Knowledge on Gene Functional Groups

Diplomarbeit am Institut für Statistik
der Ludwig-Maximilians Universität München
von
Monika Jelizarow

21. Dezember 2009

Erstgutachterin : **Prof. Dr. Anne-Laure Boulesteix**
Zweitgutachter : **Prof. Dr. Gerhard Tutz**

Acknowledgements

The time I spent for my diploma thesis could not have been such an exciting one if I had no support during the whole period.

First of all I deeply thank my advisor Prof. Dr. Anne-Laure Boulesteix from the Institute of Medical Informatics, Biometry and Epidemiology (IBE) who, from the beginning, provided me help and guidance. She motivated me to enter the field of biostatistics, always being not only an excellent teacher, but also giving me advice and the feeling of being supported.

I am grateful to Dr. Arthur Tenenhaus and Vincent Guillemot who made me feel comfortable during my one month stay at the Department of Signal Processing and Electronic Systems at the École Supérieure d' Électricité (Supélec) in Paris, which was part of this diploma thesis and where I had the possibility to gain insight into biological networks. I thank the Bayerisch-Französisches Hochschulzentrum for the financial support of this project.

Furthermore, I wish to thank Prof. Dr. Thomas Augustin from the Institute of Statistics for the time in his group and especially for his lectures on decision theory, Prof. Dr. Korbinian Strimmer from the Institute of Medical Informatics, Statistics and Epidemiology (IMISE) of the University of Leipzig for his numerous comments and his help at the beginning of the project, Prof. Dr. Ulrich Mansmann for giving me the opportunity to write this thesis at the IBE, Prof. Dr. Gerhard Tutz from the Institute of Statistics for taking over the co-refereeing of this work, Christoph Bernau for helping me to use the CMA package, and Esther Herberich for proof-reading and helpful comments.

Finally, I am grateful to all persons, both from my university and my personal environment, who accompanied my way through the studies of statistics at the Ludwig-Maximilians University of Munich.

Zusammenfassung

Motiviert durch mögliche Anwendungen bei der Tumorklassifikation auf Basis microarray-basierter Daten kehrte die Diskriminanzanalyse im letzten Jahrzehnt als Forschungsgegenstand zurück. Die folgende Arbeit lässt sich durch drei Hauptpunkte charakterisieren. **1.** Wir führen zunächst ein Verfahren zur Schätzung regularisierter Kovarianzmatrizen ein, das auf dem Shrinkage Kovarianzschätzer nach Ledoit und Wolf [31, 33, 32] basiert, zusätzlich jedoch a priori Wissen über die Zugehörigkeit von einzelnen Genen zu bestimmten funktionellen Gruppen aus der Datenbank KEGG integriert. Es wird in dieser Arbeit mit **SHIP** (**SH**rinking and **I**ncorporating **P**rior knowledge) bezeichnet. Für die konkrete Integration von Wissen entwickeln wir mehrere, in ihrem Informationsgehalt unterschiedliche Ansätze. Die optimale Intensität der Schrumpfung wird entgegen der üblichen Prozedur nicht durch Kreuzvalidierung, sondern gemäß Ledoit und Wolf analytisch bestimmt. **2.** Wir schlagen weiterhin eine modifizierte Form der linearen Diskriminanzanalyse (LDA) vor, die den oben genannten regularisierten Kovarianzschätzer technisch einbettet. **3.** Im letzten wesentlichen Teil evaluieren wir die Klassifikationsgenauigkeit der hier eingeführten Methode anhand realer Genexpressionsdaten. Hierbei berücksichtigen wir sowohl den Zwei- als auch den Mehr-Klassen-Fall und wählen zu Vergleichszwecken die diagonale lineare Diskriminanzanalyse sowie die nearest shrunken centroids Methode [15]. Es wird gezeigt, dass die rlda.TG - eine der hier eingeführten Varianten der LDA - insgesamt in allen Klassifikationsproblemen gut abschneidet und die anderen Methoden, wenn auch geringfügig, in manchen Datensituationen übertrifft. Es stellt sich jedoch heraus, dass eine weitere auf dem Kovarianzschätzer nach Ledoit und Wolf basierende Variante der LDA, die *kein* biologisches Wissen integriert, ebenso genau klassifiziert wie die rlda.TG.

Abstract

In the last decade, the renaissance of interest in discriminant analysis has been primarily motivated by possible applications to tumor classification using high-dimensional microarray-based data. In this thesis, we do three things: **1.** First, we introduce a new regularizing covariance estimation procedure we refer to as **SHIP**: **SH**rinking and **I**ncorporating **P**rior knowledge. The resulting covariance estimator is based on the shrinkage estimator by Ledoit and Wolf [31, 33, 32], but additionally incorporates prior knowledge on gene functional groups extracted from the database KEGG. In order to integrate this knowledge into the shrinkage estimator, we develop multiple options. Instead of using a standard cross-validation procedure for determining the optimal shrinkage intensity, we determine it analytically as introduced by Ledoit and Wolf. **2.** Second, we propose a variant of regularized linear discriminant analysis. This method generalizes the idea of the shrinkage estimator from above into the linear discriminant analysis (LDA). **3.** Third, we apply our method to public gene expression data sets and examine the classification performance in both the binary and the c-nary case, where $c > 2$. We choose the diagonal linear discriminant analysis and the nearest shrunken centroids method [15] as competitors. It is shown that the rlda.TG - one of our variants of LDA 'via the SHIP' - performs well in all classification problems and even outperforms, albeit marginally, the competitors in some situations. Unexpectedly, we find that another variant of LDA which is based on the shrinkage estimator by Ledoit and Wolf and which does *not* incorporate any biological knowledge is as competitive as the rlda.TG.

Contents

Chapter 1

Overview

1.1 Introduction

In the last decade, biomedical research has experienced a revival due to microarray technology which allows the measurement of expression levels of thousands of genes simultaneously. During this period, the number of publications within the scope of microarray-based research increased explosively from few hundreds to several thousands per year [36]. Concurrently, however, the so-called 'small n, large p' problem arised. It describes the typical data setting in all applications of microarray technology where the number of variables (genes) p is considerably larger than the number of observations n (chips), hence the term 'high-dimensional'. Since traditional methods often yield deficient results in these high-dimensional data situations or even become inapplicable, it has been a challenging task to develop new adequate methods. As a consequence of both the difficulty of the methodological statistical questions and the uncertainty about the reliability of microarray-based data, statisticians have been split into two camps, the optimistic and the pessimistic one. Those constituting the former camp have often been lead by the objective the biochemist Mark Schena formulated in 2003, namely that '[...] all human illness can be studied by microarray analysis, and the ultimate goal [...] is to develop effective treatments [...] for every human disease by 2050' [39]. On the other hand, John P. A. Ioannidis stated in 2005 that 'Microarrays need evidence and this cannot be obtained from a couple of small studies, no matter how high-tech' [25, 26].

It may has been a disillusioning experience for some of those involved, but has also led to a plethora of new methodological developments. For instance, classification based on high-dimensional gene expression data has been of major interest in cancer research since a precise prediction of tumor classes is essential for successful diagnosis and treatment. A comprehensive review on classification methods using gene expression data is given by Dudoit et al. [11]. In particular, this article includes nearest-neighbor classification methods, classification trees, modern approaches like bagging and boosting and the (regularized) linear discriminant analysis which is still of interest in current research. A crucial property of the latter is that the within-class covariance matrices are assumed to be equal. Moreover, since the linear discriminant analysis encloses the inverse of the covariance matrix in its discriminant function used for classification of an observation to the most likely underlying class, the estimator of the covariance matrix is required to be both invertible and well-conditioned. Traditionally, one employs the pooled empirical covariance matrix as estimator which, however, has undesirable characteristics in the high-dimensional data setting: it is ill-conditioned and singular, thus not invertible. Hence, the objective of regularized linear discriminant analysis is to modify the pooled covariance matrix such that the resulting estimator has the desirable properties from above and yields an accurate classification. Furthermore, an increasingly popular approach is to regularize the within-class covariance by incorporating external biological knowledge on the functions of genes from databases like the **K**yoto **E**ncyclopedia of **G**enes and **G**enomes [28]. While Guillemot et al. [17] and Tai and Pan [46] propose approaches embedding biological knowledge into the regularized linear discriminant analysis, there is a growing number of authors addressing other class prediction methods incorporating biological knowledge, for instance Li and Li [34], Rapaport et al. [14], Binder and Schumacher [5] and Slawski et al. [44].

Especially for scientists who, from the point of view of statistical research, have grown up with microarray-based data, the additional incorporation of recent biological knowledge from databases into statistical methods might be what high-dimensional molecular data once were: a mystery splitting statisticians into two camps.

1.2 Subject of this work

Within the scope of current scientific focus, this thesis deals with a further variant of regularized linear discriminant analysis incorporating biological knowledge on gene functional groups.

It is organized as follows. Chapter 1 completes with an overview of the five microarray gene expression data sets we use throughout this thesis. Chapter 2 presents the scientific scope on which this thesis is built. In particular, we start with explaining the idea behind discriminant analysis and discuss its generalization to the high-dimensional setting, where $n \ll p$. We further address the issue of measuring the prediction accuracy. We give some basic insights into the database KEGG and define what biological knowledge means from this work's perspective. The chapter completes with an outline of existing approaches incorporating prior knowledge into the regularized linear discriminant analysis. In Chapter 3, the main and most extensive part of this thesis, we introduce a new covariance estimation procedure we refer to as **SHIP**: **SH**rinking and **I**ncorporating **P**rior knowledge. The resulting covariance estimator represents the shrinkage estimator introduced by Ledoit and Wolf [31, 33, 32], being enhanced by consideration of prior knowledge on gene functional groups. An important feature of this estimator is that the optimal shrinkage intensity it is based on is determined analytically. This constitutes a clear advantage over common approaches like cross-validation depending on computationally very expensive procedures. We give a detailed derivation of this shrinkage intensity. Chapter 4 addresses a variant of regularized linear discriminant analysis which generalizes the idea of the shrinkage estimator introduced in Chapter 3. We demonstrate in detail how the ideas from Chapter 3 can technically be included into the framework of linear discriminant analysis. We further examine the classification performance of the method proposed in this work using the real-life data presented in Chapter 1. We complete with a summary of the most important results in Chapter 5 and provide an outlook to our future work in this field.

We have implemented the methods proposed in this thesis in the language `R 2.9.1` [47]. In Appendix A, we give an outline of the programming code which can be found in its complete and commented version on the attached CD.

1.3 Real data sets

In this section, we give a brief overview of the five public microarray gene expression data sets we use throughout this thesis. For computational reasons, we do not employ all available genes, but perform a variable selection before. In particular, we use the method `GeneSelection()` of the open source `R` package `CMA` [43]. Since it is not within the scope of this thesis, we do not address this topic in detail. Without any deeper motivation, we thus choose an ordinary two sample t.test (`method="t.test"`) as concrete variable selection method. We generate the learning and test samples by employing the CMA method `GenerateLearningsets()` and use, except where indicated, a stratified five-fold cross-validation (`method="CV", fold=5, niter=10, strat=TRUE`) as evaluation scheme, repeated ten times in order to achieve more stable results [10, 8, 47]. As the number of top genes can vary, we specify it separately in the respective sections.

- **Golub_Merge [16]:**
  The Golub_Merge data set is available from the Bioconductor package `golubEsets` and consists of gene expression intensities for 7 129 genes of 72 different individuals from two cancer classes, 47 with acute lymphoblastic leukemia (ALL) ($\hat{=}$ '0') and 25 with acute myeloid leukemia (AML) ($\hat{=}$ '1').

- **sCLLex (chronic lymphocytic leukemia):**
  The sCLLex data set is available from the Bioconductor package `CLL` and consists of gene expression intensities for 12 625 genes of 22 individuals with chronic lymphocytic leukemia, from which 14 are classified as progressive ($\hat{=}$ '0') and 8 as stable ($\hat{=}$ '1') in regard to disease progression.

- **ALL (Acute Lymphoblastic Leukemia Data):**
  The ALL data set is available from the Bioconductor package `ALL` and consists of gene expression intensities for 12 625 genes of 128 different individuals with acute lymphoblastic leukemia, whereas the classes are built by the type and stage of the disease (five subtypes of B-cell ALL and five subtypes of T-cell ALL, respectively). Thus, the original data set is a ten-class data set which

leads to rather inaccurate within-class estimates. Consequently, the number of classes should be decreased by adequately pooling together the subtypes of both B-cell ALL and T-cell ALL, respectively. For instance, taking together all subtypes yields two classes, where 95 patients are diagnosed B-cell ALL and 33 are diagnosed T-cell ALL. Note that the data sets Golub_Merge and sCLLex are both two-group classification problems, and since it is easier for most classification methods to work well in the binary setting our objective is to find a compromise between two and ten classes, with regard to an accurate evaluation of the classification method proposed in this thesis. The data sets thus obtained are characterized as follows:

ALL_a:
The data set ALL_a consists of gene expression intensities for 12 625 genes of 128 different individuals from 6 cancer classes, 24 with B or B1 B-cell ALL ($\hat{=}$ '0'), 36 with B2 B-cell ALL ($\hat{=}$ '1'), 35 with B3 or B4 B-cell ALL ($\hat{=}$ '2'), 6 with T or T1 T-cell ALL ($\hat{=}$ '3'), 15 with T2 T-cell ALL ($\hat{=}$ '4') and 12 with T3 or T4 T-cell ALL ($\hat{=}$ '5').

ALL_b:
The data set ALL_b consists of gene expression intensities for 12 625 genes of 128 different individuals from 4 cancer classes, 60 with B or B1 or B2 B-cell ALL ($\hat{=}$ '0'), 35 with B3 or B4 B-cell ALL ($\hat{=}$ '1'), 21 with T or T1 or T2 T-cell ALL ($\hat{=}$ '2') and 12 with T3 or T4 T-cell ALL ($\hat{=}$ '3').

ALL_c:
The data set ALL_c consists of gene expression intensities for 12 625 genes of 128 different individuals from 2 cancer classes, 95 with B-cell ALL ($\hat{=}$ '0') and 33 with T-cell ALL ($\hat{=}$ '1').

The following figures illustrate graphically, for each data set, the number of observations in each class of the dependent variable.

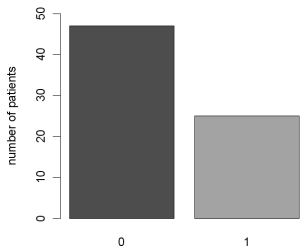

Figure 1.1: Number of observations in each cancer class for the data set Golub_Merge.

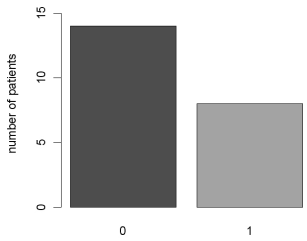

Figure 1.2: Number of observations in each cancer class for the data set sCLLex.

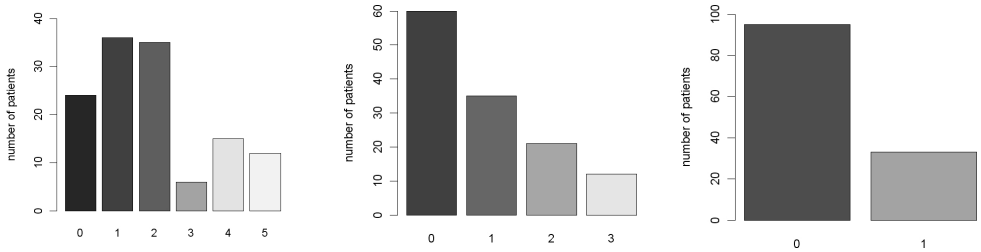

Figure 1.3: Number of observations in each cancer class for the data sets ALL_a, ALL_b and ALL_c.

Chapter 2

Scientific scope

The subject of this work merges traditional methods with very modern ideas and applications resulting from recent technical advances. Hence, it is a challenge to define the scientific scope on which we build our work. In this chapter, we though try to distinguish between essential and negligible information. We start with explaining the idea behind discriminant analysis. Subsequently, we discuss its generalization to the high-dimensional setting, where $n \ll p$. The chapter completes, after a brief introduction to the database KEGG, with an outline of modern approaches incorporating prior knowledge into discriminant analysis.

2.1 Discriminant analysis

The discriminant analysis is a widely used classification method belonging to the supervised learning techniques in the machine learning framework. For the first time, it was introduced by Ronald Aylmer Fisher in 1936. Although, since Fisher's discriminant analysis, a multiplicity of other variants has been developed, it is still of interest in current research. In this thesis, we briefly outline the basic concept behind this classification method, being aware of possible incompleteness because of the large amount of literature on this subject. The explanations in this section are based on Fahrmeir, Hamerle and Tutz [18] and the lecture notes on multivariate statistics by Tutz [48]. In a nutshell, the objective in such classification problems can be

described as follows: suppose there are c different populations which are represented by a finite set of class labels $\{1, ..., c\}$. Let Y be the *stochastic* variable indicating the underlying class, i.e. $Y \in \{1, ..., c\}$. Further, $\mathbb{X}^T = (X_1, ..., X_p)$ denotes the $(1 \times p)$ *stochastic* vector of predictor variables. Consider now a set of *observed* predictors $\boldsymbol{x}^T = (x_1, ..., x_p)$ for each sample of an object with known class y. Let a finite sample of n predictor-class pairs be given, i.e. $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$. Now suppose that we have a new observation given by $(\boldsymbol{x}_{n+1}, Y_{n+1})$, i.e. only the vector $\boldsymbol{x}_{n+1}$ of predictor variables is observed while Y_{n+1} can take values from $\{1, ..., c\}$, thus is unobserved. The question is now how to identify correctly the class from which the new observation comes. Hence, the classification problem consists in finding an accurate classification rule - often denoted as classifier - for the class Y, being based on the given sample of n objects with both observed predictor variables and observed class. Classifiers are thus built from past experience. In the following, we describe in detail how to derive such a classification rule, taking into account the given information. Note that a classification problem in this sense can be interpreted as a prediction problem since the true underlying class is, in fact, predicted.

2.1.1 Bayes classification rule

In the explanations above, we pointed out the objective in a typical classification problem which consists in finding an accurate classification rule for the class Y, where $Y \in \{1, ..., c\}$. Since there is a multiplicity of different classification rules in the literature [48, 18], we focus on the intuitive and widely used *Bayes classification rule* in this thesis. Before explaining the latter in detail we first introduce some essential terms, whereas the assumptions described above hold.

- Prior probabilities:
  The prior probabilities for the particular classes or populations are denoted by $p(r) = P(Y = r)$, where $r = 1, ..., c$.
- Posterior probabilities:
  The posterior probabilities for the particular classes or populations are denoted by $p(r|\boldsymbol{x}) = P(Y = r|\boldsymbol{x})$, where $r = 1, ..., c$. Such a posterior probability is a conditional probability for class r given a vector $\boldsymbol{x}$ of observed predictor

variables. Note that it seems to be obvious to compare these probabilities for the purpose of classification.

- Within-class densities:
  The within-class densities, i.e. the densities of the predictor variables given the underlying class, are denoted by $f(\boldsymbol{x}|1), ..., f(\boldsymbol{x}|c)$, whereas a special distribution has to be specified.

- Mixture density of the population:
  The density of the whole population, that means the density of the predictor variables not separated according to the respective classes, is denoted by $f(\boldsymbol{x}) = p(1)f(\boldsymbol{x}|1) + ... + p(c)f(\boldsymbol{x}|c)$.

Definition 1 (Classification rule) *A classification rule or classifier can be defined as a mapping $\delta(\cdot)$, for which it holds:*

$$\begin{aligned} \delta(\cdot): \mathbb{R}^p &\longrightarrow \{1, ..., c\} \\ \boldsymbol{x} &\longmapsto \delta(\boldsymbol{x}), \end{aligned}$$

where $\{1, ..., c\}$ is a finite set of class labels and $\boldsymbol{x}$ is set of predictor variables, i.e. $\boldsymbol{x}^T = (x_1, ..., x_p)$.

A basic classification rule is the Bayes classification rule which has the following form:

$$\delta^*(\boldsymbol{x}) = r \iff p(r|\boldsymbol{x}) = \max_{i=1,...,c} p(i|\boldsymbol{x}). \tag{2.1}$$

Thus it appears that, for given $\boldsymbol{x}$, the class is chosen for which the posterior probability is maximal. However, for instance if the posterior probabilities are not available, alternative forms of the Bayes classification rule can be formulated by using the prior and the within-class predictor densities. For this purpose, it is helpful to consider the Bayes classification rule as a maximizer of discriminant functions [48]. Let for each $\boldsymbol{x}$ the functions $d_r(\boldsymbol{x})$, $r = 1, ..., c$, measure the 'plausibility' that observation

$\boldsymbol{x}$ comes from class r. By using $d_r(\boldsymbol{x}) = p(r|\boldsymbol{x})$ we obtain the following notation for the Bayes classification rule:

$$\delta^*(\boldsymbol{x}) = r \iff d_r(\boldsymbol{x}) = \max_{i=1,...,c} d_i(\boldsymbol{x}). \tag{2.2}$$

Note that the functions $d_r(\boldsymbol{x})$, where $r = 1, ..., c$, are called *discriminant functions*. Alternative formulations may be obtained by using the Bayes theorem, which has the form:

$$p(r|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|r)p(r)}{f(\boldsymbol{x})} = \frac{f(\boldsymbol{x}|r)p(r)}{\sum_{i=1}^{c} f(\boldsymbol{x}|i)p(i)}. \tag{2.3}$$

According to Eq. 2.3, it follows directly for the comparison of two different classes r and s:

$$p(r|\boldsymbol{x}) \geq p(s|\boldsymbol{x})$$

$$\Leftrightarrow \quad \frac{f(\boldsymbol{x}|r)p(r)}{f(\boldsymbol{x})} \geq \frac{f(\boldsymbol{x}|s)p(s)}{f(\boldsymbol{x})} \tag{2.4}$$

$$\Leftrightarrow \quad f(\boldsymbol{x}|r)p(r) \geq f(\boldsymbol{x}|s)p(s) \tag{2.5}$$

$$\Leftrightarrow \; log(f(\boldsymbol{x}|r)) + log(p(r)) \geq log(f(\boldsymbol{x}|s)) + log(p(s)). \tag{2.6}$$

Thus it appears that the maximization of $p(r|\boldsymbol{x})$ over the classes $r = 1, ..., c$ can furthermore be obtained by maximization of the discriminant functions employed in Eq. 2.4, 2.5 and 2.6. The different forms may be used in order to emphasize different aspects of the Bayes classification rule as well as for simplification reasons. For instance, the logarithmic form in Eq. 2.6 is rather beneficial in the case of normally distributed predictors since it simplifies the Bayes classification rule in a crucial way. We will deal with this aspect in 2.1.2.

2.1.2 Bayes classification rule with normally distributed predictors

In 2.1.1, we have studied the Bayes classification rule in general. Hence, we did not specify any concrete within-class distribution $f(\boldsymbol{x}|r)$, $r = 1, ..., c$, in order to determine the posterior probability $p(r|\boldsymbol{x})$, $r = 1, ..., c$, or one of the equivalent discriminant functions described by Eq. 2.4, 2.5 and 2.6. In the following, however, we assume normally distributed predictor variables $\boldsymbol{x}^T = (x_1, ..., x_p)$, i.e. we assume $\boldsymbol{x}|Y = r \sim N(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$. The within-class densities thus have the form:

$$f(\boldsymbol{x}|r) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_r|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_r)^T \boldsymbol{\Sigma}_r^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_r)\right), \tag{2.7}$$

where $\boldsymbol{\mu}_r$ is the $(p \times 1)$ mean vector for class r, $r = 1, ..., c$, and $\boldsymbol{\Sigma}_r$ denotes the $(p \times p)$ covariance matrix for class r, $r = 1, ..., c$.

As pointed out in 2.1.1, the discriminant function $d_r(\boldsymbol{x}) = \log(f(\boldsymbol{x}|r)) + \log(p(r))$ simplifies considerably the Bayes classification rule in the context of normally distributed predictors. Moreover, we distinguish between two different assumptions concerning the within-class covariance $\boldsymbol{\Sigma}_r$, which results in two variants of discriminant analysis, namely the *linear* and the *quadratic discriminant analysis (LDA and QDA, respectively).*

- **Homogeneous case**

  The homogeneous case implies equivalent within-class covariance matrices, i.e. $\boldsymbol{x}|Y = r \sim N(\boldsymbol{\mu}_r, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = ... = \boldsymbol{\Sigma}_c$. Note that we consider the logarithmic discriminant function specified above by Eq. 2.6. By employing the within-class density from Eq. 2.7 and by leaving out irrelevant terms, we obtain the following discriminant function:

  $$d_r(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_r)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_r) + \log(p(r)). \tag{2.8}$$

  Furthermore, it is of interest how the maximization of $d_r(\boldsymbol{x})$ discriminates between two arbitrary classes r and s. Let us consider these two classes. We then

obtain, after simple rearrangements, the following expression for the difference between the respective discriminant functions:

$$\begin{aligned} d_r(\boldsymbol{x}) - d_s(\boldsymbol{x}) &= \underbrace{-\frac{1}{2}\boldsymbol{\mu}_r^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_r + \frac{1}{2}\boldsymbol{\mu}_s^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_s + \log\left(\frac{p(r)}{p(s)}\right)}_{:=\boldsymbol{\beta}_{0rs}} \\ &\quad + \boldsymbol{x}^T \underbrace{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_r - \boldsymbol{\mu}_s)}_{:=\boldsymbol{\beta}_{rs}} \\ &= \boldsymbol{\beta}_{0rs} + \boldsymbol{x}^T\boldsymbol{\beta}_{rs}. \end{aligned} \tag{2.9}$$

Thus it appears that the classification rule is linear since we prefer class r over class s if $\boldsymbol{\beta}_{0rs} + \boldsymbol{x}^T\boldsymbol{\beta}_{rs} \geq 0$. More precisely, this leads us to the *linear discriminant analysis (LDA)*, where the linearity results from the assumption of equal covariance matrices. In this thesis, we solely constrain our attention on this variant of discriminant analysis. Note that in reality, both $\boldsymbol{\mu}_r$ and $\boldsymbol{\Sigma}$ are unknown and thus have to be estimated from the sample, which yields an estimated classification rule or discriminant function, i.e. $\hat{\delta}^*(\boldsymbol{x}) = r \iff \hat{d}_r(\boldsymbol{x}) = \max\limits_{i=1,\ldots,c} \hat{d}_i(\boldsymbol{x})$. Note further that the priors $p(r)$ may be replaced by the proportion $\hat{p}(r) = \frac{n_r}{n}$. Hence, in order to obtain such an estimated discriminant function from above, we replace $\boldsymbol{\mu}_r$ and $\boldsymbol{\Sigma}$ in Eq. 2.8 by the following estimators:

$$\hat{\boldsymbol{\mu}}_r = \bar{\boldsymbol{x}}_r = \frac{1}{n_r}\sum_{k=1}^{n_r}\boldsymbol{x}_{rk}, \tag{2.10}$$

$$\hat{\boldsymbol{\Sigma}} = \mathbf{S}_{\mathbf{pool}} = \frac{1}{n-c}\sum_{r=1}^{c}\sum_{k=1}^{n_r}(\boldsymbol{x}_{rk} - \bar{\boldsymbol{x}}_r)(\boldsymbol{x}_{rk} - \bar{\boldsymbol{x}}_r)^T, \tag{2.11}$$

where n_r : number of observations in class r, $r = 1, ..., c$, where $\sum_{r=1}^{c} n_r = n$

$\bar{\boldsymbol{x}}_r$: $(p \times 1)$ mean vector for class r, $r = 1, ..., c$

$\boldsymbol{x}_{rk}$: $(p \times 1)$ vector of predictor variables corresponding to the k-th observation in class r, $r = 1, ..., c$

$\mathbf{S}_{\mathbf{pool}}$: $(p \times p)$ pooled empirical covariance matrix.

- **Heterogeneous case**

  In the heterogeneous case, in contrast to the homogeneous one, differing within-class covariance matrices are allowed. Thus it holds $\boldsymbol{x}|Y = r \sim N(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$. Note that here, the classification rule does not simplify to a linear form since the difference between the discriminant functions $d_r(\boldsymbol{x})$ and $d_s(\boldsymbol{x})$ contains both quadratic terms $x_1^2, ..., x_p^2$ and interaction terms $x_i x_j$, where $i, j = 1, ..., p$, $i \neq j$. Hence, this yields the *quadratic discriminant analysis (QDA)*. Although the latter offers more flexibility, it is not widely applied because of the multiplicity of parameters to be estimated, which often results in a poor performance in the case of small sample sizes. Since in this work, our main focus will be the linear discriminant analysis, we do not address this topic in detail. For further reading we refer especially to [18].

Additionally, without giving detailed explanations, we want to point out two further variants of discriminant analysis, namely the *diagonal linear discriminant analysis (DLDA)* and the *nearest shrunken centroids method (NSC)*. The former assumes equal diagonal within-class covariances, whereas the latter can be interpreted as a variant of the diagonal linear discriminant analysis, in which only the most relevant variables contribute to classification by identifying subsets of variables that best characterize each class [15].

2.1.3 Measuring the prediction accuracy

Having studied the question of identifying the underlying class of a new observation, we now concentrate on how to measure the prediction accuracy of an estimated classification rule which is a widely discussed topic in the literature. For example, a comprehensive review on this topic is given by Boulesteix et al. [9]. Nevertheless, we do not address this topic to its full extent, but give a brief outline concerning the prediction measures employed in this thesis. Note that for investigating the performance of a classification rule it is essential to distinguish between prediction measures based on the learning sample and prediction measures based on new observations, i.e. the test sample. More precisely, the terms can be explained as follows. The *learning* or *training set* is denoted by $L = \{(y_k, \boldsymbol{x}_k), k = 1, ..., n\}$, from which the classification rule is derived. The *test set* $T = \{(y_k, \boldsymbol{x}_k), k = 1, ..., n_T\}$ is a sample of new observations, being used in order to assess the performance of the classification rule. Usual prediction measures are empirical error rates, whereas some of these error rates have the drawback of being based on the learning sample. Accordingly, the learning sample is used twice, the first time for the derivation of the classification rule and the second time for the evaluation of its accuracy. As a result, such empirical error rates tend to be underestimated, thus have a negative bias. Further, choosing a classification rule based on the learning sample may lead to an overfitting of the sample, which often results in a poor performance on independent data. Alternatively, the so-called *empirical test error* can be employed, which has the following form:

$$\hat{\epsilon}_{Test}(\hat{\delta}) \quad = \quad \frac{1}{n_T} \sum_{(y_k,\, \boldsymbol{x}_k) \in\, T} I(y_k \neq \hat{\delta}(\boldsymbol{x}_k)), \tag{2.12}$$

where n_T is the number of observations in the test sample T. Note that here, the derivation of the classification rule and the evaluation of the accuracy are carried out by means of **different** samples since the data are split into a learning and a test set. Although the empirical test error is a popular prediction measure, it may sometimes be rather unsuitable [9]. Employing the test error as defined in Eq. 2.12, we implicitly consider all misclassifications symmetrically and hence the corresponding costs to be equal. This point of view, however, is not adequate in all cases. Note

that the term 'costs' has to be considered from a decision theoretic perspective and thus does not only comprise monetary costs. Let us now consider two cancer classes indicating the stage of the disease, e.g. 0 ($\hat{=}$ early stage of the disease) and 1 ($\hat{=}$ advanced stage of the disease). Let us further consider two therapies T_0 and T_1 especially developed for class 0 and 1, respectively. If a patient from class 0 is incorrectly classified as a patient from class 1, the costs of misclassification could be severe side-effects of a useless therapy and the monetary costs for this therapy. On the other hand, if a patient from class 1 is incorrectly classified to belong to class 0, this patient is not medicated effectively like in the first scenario, but the costs of misclassification might be impairment or even the patient's death. Thus it appears that it might be beneficial to consider the misclassifications asymmetrically which leads us to the terms *sensitivity* and *specificity*. Relating to the settings above, the sensitivity of a classification rule is the probability of correctly identifying a patient from class 1. It can be estimated by the proportion of observations from the test set that are correctly classified to class 1:

$$\hat{s}_{Test}(\hat{\delta}) \quad = \quad \frac{\sum_{(y_k,\, \boldsymbol{x}_k)\in\, T} I(y_k = 1) \cdot I(\hat{\delta}(\boldsymbol{x}_k) = 1)}{\sum_{(y_k,\, \boldsymbol{x}_k)\in\, T} I(y_k = 1)}. \tag{2.13}$$

The specificity is the probability of correctly identifying a patient from class 0 and can be estimated by the proportion of observations from the test set that are correctly classified to class 0:

$$\hat{sp}_{Test}(\hat{\delta}) \quad = \quad \frac{\sum_{(y_k,\, \boldsymbol{x}_k)\in\, T} I(y_k = 0) \cdot I(\hat{\delta}(\boldsymbol{x}_k) = 0)}{\sum_{(y_k,\, \boldsymbol{x}_k)\in\, T} I(y_k = 0)}. \tag{2.14}$$

Hence, the calculation procedure for the three prediction measures follows the intention to separate model selection and model evaluation. However, the fact that only a subset of the data determines the classification rule could be seen as a drawback, which leads us to the *K-fold cross-validation*. For simplicity's sake, we consider the empirical test error in the following, but the same principles hold for the other prediction measures such as sensitivity or specificity. In K-fold cross-validation the data of the learning set is split into K parts of roughly equal size. Let $T_1, ..., T_K$,

where $T_1 \cup T_2 \cup \cdots \cup T_K = L$, denote the disjoint partition of the learning sample. W.l.o.g. consider T_1. Then the classification rule is derived from $T_2 \cup \cdots \cup T_K$ and the empirical test error rate is computed using T_1. This procedure is carried out for $m = 1, ..., K$, yielding the K-fold cross-validation error:

$$\hat{\epsilon}_{KCV}(\hat{\delta}) = \frac{1}{n} \sum_{m=1}^{K} \sum_{(y_k,\, \boldsymbol{x}_k) \in\, T_m} I(y_k \neq \hat{\delta}_{\setminus m}(\boldsymbol{x}_k)), \tag{2.15}$$

where $\hat{\delta}_{\setminus m}$ means that the classification rule is estimated without part T_m. Accordingly, in the extreme case it holds $K = n$, which is known as *leave-one-out cross-validation*. Note that, by using K-fold cross-validation, it is possible to obtain improved estimates. While this is not necessarily of relevance if $p \ll n$, it becomes beneficial in the inverse case, i.e. if $n \ll p$. We will deal with this aspect in 2.2.1.

2.2 Discriminant analysis in the high-dimensional setting

In the previous section, we have discussed the discriminant analysis, being especially interested in the linear variant (LDA) which results from the assumption of equal within-class covariance matrices. Starting from this assumption, we now address the linear discriminant analysis in the high-dimensional setting, thus for $n \ll p$, where p is the number of variables and n is the number of observations. In this thesis, we work with high-dimensional microarray gene expression data as described in 1.3. Note that, henceforth, we concentrate solely on the linear discriminant analysis and leave the other variants for further research. In the following, we first analyze the diverse methodological challenges emerging if $n \ll p$. Subsequently, we present the idea behind the approaches coping with high-dimensionality. Both topics are depicted briefly in order to provide a superficial insight into the crucial methodological questions in the $n \ll p$ setting. By far more detailed information and illustrations are given in Chapter 3, where we first detach our explanations from the special case of linear discriminant analysis in order to present a general framework.

2.2.1 Methodological challenges

The linear discriminant analysis discussed in 2.1 can be applied in a straightforward way in the $p \ll n$ case, i.e. if the number of predictor variables does not exceed the number of observations. In the high-dimensional setting, however, using this method is associated with undesirable characteristics of the resulting covariance estimator, which traditionally is the pooled empirical $(p \times p)$ covariance matrix $\mathbf{S_{pool}}$ as denoted by Eq. 2.11. In particular, $\mathbf{S_{pool}}$ is singular and cannot be inverted. Thus, the linear discriminant analysis in its known nature turns out to be inapplicable if $n \ll p$. Therefore, a modified version of the original linear discriminant analysis has to be applied to circumvent these difficulties. In order to resolve the singularity problem, we 'regularize' $\mathbf{S_{pool}}$ according to the shrinkage principle described in 2.2.2.

Moreover, since matrix operations become very extensive due to high-dimensionality, it is worthwhile to simplify the computation of the modified discriminant function which results from employing the regularized covariance estimator. By means of the *singular value decomposition (SVD)* it is possible to compute the inverse of a matrix in an efficient way. It can be shown for the $n \ll p$ case that, by applying the singular value decomposition to a $(p \times p)$ matrix, a $(n \times n)$ matrix remains to be inverted. The interested reader is suggested to study Hastie, Tibshirani and Friedman [23] for more details about the algorithm. In this thesis, however, we do not address this topic in a precise way since our main interest focuses on finding a covariance estimator being both invertible and well-conditioned. Note that we examine this topic in Chapter 3.

Additionally, besides the construction of a classification rule in the high-dimensional case the estimation of its prediction accuracy demands further considerations. By increasing the size of the learning set L the constructed classification rule can usually be improved. On the other hand, the reliability of its evaluation decreases. Conversely, increasing the size of the test set T leads to an improvement of the accuracy estimation. However, as a negative result one typically obtains poorly performing classification rules [43]. In 2.1.3, we discussed prediction measures and pointed out that the K-fold cross-validation error $\hat{\epsilon}_{KCV}$ is more adequate than the empirical test error $\hat{\epsilon}_{Test}$ in the case of small sample sizes. The underlying motivation is to reduce the error estimator's variance which is achieved by averaging. Note that

the method `GenerateLearningsets()` from the open source `R` package `CMA` allows the choice between several techniques generating L and T. One of these techniques is the K-fold cross-validation as described in 2.1.3 by Eq. 2.15. Braga-Neto and Dougherty [10] recommend to repeat the whole procedure several times in order to obtain more stable results. Thus, the results obtained for several different partitions are proposed to be averaged. The corresponding technique generating L and T, respectively, can be carried out by including the argument `niter` into the method `GenerateLearningsets()` [43].

2.2.2 Shrinkage based approaches

Let us consider $\mathbf{S}_{\mathbf{pool}}$. As indicated above, this covariance estimator generally employed in linear discriminant analysis is singular and cannot be inverted. For this reason, the *regularized linear discriminant analysis* has become an established method with its distinct advantage being the applicability for $n \ll p$. However, the term 'regularized discriminant analysis' does not pinpoint one special technique, but is rather a superordinate concept for a multiplicity of methods. For instance, in 1989 Jerome H. Friedman published a seminal work on regularized (Fisher's) discriminant analysis, which we recommend to the interested reader [19]. Nowadays, numerous related methods exist, from which we want to emphasize the *shrunken centroids regularized discriminant analysis (SCRDA)* by Guo et al. [22], a further development of the nearest shrunken centroids mentioned in 2.1.2. The property these methods share is that they are based on the *shrinkage principle* outlined as follows.

The shrinkage principle has a long history in statistics, albeit it is often regarded as a new technique due to its successful application during the last ten years in the context of microarray data analysis. For the first time, it was introduced by Professor Charles Stein of Stanford University in 1955 in the context of estimating the mean vector of a multivariate normal distribution [45]. In 1977, Efron and Morris published a worth reading non-technical primer on shrinkage using a real-life setting of baseball batting averages [13] which we recommend to both statisticians and non-statisticians since it conveys the idea behind the shrinkage principle in an unique way. The main statement of the concept can be depicted in a few words: by properly 'combining' two extreme estimators it is possible to obtain an estimator that outper-

forms either of the extreme ones both in terms of accuracy and statistical efficiency. The 'combined' estimator, hence, dominates the two individual estimators, i.e. the individual estimators are inadmissible from a decision theoretic point of view [4]. 'Combining', in this context, is meant as follows. Instead of choosing between one of these two extreme estimators, the shrinkage approach suggests to build a weighted average of them. As a result, we both resolve the singularity problem and stabilize the covariance estimator, thus its variance.

In the following, we briefly describe the shrinkage estimator for the covariance matrix. In particular, a shrinkage estimator consists of an estimator with no structure, an estimator with a lot of structure and a shrinkage intensity $\lambda \in [0, 1]$ which, intuitively, measures the weight given to the structured estimator. As a result, the estimator with no structure is 'shrunken' towards the structured estimator. The latter contains relatively little estimation error, but is usually misspecified and biased, whereas the former has a lot of estimation error, but is unbiased. Hence, the shrinkage principle responds to the fundamental question of the optimal trade-off between bias and estimation error. More concretely, we will study this topic in Chapter 3. Note that if $\lambda = 1$ the shrinkage estimator equals the highly structured estimator. If $\lambda = 0$ the unstructured estimator is recovered. Note further that, in general, it is possible to combine more than two estimators, for instance see Tai and Pan [46]. In this thesis, however, except in 2.4 where we present the work by Tai and Pan, we focus on shrinkage estimators which result from combining solely two estimators.

2.3 Prior knowledge on gene functional groups: the database KEGG

An increasingly popular approach is to regularize the within-class covariance by incorporating prior biological knowledge from databases. For instance, the **K**yoto **E**ncyclopedia of **G**enes and **G**enomes [28] is a freely available database of biological systems; it itself consists of several databases, each providing special information about biological and chemical objects. KEGG PATHWAY as one of these databases contains a collection of pathway maps representing the current knowledge on molecular interaction and reaction networks for metabolism, various cellular processes and

human diseases. More precisely, KEGG PATHWAY represents pathways as graphs in which the edges stand for the chemical reactions or relations and the vertices stand for the genes taking part in these reactions or relations. Prior biological information is thus encoded by graphs. Here, when talking about 'gene functional groups', we mean special sets of related genes, i.e. a KEGG pathway forms a gene functional group. Note that we define the congruency of a KEGG pathway and a gene functional group in this work, which corresponds to Tai and Pan [46]. Therefore, for each gene expression data set that is compatible with KEGG PATHWAY, it is possible to pinpoint which gene occurs in which functional group. Note that the denomination of such gene functional groups containing information on molecular interaction for human beings begins with 'hsa', which stands for *homo sapiens*.

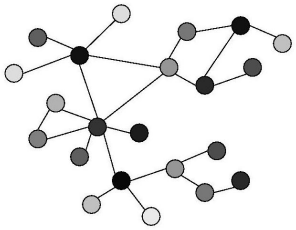

Figure 2.1: A fictional example graph or gene functional group, respectively.

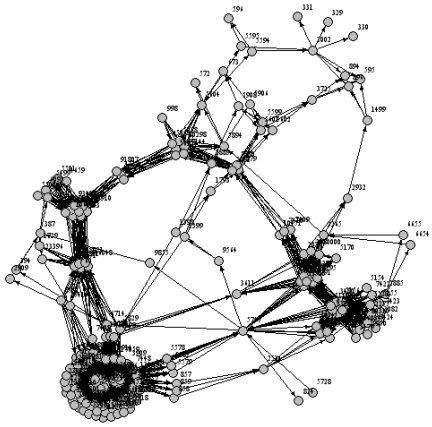

Figure 2.2: Graphical representation of the real KEGG pathway hsa04510: The graph consists of 203 vertices and 1906 edges.

2.4 Approaches incorporating prior knowledge into discriminant analysis

Let us consider the explanations given in the previous section. We now briefly depict selected approaches incorporating prior biological knowledge on gene functional groups into regularized linear discriminant analysis, i.e. the high-dimensional case is addressed. Besides, a multiplicity of other class prediction methods incorporating biological knowledge has been proposed, for instance by Li and Li [34], Rapaport et al. [14], Binder and Schumacher [5] and Slawski et al. [44]. In the latter, for example, the authors propose an extend of the elastic net [50] using biological knowledge on association structures of features. The motivation behind all these approaches is to improve both the prediction accuracy and the results' interpretability. In particular, we first present an outline of the regularized linear discriminant analysis version by Guillemot et al. [17]. Subsequently, we summarize an approach by Tai and Pan [46], which constitutes the starting point with regard to our own idea presented at the end of this section. While Guillemot et al. base their work on Fisher's discriminant analysis, Tai and Pan apply the Bayes classification rule. However, apart from the fact that these two variants correspond for $c = 2$ classes and the assumption of normally distributed predictors, we mainly constrain our attention on the following two aspects. First, we want to study how, in these two approaches, the pooled empirical covariance matrix $\mathbf{S}_{\mathbf{pool}}$ is regularized, i.e. towards which estimator it is shrunken. Second, we are interested in how prior biological knowledge is incorporated into the regularization or shrinkage process.

2.4.1 Guillemot et al.

The *graph-constrained discriminant analysis (gCDA)* proposed by Guillemot et al. in 2008 integrates prior information from graphs into the classification algorithm. Note that, in this approach, the respective gene functional groups are not differentiated and do not need to be. In the linear version of gCDA, Guillemot et al. assume the availability of one single graph, including preferably all variables (genes) from the given data set. Since detailed knowledge on the connectivity between the graph's vertices is extracted, we first introduce the essentially relevant definitions [17]:

Definition 2 (Graph) *A graph $\mathcal{G}$ is defined by a set of edges $\mathcal{E}$ and a set of vertices or nodes $\mathcal{V}$, i.e. $\mathcal{G}$ can be written as follows:*

$$\mathcal{G} := (\mathcal{V}, \mathcal{E}).$$

Definition 3 (Connectivity degree) *Let us consider Definition 2. Let w be the mapping $w: \mathcal{V} \times \mathcal{V} \to \{0, 1\}$, where it holds for $i, j = 1, ..., p$:*

$$w_{ij} = \begin{cases} 1 & \text{if there exists an edge between vertex } i \text{ and vertex } j \\ 0 & \text{otherwise.} \end{cases}$$

For each vertex i of $\mathcal{V}$, the connectivity degree d_i is defined as the cardinality of the set of vertices in $\mathcal{V}$ being connected to i.

Definition 4 (Laplacian matrix) *The Laplacian matrix $\mathcal{L}_\mathcal{G}$ is a matrix representation of a graph as defined in Definition 2. In particular, $\mathcal{L}_\mathcal{G}$ is a positive semi-definite $p \times p$ matrix whose entries are:*

$$l_{\mathcal{G}\,i,j} = \begin{cases} -w_{ij} & \text{if } i \neq j \\ d_i & \text{if } i = j, \end{cases}$$

where - according to Definition 3 - each null term corresponds to an absence of an edge between two vertices in $\mathcal{G}$.

Based on the terms defined above, the approach by Guillemot et al. can be explained as follows. As described in 2.3, graphs are represented by edges which stand for the chemical reactions or relations and by vertices which stand for the genes taking part in these reactions or relations. In a nutshell, each vertex represents a variable (gene)

and the absence of an edge between two vertices indicates that the two variables are independent - and thus uncorrelated - given the remaining variables. In order to describe this in a more statistical framework, Guillemot et al. consider the following natural property of the inverse covariance matrix $\mathbf{\Sigma}^{-1}$ of normally distributed predictor variables $\boldsymbol{x}^T = (x_1, ..., x_p)$:

$$x_i \perp x_j \mid \{x_l, l \in \{1, ..., p\} \backslash \{i, j\}\}$$

$$\Leftrightarrow \qquad \sigma_{i,j}^{-1} \; = \; 0.$$

Thus it appears that an estimator of the inverse covariance matrix, which is used in the linear discriminant analysis, can be derived from the Laplacian matrix $\mathcal{L}_{\mathcal{G}}$ of a prior graph $\mathcal{G}$. Guillemot et al. propose to consider $\mathcal{L}_{\mathcal{G}}$ as the matrix towards the empirical covariance matrix $\mathbf{S_{pool}}$ is shrunken. As pointed out in Definition 4, however, $\mathcal{L}_{\mathcal{G}}$ is positive semi-definite. It is though not surprising that the highly structured estimator should be positive definite since, otherwise, regularization in terms of resolving the singularity problem and stabilizing the covariance estimator turns out to be impossible. Hence, Guillemot et al. circumvent this problem by adding a small positive constant, i.e. $\epsilon > 0$, on the diagonal of $\mathcal{L}_{\mathcal{G}}$. Thus it follows:

$$\begin{aligned} \hat{\mathbf{\Sigma}}_{\mathcal{G}}^{-1} \quad &= \quad \mathcal{L}_{\mathcal{G}} + \epsilon \mathbf{I} \\ \Leftrightarrow \quad \hat{\mathbf{\Sigma}}_{\mathcal{G}} \quad &= \; (\mathcal{L}_{\mathcal{G}} + \epsilon \mathbf{I})^{-1}, \end{aligned} \tag{2.16}$$

where $\mathbf{I}$ is the $(p \times p)$ identity matrix. Then it holds for the regularized covariance estimator:

$$\hat{\mathbf{\Sigma}} \underbrace{=}_{\text{Eq. 2.16}} \lambda \mathbf{S_{pool}} + (1 - \lambda)(\mathcal{L}_{\mathcal{G}} + \epsilon \mathbf{I})^{-1}, \tag{2.17}$$

where $\lambda \in [0, 1]$ denotes the shrinkage intensity, being determined by a cross-validation procedure [17].

2.4.2 Tai and Pan

In order to understand the approach introduced by Tai and Pan in 2007, it is essential to comprehend the idea behind both the nearest shrunken centroids method (NSC) [15], often referred to as predictive analysis of microarrays (PAM), and the shrunken centroids regularized discriminant analysis (SCRDA) [22], which is a further development of the former. Therefore, we first briefly outline these two approaches without claiming completeness. The reader who is familiar with these methods is recommended to skip the respective explanations. We point out that, in large parts, the following two paragraphs are adopted from [15, 22].
Let us first list the notations, where $i = 1, ..., p$ and $k = 1, ..., n$, being essential for the terms defined in NSC and SCRDA, respectively.

n_r	:	number of observations in class r, $r = 1, ..., c$, where $\sum_{r=1}^{c} n_r = n$
x_{ki}	:	k-th observation of the variable (gene) X_i
$\bar{x}_i$	:	i-th component of the overall centroid (overall mean), where $\bar{x}_i = \frac{1}{n} \sum_{k=1}^{n} x_{ki}$
x_{rki}	:	k-th observation of the variable (gene) X_i in class r
$\bar{x}_{ri}$	:	i-th component of the centroid (mean) for class r, where $\bar{x}_{ri} = \frac{1}{n_r} \sum_{k=1}^{n_r} x_{rki}$
s_i	:	pooled standard deviation of the variable (gene) X_i, i.e. $s_i = \sqrt{s_{ii}^2}$ and s_{ii}^2 is the i-th diagonal entry of the $(p \times p)$ pooled empirical covariance matrix $\mathbf{S_{pool}}$

Nearest shrunken centroids (NSC)

In microarray analysis, a general assumption is that most genes do not have differential expression levels among the classes and the differences we observe result from random fluctuations. The nearest shrunken centroids method introduced by Tibshirani et al. in 2002 removes the noisy information arising from such fluctuations by setting a soft threshold, which effectively eliminates a lot of non-contributing genes. In particular, Tibshirani et al. shrink the class centroids (class means) towards the overall centroid (overall mean) after standardizing by the within-class standard deviation for each gene. This standardization has the effect of giving higher weight to

the genes whose expression is stable within the observations of the same class. Note that the class centroids of each gene are shrunken individually, i.e. the genes are assumed to be independent and thus uncorrelated of each other. This, however, is not adequate in the majority of the cases, but will not be considered further in this paragraph.

Let now $\boldsymbol{x}^* = (x_1^*, ..., x_p^*)^T$ be the $(p \times 1)$ vector of predictor variables of a new observation, where x_i^* is the i-th component of $\boldsymbol{x}^*$, $i = 1, ..., p$. Let further be $\tilde{\bar{x}}_{ri}$ the i-th component of the shrunken centroid (mean) $\tilde{\bar{\boldsymbol{x}}}_r$ for class r, i.e. $\tilde{\bar{x}}_{ri}$ is the shrunken centroid of class r for gene i. The shrinkage Tibshirani et al. use is called 'soft thresholding' and works as follows:

$$\tilde{\bar{x}}_{ri} \quad = \quad sgn(\bar{x}_{ri})(|\bar{x}_{ri}| - \Delta)_+, \tag{2.18}$$

where $+$ is the positive part and Δ is a threshold which plays the role of the shrinkage parameter, being determined by cross-validation. Thus it appears from Eq. 2.18 that each $\bar{x}_{ri}$ is reduced by an amount Δ in the absolute value and is set to zero if its absolute value is smaller than zero. Since, thereby, non-contributing genes are eliminated this method is often regarded as variable selection procedure.
Having shrunken the class centroids of the particular genes i, where $i = 1, ..., p$, the gene-specific score for an observation $\boldsymbol{x}^* = (x_1^*, ..., x_p^*)^T$ can be computed. It holds for its i-th component:

$$d_{ri}(x_i^*) \quad = \quad \frac{(x_i^* - \tilde{\bar{x}}_{ri})^2}{2s_i^2} \quad = \quad \frac{(x_i^*)^2}{2s_i^2} - \frac{x_i^* \tilde{\bar{x}}_{ri}}{s_i^2} + \frac{(\tilde{\bar{x}}_{ri})^2}{2s_i^2}. \tag{2.19}$$

Thus the new observation $\boldsymbol{x}^*$ is classified to class r if for class r the sum of the scores over all genes is minimized, i.e.:

$$\boldsymbol{x}^* \in \text{class } r \quad \Leftrightarrow \; d_r(\boldsymbol{x}^*) = \min_{r'=1,\ldots,c} \sum_{i=1}^{p} d_{r'i}(x_i^*) - \log(\hat{p}(r')) \tag{2.20}$$

$$\Longleftrightarrow$$

$$\boldsymbol{x}^* \in \text{class } r \quad \Leftrightarrow \; d_r(\boldsymbol{x}^*) = \min_{r'=1,\ldots,c} \left(\boldsymbol{x}^* - \tilde{\bar{\boldsymbol{x}}}_{r'}\right)^T \hat{\mathbf{D}}^{-1} \left(\boldsymbol{x}^* - \tilde{\bar{\boldsymbol{x}}}_{r'}\right) - \log(\hat{p}(r')), \tag{2.21}$$

where $\hat{\mathbf{D}} = \text{diag}(s_1^2, ..., s_p^2) = \text{diag}(\mathbf{S}_{\mathbf{pool}})$. Note that Eq. 2.21 has a similar form like the discriminant function from Eq. 2.8. Here, $\boldsymbol{\Sigma}$ is replaced by the diagonal matrix $\hat{\mathbf{D}}$ and $\boldsymbol{\mu}_r$ by the shrunken centroid vector $\tilde{\bar{\boldsymbol{x}}}_{r'}$. Note that $\hat{p}(r) = \frac{n_r}{n}$ denotes the prior information on the classes.

Shrunken centroids regularized discriminant analysis (SCRDA)

Let us first consider an alternative notation of the linear discriminant function from Eq. 2.8, yielding to equivalent results:

$$d_r(\boldsymbol{x}) \quad = \quad \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_r - \frac{1}{2} \boldsymbol{\mu}_r^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_r + \log(p(r)). \tag{2.22}$$

We obtain the associated estimated discriminant function by replacing $\boldsymbol{\mu}_r$, $\boldsymbol{\Sigma}$ and $p(r)$ in Eq. 2.22 by appropriate estimators. In general, $\boldsymbol{\mu}_r$ is replaced by $\bar{\boldsymbol{x}}_r = \frac{1}{n_r} \sum_{k=1}^{n_r} \boldsymbol{x}_{rk}$ and $p(r)$ by $\hat{p}(r) = \frac{n_r}{n}$, which is independent of the relation between n and p. In the high-dimensional case, however, the usual covariance estimator $\mathbf{S}_{\mathbf{pool}}$ for $\boldsymbol{\Sigma}$ has to be regularized. This leads us to the shrunken centroids regularized discriminant analysis (SCRDA) proposed by Guo et al. in 2007. Here, the mainly used version of regularization in order to resolve the singularity problem is:

$$\hat{\boldsymbol{\Sigma}} \quad = \quad \lambda \mathbf{S}_{\mathbf{pool}} + (1 - \lambda) \mathbf{I}_p, \tag{2.23}$$

where $\mathbf{I}$ is the $(p \times p)$ identity matrix and $\lambda \in [0, 1]$ denotes the shrinkage intensity.

Thus it follows for the estimated discriminant function:

$$\hat{d}_r(\boldsymbol{x}) \quad = \quad \boldsymbol{x}^T \hat{\boldsymbol{\Sigma}}^{-1} \bar{\boldsymbol{x}}_r - \frac{1}{2} \bar{\boldsymbol{x}}_r^T \hat{\boldsymbol{\Sigma}}^{-1} \bar{\boldsymbol{x}}_r + \log(\hat{p}(r)). \tag{2.24}$$

Moreover, a modification of Eq. 2.24 in order to incorporate the idea of the NSC method is to shrink the centroids $\bar{\boldsymbol{x}}_r$, $r = 1, ..., c$, before calculating the discriminant score. In addition to shrinking the centroids directly, $\hat{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{x}}_r$ or $\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}\bar{\boldsymbol{x}}_r$ can be shrunken, whereas Guo et al. decide for $\hat{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{x}}_r$. For clarity's sake, we do not go into detail, but keep the idea in mind. Note that the SCRDA requires determining a pair of shrinkage parameters, often referred to as tuning parameters, i.e. (λ, Δ). We want to mention briefly that Guo et al. use cross-validation in order to determine the 'best' parameter pairs. For further details we refer to [22].

Approach developed by Tai and Pan

Having studied the NSC method and the SCRDA in the previous two paragraphs, we now have the methodical basis for an approach proposed by Tai and Pan in 2007 [46]. In their work, Tai and Pan criticize the assumptions made in both the NSC method and the SCRDA to be too extreme. While the covariance matrix in the former is restricted to be diagonal, i.e. the genes are assumed to be independent of each other, there are no restrictions concerning the covariance structure in the latter. Hence, Tai and Pan propose to estimate the covariance matrix as an intermediate between the two from above which, in addition, integrates biological knowledge on gene functions. The motivation behind that can be depicted in a few words: many genes are known to have the same function or to be involved in the same pathway. For instance, nowadays it is possible to extract biological expertise on cancer-related genes from databases like KEGG [28]. Thus the genes from the same functional group or pathway are assumed to co-express more likely than genes from different gene functional groups, hence their expression levels tend to be correlated. Note that, for the purpose of convenience, Tai and Pan assume the congruency of a KEGG pathway and a gene functional group. In particular, their approach incorporating biological knowledge into discriminant analysis can be explained as follows. The genes from a given data set are grouped according to their biological

functions, i.e. we obtain G gene functional groups. Note that not all genes are annotated in one of the KEGG pathways. Note further that the functional groups are not necessarily disjoint, i.e. there are genes annotated in multiple pathways. In order to deal with these cases, Tai and Pan use the following procedure: if a gene does not occur in any gene functional group, they assume this gene to form its own group with group size one. If a gene occurs in multiple gene functional groups, **(i)** the gene is kept in the smallest functional group and ignored in the other ones it belongs to or **(ii)** the gene is duplicated in order to occur in each functional group. In [46], strategy **(i)** is mainly chosen.

Tai and Pan now regularize the unstructured $(p \times p)$ pooled empirical covariance matrix $\mathbf{S_{pool}}$ by shrinking it towards a between-group independence structure. The latter results from grouping the genes according to their biological functions and from circumventing the overlapping of the groups by using strategy **(i)** as described above. Thus it follows:

$$\hat{\mathbf{\Sigma}} = \lambda_1 \mathbf{S_{pool}} + \lambda_2 \hat{\mathbf{\Sigma}}^* + (1 - \lambda_1 - \lambda_2)\hat{\mathbf{D}}, \tag{2.25}$$

where $\lambda_1, \lambda_2 \in [0, 1]$ and $\lambda_1 + \lambda_2 \leq 1$ are the shrinkage parameters determined by cross-validation. $\hat{\mathbf{D}} = \text{diag}(\mathbf{S_{pool}})$ denotes the $(p \times p)$ diagonal matrix with the pooled empirical variances as entries. Further, $\hat{\mathbf{\Sigma}}^* = \text{diag}(\mathbf{S_{pool}}_1, ..., \mathbf{S_{pool}}_G)$ represents a block-diagonal matrix, where $\mathbf{S_{pool}}_g$, $g = 1, ..., G$, is a $(p_g \times p_g)$ pooled empirical covariance matrix for the genes in the functional group g. Note that the within-group correlation structure may be of any general form. A simpler alternative is defined as follows:

$$\hat{\mathbf{\Sigma}} = \lambda \hat{\mathbf{\Sigma}}^* + (1 - \lambda)\hat{\mathbf{D}}, \tag{2.26}$$

where $\lambda \in [0, 1]$ stands for the shrinkage intensity. Furthermore, Tai and Pan propose a group shrinkage scheme which tends to retain or remove a whole functional group of genes altogether, in contrast to the standard shrinkage on individual genes. Since, in this thesis, our main objective is to study towards which estimator $\mathbf{S_{pool}}$ is

shrunken and how prior biological knowledge is incorporated into the regularization or shrinkage process, we do not go into detail and refer to [46].

2.4.3 Discussion

In a nutshell, let us consider the crucial statements from above. As far as the concrete form of biological knowledge is concerned, the approaches by Guillemot et al. and Tai and Pan differ greatly. In the linear version of gCDA, Guillemot et al. assume the availability of one single graph, including preferably all variables (genes) from the given data set. Having extracted such a graph, detailed knowledge on the connectivity between the graph's vertices is extracted. Further, this knowledge is reflected in the Laplacian matrix $\mathcal{L}_{\mathcal{G}}$ of a prior graph $\mathcal{G}$, which Guillemot et al. propose to consider as the matrix towards the empirical covariance matrix $\mathbf{S}_{\mathbf{pool}}$ is shrunken. Note that $\mathcal{L}_{\mathcal{G}}$ is positive semi-definite and thus has to be modified in order to achieve positive definiteness.

Tai and Pan, on the contrary, differentiate gene functional groups. Thus, the genes from a given data set are grouped according to their biological functions. Since the functional groups extracted from KEGG are not necessarily disjoint, strategies have to be found in order to deal with these cases. One strategy Tai and Pan propose is duplicating the genes that occur in multiple gene functional groups. This procedure, however, increases the matrix's dimension. Consequently, the dimension of $\mathbf{S}_{\mathbf{pool}}$ has to be adapted. Having circumvented the overlapping of the groups, Tai and Pan regularize the unstructured pooled empirical covariance matrix $\mathbf{S}_{\mathbf{pool}}$ by shrinking it towards a between-group independence structure, thus a block-diagonal matrix. In addition, a diagonal matrix is employed in order to ensure positive definiteness. Note that in both approaches a cross-validation procedure is employed for determining the shrinkage intensity.

As further contribution to ongoing research, we aim at developing a simplified version of the regularized linear discriminant analysis proposed by Tai and Pan [46]. Our idea elaborated in this thesis can be outlined as follows. In this simplified version, we replace the empirical within-class covariance matrix by a shrinkage estimator originally introduced by Ledoit and Wolf [31, 33, 32] and picked up by Schäfer and Strimmer in the context of genomic data [41, 40]. In Chapter 3, we

will study this shrinkage estimator in detail. Moreover, we extract prior knowledge on gene functional groups from the database KEGG according to Tai and Pan. In order to incorporate this knowledge into the regularization or shrinkage process, we propose an alternative covariance target similar to target **F** from Schäfer and Strimmer, where genes that are biologically connected, i.e. genes that occur in the same gene functional group, have constant correlation. Note that the term 'covariance target' denotes the highly structured estimator towards the unstructured empirical covariance matrix is shrunken. Unlike Tai and Pan who use a cross-validation procedure for determining the shrinkage intensity, we determine it analytically as introduced by Ledoit and Wolf.

Chapter 3

The shrinkage estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$

The so-called '$n \ll p$' problem is widely known in the context of statistical analysis for high-dimensional microarray data, where the number of variables p (genes) is considerably larger than the number of observations n (chips). Starting from the methodological challenges and approaches discussed in 2.2 and in 2.4, this chapter addresses a further covariance estimation procedure we refer to as **SHIP**: **SH**rinking and **I**ncorporating **P**rior knowledge. Note that, in this chapter, it is our intention to present a new approach concerning covariance estimation in the high-dimensional setting. For this reason, we refer to a standard framework which does not correspond directly to the framework of discriminant analysis, but which can be adapted to it. The special case of discriminant analysis will be studied in Chapter 4. Considering **SHIP**, the resulting covariance estimator is denoted by $\hat{\mathbf{\Sigma}}_{\mathbf{SHIP}}$. It represents the shrinkage estimator introduced by Ledoit and Wolf [31, 33, 32] we refer to as $\hat{\mathbf{\Sigma}}_{\mathbf{SH}}$, being enhanced by consideration of prior knowledge on gene functional groups as described in 2.3 and 3.2.2.

$$\hat{\mathbf{\Sigma}}_{\mathbf{SH}} \xrightarrow{+\ \text{PRIOR KNOWLEDGE}} \hat{\mathbf{\Sigma}}_{\mathbf{SHIP}}$$

We will see that $\hat{\mathbf{\Sigma}}_{\mathbf{SH}}$ and $\hat{\mathbf{\Sigma}}_{\mathbf{SHIP}}$ only differ in terms of a covariance target whose choice we discuss in detail in 3.2. Hence, in the remainder of this work we use the notation $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ when discussing the method in general. Moreover, since we pursue

the aim of proposing a method that embeds prior knowledge on gene functional groups, we will pinpoint clearly the transition from $\hat{\mathbf{\Sigma}}_{\mathbf{SH}}$ to $\hat{\mathbf{\Sigma}}_{\mathbf{SHIP}}$.

3.1 Introduction to $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$

Many statistical methods require an estimator of the covariance matrix that is both invertible and well-conditioned (i.e. inversion of the matrix does not amplify the estimation error). For instance, the linear discriminant analysis described in 2.1 encloses the inverse of the covariance matrix estimator in its discriminant function used for classification of an observation to the most likely underlying class. Generally, the traditional estimators are the maximum likelihood estimator $\hat{\mathbf{\Sigma}}_{\mathbf{ML}}$ or the related unbiased empirical covariance matrix $\mathbf{S} = \frac{n}{n-1}\,\hat{\mathbf{\Sigma}}_{\mathbf{ML}}$, whose entries are defined as

$$s_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j), \tag{3.1}$$

where $\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki}$ and x_{ki} is the k-th observation of the variable X_i. However, in the special case of linear discriminant analysis, the traditional estimators are the pooled maximum likelihood estimator $\hat{\mathbf{\Sigma}}_{\mathbf{ML_{pool}}}$ or the related unbiased pooled empirical covariance matrix $\mathbf{S}_{\mathbf{pool}} = \frac{n}{n-c}\,\hat{\mathbf{\Sigma}}_{\mathbf{ML_{pool}}}$, whose entries are defined as

$$\begin{aligned} s_{ij_{\mathbf{pool}}} &= \frac{1}{n-c}\sum_{r=1}^{c}\underbrace{\sum_{k=1}^{n_r}(x_{rki}-\bar{x}_{ri})(x_{rkj}-\bar{x}_{rj})}_{(n_r-1)s_{ij}^{(r)}} \\ &= \frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)s_{ij}^{(r)}, \end{aligned} \tag{3.2}$$

where $\bar{x}_{ri} = \frac{1}{n_r}\sum_{k=1}^{n_r} x_{rki}$, x_{rki} is the k-th observation of the variable X_i in class r and $s_{ij}^{(r)}$ is the (ij)-th entry of the standard unbiased empirical covariance matrix for class r, $r = 1, ..., c$ [18, 48]. Thus is appears that the pooled empirical covariance matrix $\mathbf{S}_{\mathbf{pool}}$ can be written as a weighted sum of the within-class covariance matrices, which in turn are estimated by the standard empirical covariance matrix as

denoted by equation 3.1. The latter can be regarded as the more general estimator since, besides classification, a multiplicity of other methods, for example interval estimation and graphical models, require a well-conditioned estimator of the inverse covariance matrix. Therefore, in this chapter we constrain our attention on the empirical covariance matrix $\mathbf{S} = (s)_{ij}$, $i, j = 1, ..., p$.

However, in the high-dimensional data setting, the usual estimation procedure yields undesirable characteristics of the resulting estimator: generally, it is ill-conditioned and singular, thus not invertible. According to Schäfer and Strimmer [41, 40], we study for fixed $p = 100$ and various ratios $\frac{p}{n}$ the sorted eigenvalues of the sample covariance matrix $\mathbf{S} = (s)_{ij}$ and compare it to the true eigenvalues. The resulting Figure 3.1 presented below shows that for $\frac{p}{n} > 1$ the eigenvalues differ greatly, whereas for $\frac{p}{n} < 1$ the difference is rather small. Further, Figure 3.1 illustrates clearly that for $n \ll p$ the sample covariance matrix loses its full rank as a growing number of eigenvalues become zero. As a result, the sample covariance matrix is neither positive definite nor invertible. Note that the positive-definiteness requirement is an intrinsic property of the true covariance matrix; it is fulfilled as long as the considered random variables have non-zero variance.

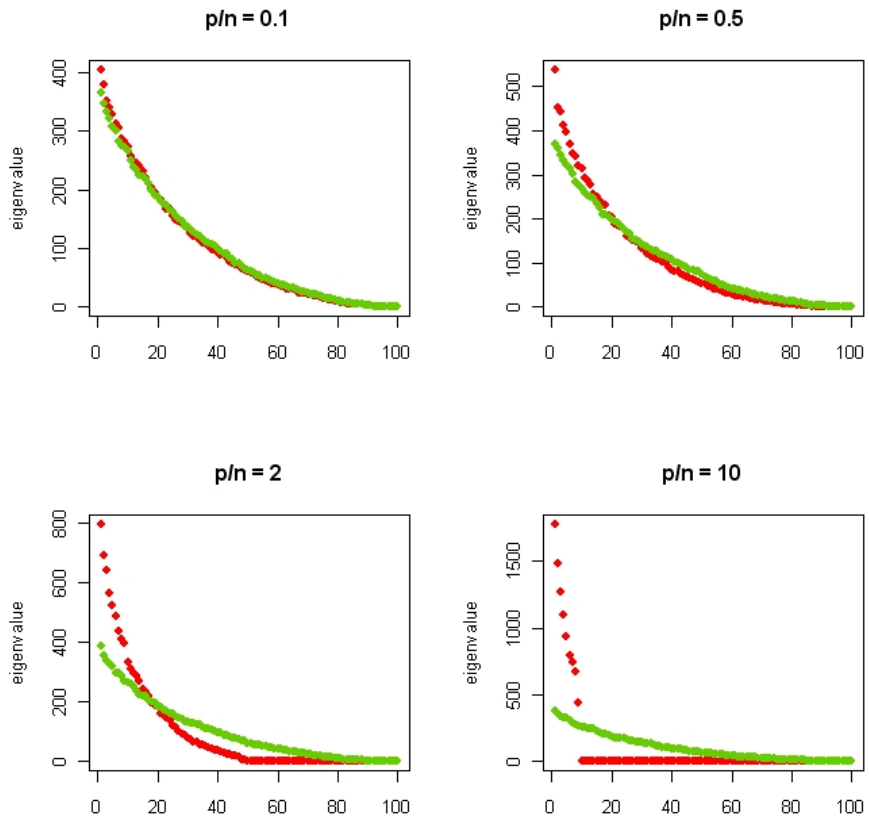

Figure 3.1: Ordered eigenvalues of the sample covariance matrix (red points) and true eigenvalues (green points), calculated from simulated data with underlying p-variate normal distribution, for $p = 100$ and various ratios p/n. The figure is, with minor modifications, adopted from Schäfer and Strimmer [41].

Consequently, the sample covariance matrix $\mathbf{S}$ as the most commonly used covariance estimator is estimated with an extreme amount of error unless p is considerably smaller than n. Therefore, in the recent years statisticians have been engaged in developing methods which improve the estimation of the covariance matrix and thus circumvent these drawbacks. A clearly arranged review on this topic is given by Schäfer and Strimmer [41, 40]. For instance, a strategy to obtain a positive definite estimator of the covariance matrix is the application of the algorithm by Higham [24] to the sample covariance matrix. The algorithm adjusts all eigenvalues to be larger than some prespecified threshold ϵ and thereby guarantees positive def-

initeness. Nevertheless, the resulting matrix is not well-conditioned. As a further contribution to this problem, Ledoit and Wolf propose an estimator [31, 33, 32], referred to as $\hat{\mathbf{\Sigma}}_{\mathbf{SH}}$ in this thesis, which is based on the widely employed shrinkage principle as outlined in 2.2.2. In our context, the estimator without structure is the sample covariance matrix $\mathbf{S}$. However, the structured estimator, referred to as shrinkage target $\mathbf{T}$, has to be chosen suitably. More precisely, $\mathbf{T}$ should involve only a small number of free parameters and it must be positive definite. Nevertheless, it should reflect important characteristics of the shrinkage estimator.
Another challenge, from the statisticians' point of view, is the computation of the optimal shrinkage intensity, referred to as λ. Ledoit and Wolf introduce an analytic determination of λ, which is a distinct advantage over determining it heuristically, usually by cross-validation [19]. The main drawback of such heuristic approaches is that they are computationally very intensive. The difficulty of the analytic determination is that λ depends on the unobservable true covariance matrix. Ledoit and Wolf solve this difficulty by replacing the true optimal λ by a consistent estimator $\hat{\lambda}$ and by proving the asymptotic equality of λ and $\hat{\lambda}$. In detail, we will deal with these aspects in 3.2 and in 3.3, respectively.

Assuming that the shrinkage target $\mathbf{T}$ is chosen and the shrinkage intensity λ is computed, the shrinkage estimator proposed by Ledoit and Wolf is the following asymptotically optimal convex linear combination:

$$\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}} \quad = \quad \hat{\lambda}\mathbf{T} + (1 - \hat{\lambda})\mathbf{S}, \tag{3.3}$$

where $\lambda \in [0, 1]$: shrinkage intensity that is determined analytically according to Ledoit and Wolf
$\mathbf{T}$: covariance target to be chosen suitably
$\mathbf{S}$: unbiased empirical covariance matrix $\mathbf{S} = \frac{n}{n-1}\,\hat{\mathbf{\Sigma}}_{\mathbf{ML}}$.

In this context, optimality is meant with respect to a quadratic loss function, which is common and intuitive in statistical decision theory [4]. The asymptotic result, however, is less intuitive and requires further explanations: standard asymptotics assume the number of variables p to be finite and the number of obervations n to

go to infinity. In this framework, the sample covariance matrix is well-conditioned asympotically. Nevertheless, the high-dimensional data setting does not comply with the assumptions of standard asymptotics. Hence, Ledoit and Wolf use the framework of general asymptotics, which allows both the number of variables p and the number of observations n to go to infinity, whereas the ratio $\frac{p}{n}$ must remain bounded. Detailed information concerning general asymptotics can be found in [31, 33, 32]. Since Monte-Carlo simulations confirm that the asymptotic results hold well in finite samples with at least twenty observations and variables [31], this estimation procedure is appropriate for the analysis of microarray gene expression data where the number of variables p goes to infinity, but the number of observations n remains small. The resulting estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ has the following properties: it is more efficient and more accurate than the sample covariance matrix, it is positive definite, well-conditioned and invertible, which are crucial properties with regard to the estimation of the inverse of the true covariance matrix. Further, $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ has guaranteed minimum mean squared error, which results from the quadratic loss function [4]. Another interesting property of $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ is that it does not assume any fully specified distribution. Since merely second moments are required, $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ is distribution-free in principle. Note that $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ is not only feasible for genomic data, but can be employed in each high-dimensional setting such as financial data, which actually was the original objective of Ledoit and Wolf in [31, 33, 32]. Schäfer and Strimmer [41, 40] and Opgen-Rhein and Strimmer [37] proposed the application to genomic data and could illustrate its high performance.

At this point, we assume that the chosen covariance target $\mathbf{T} = (t)_{ij}$ incorporates prior knowledge on gene functional groups. Hence, we obtain the concrete covariance estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SHIP}}$. In addition to the previous explanations, the following Figure 3.2 summarizes for the $n \ll p$ case both the properties of the sample covariance matrix $\mathbf{S}$ and the properties of the covariance estimator obtained ***'via the SHIP'***. This new estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SHIP}}$ results from shrinking the sample covariance matrix $\mathbf{S} = (s)_{ij}$ and from incorporating prior knowledge into the shrinkage process.

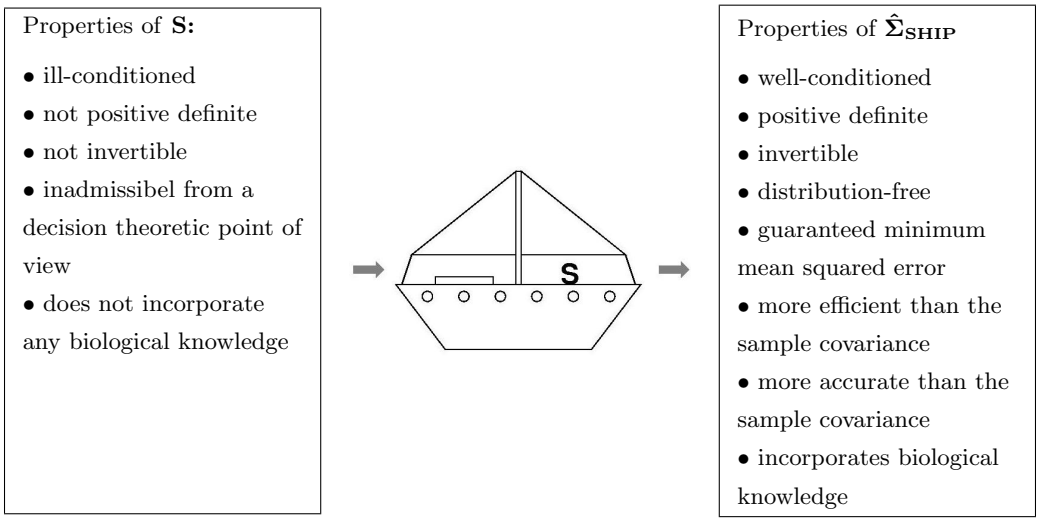

Figure 3.2: Properties of the covariance estimator (for $n \ll p$) before and after SHIP.

3.2 The covariance target T

In this section we want to focus on the choice of the covariance target $\mathbf{T} = (t)_{ij}$ that plays an essential role in the computation of the shrinkage estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$. It holds (see Eq. 3.3):

$$\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}} \quad = \quad \hat{\lambda}\mathbf{T} + (1 - \hat{\lambda})\mathbf{S},$$

where $\lambda \in [0, 1]$: shrinkage intensity that is determined analytically according to Ledoit and Wolf

$\mathbf{T}$: covariance target to be chosen suitably

$\mathbf{S}$: unbiased empirical covariance matrix $\mathbf{S} = \frac{n}{n-1}\,\hat{\mathbf{\Sigma}}_{\mathbf{ML}}$.

The choice of a suitable lower-dimensional covariance target turns out to be very complex. In a nutshell, $\mathbf{T}$ has to fulfill the following requirements:

i) **T** must be positive definite.
ii) **T** should involve only a small number of free parameters.
iii) **T** should reflect important characteristics of the shrinkage estimator.

We will see that, in order to fulfill **i)**, a compromise between **ii)** and **iii)** is inevitable. In the first part of this section we give a brief overview of the lower-dimensional targets for the covariance matrix outlined in Schäfer and Strimmer [41]. Examples for these covariance targets can be found easily in the literature, albeit not in the combination with an analytic determination of the shrinkage intensity. Second, we propose target **G**, a modified version of target **F** from Schäfer and Strimmer that incorporates this biological knowledge, i.e. genes that are biologically connected have constant correlation. We compute target **G** for real data and investigate its adequacy. Third, we propose target **G***, an alternative to target **G** that is more adequate in the context of biological interpretation. Further, we point out some algorithmic aspects. The section completes with studying the definiteness of the covariance targets incorporating prior knowledge on gene functional groups.

3.2.1 Common covariance targets

Schäfer and Strimmer [41] compile the following overview of commonly used covariance targets which we will extend by proposing new covariance targets in 3.2.2, taking into account prior knowledge on gene functional groups. A complete overview of all covariance targets including the associated estimators of the optimal shrinkage intensity will be depicted in 3.4.

- **Target A:** 'diagonal, unit variance'; 0 estimated parameters

$$t_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

- **Target B:** 'diagonal, common variance'; 1 estimated parameter: ν

$$t_{ij} = \begin{cases} \nu = \text{avg}(s_{ii}) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

- **Target C:** 'common (co)variance'; 2 estimated parameters: ν, c

$$t_{ij} = \begin{cases} \nu = \text{avg}(s_{ii}) & \text{if } i = j \\ c = \text{avg}(s_{ij}) & \text{if } i \neq j \end{cases}$$

- **Target D:** 'diagonal, unequal variance'; p estimated parameters: s_{ii}

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

- **Target E:** 'perfect positive correlation'; p estimated parameters: s_{ii}

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \sqrt{s_{ii}s_{jj}} & \text{if } i \neq j \end{cases}$$

- **Target F:** 'constant correlation'; $p + 1$ estimated parameters: s_{ii}, $\bar{r}$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}\sqrt{s_{ii}s_{jj}} & \text{if } i \neq j \end{cases}$$

where ν : average of sample variances
c : average of sample covariances
$\bar{r}$: average of sample correlations.

Thus it appears that the shrinkage targets can be divided into two classes. The first class comprises target **A** ('diagonal, unit variance'), target **B** ('diagonal, common variance') and target **C** ('common (co)variance'), which share several properties. First, they are all extremely low-dimensional (0 to 2 free parameters), thus they are highly structured. Second, the resulting covariance estimators shrink all components of the sample covariance matrix, i.e. both the diagonal and the off-diagonal entries. The probably mostly employed covariance targets are target **A** and target **B**, whereas the two-parameter target **C** appears not to be widely used. The second class of covariance targets comprises target **D** ('diagonal, unequal variance'), target **E** ('perfect positive correlation') and target **F** ('constant correlation'), whereas especially the latter is employed in Ledoit and Wolf [33]. The properties shared by these three targets are that they are comparatively parameter-rich, and that they only shrink the off-diagonal elements of **S**. Schäfer and Strimmer and Opgen-Rhein and Strimmer point out that, in consequence of the grouping of the covariance targets, the diagonal and the off-diagonal elements can be treated differently in the shrinkage process. We will deal with this aspect in 3.3.3. Schäfer and Strimmer [41, 40] focused on target **D** in the process of covariance estimation. According to target **A** and target **B** it shrinks the off-diagonal entries to zero. At the same time, like target **E** and **F**, it leaves the diagonal entries intact, i.e. it does not shrink the variances. Therefore, we can consider target **D** as a compromise between the low-dimensional targets **A** and **B** and the correlation models **E** and **F**. The shrinkage estimator described in Schäfer and Strimmer is implemented in the open source `R` package `corpcor`.

3.2.2 Covariance targets incorporating prior knowledge on gene functional groups

Target G

To incorporate the external biological knowledge from KEGG PATHWAY, we propose a modified version of target **F** from Schäfer and Strimmer [41], where only the genes that occur in at least one same gene functional group have constant correlation. Consequently, in order to obtain $\bar{r}$ we just account for the correlations of the genes that have at least one gene functional group in common.
We use the same notations as Schäfer and Strimmer [41]. Moreover, the notation $i \sim j$ means that genes i and j are 'connected', i.e. genes i and j occur in the same gene functional group.

Target G: 'constant correlation between connected genes';
p+1 estimated parameters: s_{ii}, $\bar{r}$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}\sqrt{s_{ii}s_{jj}} & \text{if } i \neq j, i \sim j \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{r}$ is the average of sample correlations between connected genes.

Adequacy of Target G

In a sense, target **G** assumes positive or at least not relevant negative correlations among the genes. As the constant correlation $\bar{r}$ is the average of sample correlations between the connected genes, a high number of negative correlations leads to a falsified $\bar{r}$. Hence, we now focus on the within-group correlations. Note that, in this context, the term 'within-group' means 'within at least one same functional group'. In order to investigate whether these within-group correlations are negative or positive in practice, we compute target **G** for the public microarray gene expression data sets described in 1.3, whereas we only use the top 2000 genes in each data set.

For the gene selection we use the method `GeneSelection()` of the open source `R` package `CMA` [43] as described in 1.3. In the following, we deal with the correlation structure of target $\mathbf{G}$, with special attention to the effect of the negative correlations on the average correlation as used in target $\mathbf{G}$. Note that we only use the two-class data sets Golub_Merge, ALL_c and sCLLex.
We first present in different tables analyses of the correlation structure in these data sets. Finally, we draw the conclusion that two constant correlations, a positive and a negative one, would be more adequate to describe the within-group correlation structure.

	Golub_Merge	**ALL_c**	**sCLLex**
n	72	128	22
c (# classes)	2	2	2
p (# genes)	2 000	2 000	2 000
# genes in no gene functional group	1 158	1 217	1 260
# corr. (all)	19 090	20 839	14 862
# corr. < 0	7 526	9 669	6 018
# corr. > 0	11 564	11 170	8 844
mean corr. (all)	**0.098**	**0.047**	**0.111**
mean corr. (without neg. corr.)	**0.268**	**0.273**	**0.364**

Table 3.1: Overview of the correlation structure of target $\mathbf{G}$ for the data sets Golub_Merge, ALL_c and sCLLex. Since the covariance target is symmetric, we only consider the correlations between **different** pairs of genes without the diagonal elements.

	# sign. corr.	**# not sign. corr.**	
# neg. corr.	1 819 *(9.53 %)*	5 707 *(29.89 %)*	7 526 *(39.42 %)*
# pos. corr.	5 914 *(30.97 %)*	5 650 *(29.59 %)*	11 564 *(60.57 %)*
	7 733 *(40.50 %)*	11 357 *(59.49 %)*	19 090 *(100.00 %)*

Table 3.2: Analysis of the correlations in target **G** for the data Golub_Merge. A standard correlation test is used with a confidence level of 0.95. In brackets the percentage of the total number of correlations between **different** pairs of genes is given.

	# sign. corr.	**# not sign. corr.**	
# neg. corr.	5 569 *(26.72 %)*	4 100 *(19.67 %)*	9 669 *(46.39 %)*
# pos. corr.	7 460 *(35.79 %)*	3 710 *(17.80 %)*	11 170 *(53.60 %)*
	13 029 *(62.52 %)*	7 810 *(37.47 %)*	20 839 *(100.00 %)*

Table 3.3: Analysis of the correlations in target **G** for the data ALL_c. A standard correlation test is used with a confidence level of 0.95. In brackets the percentage of the total number of correlations between **different** pairs of genes is given.

	# sign. corr.	**# not sign. corr.**	
# neg. corr.	1 168 *(7.85 %)*	4 850 *(32.63 %)*	6 018 *(40.49 %)*
# pos. corr.	3 359 *(22.60 %)*	5 485 *(36.90 %)*	8 844 *(59.50 %)*
	4 527 *(30.46 %)*	10 335 *(69.53 %)*	14 862 *(100.00 %)*

Table 3.4: Analysis of the correlations in target **G** for the data CLL. A standard correlation test is used with a confidence level of 0.95. In brackets the percentage of the total number of correlations between **different** pairs of genes is given.

The results shown in Table 3.1 approve the assumption that target **G** does not adequately represent the real within-group correlation structure. For all data sets we obtain a noticeably higher average correlation $\bar{r}$ by leaving out the negative correlations in its computation. In order to receive an impression of the intensity of the negative correlations, we apply a standard correlation test to each different pair of genes, with a confidence level of 0.95. The results for each of the three data

sets Golub_Merge, ALL_c and sCLLex are presented in Table 3.2, Table 3.3 and Table 3.4. A noticeable part of the negative correlations is not significant. However, for biological interpretation purposes, instead of leaving out any genes we consider all genes and introduce two constant correlations $\bar{r}_-$ and $\bar{r}_+$, i.e. a positive and a negative one. The resulting covariance target **G*** is defined as follows:

Target G*: an alternative to target G

According to the results above, we propose target **G***, a modified version of the lower-dimensional target **G** that represents more adequately the real correlation structure in the gene functional groups by introducing two constant correlations, a positive and a negative one. Hence, it is more adequate in the context of biological interpretation. We point out that target **G*** is not necessarily the better choice concerning the prediction quality; we deal with this aspect in the sequel.

Target G*: 'two constant correlations between connected genes: a negative and a positive one'; $p+2$ estimated parameters: s_{ii}, $\bar{r}_-$, $\bar{r}_+$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}_- \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j, i \sim^{\ominus} j \\ \bar{r}_+ \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j, i \sim^{\oplus} j \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{r}_-$ is the average of *negative* sample correlations between connected genes and $\bar{r}_+$ is the average of *positive* sample correlations between connected genes. Moreover, the notation $i \sim^{\ominus} j$ means that genes i and j are 'negatively connected', i.e. genes i and j occur in the same gene functional group and are negatively correlated. Accordingly, the notation $i \sim^{\oplus} j$ means that genes i and j are 'positively connected', i.e. genes i and j occur in the same gene functional group and are positively correlated.

Additional aspects concerning target F

Unlike target **G**, target **F** from Schäfer and Strimmer [41] does not incorporate biological knowledge on gene functional groups. Nevertheless, the assumed correlation structure of target **F** implies the same difficulties like target **G**: a high number of negative correlations leads to a falsified $\bar{r}$. For this reason, we carry out the same analyses for target **F** as for target **G**. We use the same subsets of the data Golub_Merge, ALL_c and sCLLex as for the analyses concerning target **G**.

	Golub_Merge	**ALL_c**	**sCLLex**
n	72	128	22
c (# classes)	2	2	2
p (# genes)	2000	2000	2000
# genes in no gene functional group	1158	1217	1260
# corr. (all)	1 999 000	1 999 000	1 999 000
# corr. < 0	862 638	970 516	1 009 467
# corr. > 0	1 136 362	1 028 484	989 533
mean corr. (all)	**0.065**	**0.016**	**0.003**
mean corr. (without neg. corr.)	**0.235**	**0.224**	**0.291**

Table 3.5: Overview of the correlation structure of target **F** for the data sets Golub_Merge, ALL_c and sCLLex. Since the covariance target is symmetric, we only consider the correlations between **different** pairs of genes without the diagonal elements.

	# sign. corr.	**# not sign. corr.**	
# neg. corr.	200 079 *(10.00 %)*	662 559 *(33.14 %)*	862 638 *(43.15 %)*
# pos. corr.	505 428 *(25.28 %)*	630 934 *(31.56 %)*	1 136 362 *(56.84 %)*
	705 507 *(35.29 %)*	1 293 493 *(64.70 %)*	1 999 000 *(100.00 %)*

Table 3.6: Analysis of the correlations in target **F** for the data Golub_Merge. A standard correlation test is used with a confidence level of 0.95. In brackets the percentage of the total number of correlations between **different** pairs of genes is given.

	# **sign. corr.**	# **not sign. corr.**	
# **neg. corr.**	540 013 *(27.01 %)*	430 503 *(21.53 %)*	970 516 *(48.55 %)*
# **pos. corr.**	602 068 *(30.11 %)*	426 416 *(21.33 %)*	1 028 484 *(51.44 %)*
	1 142 081 *(57.13 %)*	856 919 *(42.86 %)*	1 999 000 *(100.00 %)*

Table 3.7: Analysis of the correlations in target **F** for the data ALL_c. A standard correlation test is used with a confidence level of 0.95. In brackets the percentage of the total number of correlations between **different** pairs of genes is given.

	# **sign. corr.**	# **not sign. corr.**	
# **neg. corr.**	231 826 *(11.59 %)*	777 641 *(38.90 %)*	1 009 467 *(50.49 %)*
# **pos. corr.**	251 184 *(12.56 %)*	738 349 *(36.93 %)*	989 533 *(49.50 %)*
	483 010 *(24.16 %)*	1 515 990 *(75.83 %)*	1 999 000 *(100.00 %)*

Table 3.8: Analysis of the correlations in target **F** for the data CLL. A standard correlation test is used with a confidence level of 0.95. In brackets the percentage of the total number of correlations between **different** pairs of genes is given.

The results in Table 3.5 show that target **F** - according to target **G** - does not adequately represent the real correlation structure. For all data sets we obtain a noticeably higher average correlation $\bar{r}$ by leaving out the negative correlations in its computation. In order to receive an impression of the intensity of the negative correlations, we apply a standard correlation test to each different pair of genes, with a confidence level of 0.95. The results for each of the three data sets Golub_Merge, ALL_c and sCLLex are presented in Table 3.6, Table 3.7 and Table 3.8. A noticeable part of the negative correlations is not significant. Analog the procedure described for target **G**, instead of leaving out any genes we consider all of them and introduce two constant correlations $\bar{r}_-$ and $\bar{r}_+$, i.e. a positive and a negative one. The resulting covariance target is defined as follows:

Target F*: 'two constant correlations between genes: a negative and a positive one'; $p+2$ estimated parameters: s_{ii}, $\bar{r}_-$, $\bar{r}_+$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}_- \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j, cor(i,j) < 0 \\ \bar{r}_+ \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j, cor(i,j) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{r}_-$ is the average of *negative* sample correlations between the genes and $\bar{r}_+$ is the average of *positive* sample correlations between the genes.
As usual, the notation $cor(i,j) < 0$ means that genes are negatively correlated. Accordingly, the notation $cor(i,j) > 0$ means that genes i and j are positively correlated.

3.2.3 Algorithmic aspects

In a nutshell, we point out some algorithmic aspects since the computation of target **G** and target **G*** requires a suitable procedure for the following occuring cases:

i) A gene does not occur in any gene functional group.

ii) A gene occurs in multiple gene functional groups.

iii) A pair of genes occurs in multiple gene functional groups.

We propose the following procedure to deal with cases **i) - iii)**:

case i):
If a gene does not occur in any gene functional group, we assume that this gene forms its own group with group size one. This corresponds to Tai and Pan [46].

case ii):
Unlike Tai and Pan [46] who assume a between-group gene independence in the co-

variance target, target **G** is more flexible concerning the between-group correlation structure. Hence, if a gene occurs in multiple gene functional groups, we do not need a special treatment as this is considered by the definition of target **G**.

case iii):
If a pair of genes occurs in multiple gene functional groups, we ignore this in our algorithm and only consider this pair once, i.e. the genes that occur in **at least** one same functional group have constant correlation. One may argue that a pair or group of genes occuring in multiple gene functional groups may be more important and the correlation between these genes should be augmented in an appropriate way. Therefore, we investigate the number of pairs of genes occuring in multiple gene functional groups:

	Golub_Merge	**ALL_c**	**sCLLex**
p (# genes)	2 000	2 000	2 000
# genes in no gene func. group	1 158	1 217	1 260
# gene func. groups	184	185	180
min. # of gene func. groups a pair of genes occurs in	0	0	0
max. # of gene func. groups a pair of genes occurs in	27	32	19
# corr. (all) = # pairs of genes	19 090	20 839	14 862
# pairs of genes in > 1 gene func. groups	3 107	4 440	3 293
# pairs of genes in > 2 gene func. groups	1 133	1 907	1 679
# pairs of genes in > 3 gene func. groups	569	1 199	344
# pairs of genes in > 4 gene func. groups	336	874	177
# pairs of genes in > 5 gene func. groups	210	663	101
# pairs of genes in > 6 gene func. groups	134	358	62
# pairs of genes in > 7 gene func. groups	90	250	35
# pairs of genes in > 8 gene func. groups	68	111	24
# pairs of genes in > 9 gene func. groups	44	71	18
# pairs of genes in > 10 gene func. groups	31	43	15

Table 3.9: Overview of the number of pairs of genes occuring in multiple gene functional groups. Analyses here are carried out for the same subsets of the data Golub_Merge, ALL_c and sCLLex as used above. Since the covariance target is symmetric, we only consider the **different** pairs of genes without the diagonal elements.

The results shown in Table 3.9 suggest a more precise algorithm, taking into consideration the pairs of genes occuring in multiple gene functional groups. Here, we focus on our algorithm that considers these pairs only once and leave the eventually more suitable algorithm for further research.

3.2.4 The definiteness of the covariance targets incorporating prior knowledge on gene functional groups

In 3.2, we have discussed the main requirements concerning the covariance target $\mathbf{T} = (t)_{ij}$. Since $\mathbf{T}$ is a low-dimensional representation of the covariance matrix in the shrinkage process, the fulfillment of the positive definiteness is essential. Target $\mathbf{D}$ which is employed in Schäfer and Strimmer has the important advantage that the resulting shrinkage covariance estimator is automatically positive definite for the following reason: target $\mathbf{D}$ as a diagonal matrix is always positive definite. Further, the convex combination of a positive definite matrix with another positive semidefinite matrix results in a positive definite matrix. Schäfer and Strimmer point out that this holds for the targets $\mathbf{A}$ and $\mathbf{B}$, but not for the targets $\mathbf{C}$, $\mathbf{E}$ and $\mathbf{F}$ [41, 40] which have off-diagonal entries not equal to zero. It is not surprising that the same problem occurs for the covariance targets **F***, **G** and **G*** since they represent modified versions of target $\mathbf{F}$. The figures presented below confirm the theoretic considerations. For each real data set we illustrate the sorted eigenvalues of the covariance targets **G** and **G*** for the top 2000, 1000, 500 and 100 genes. Note that we only use the two-class data sets Golub_Merge, ALL_c and sCLLex. For comparison purposes, we present the same figure for the diagonal covariance target $\mathbf{D}$. For all three data sets we obtain indefinite covariance targets **G** and especially **G*** for at least one set of genes, whereas the covariance target $\mathbf{D}$ remains positive definite in either case. Note that the covariance targets' structure is manipulated. Hence, indefiniteness is possible, although a covariance matrix is (semi-) positive definite per definition.

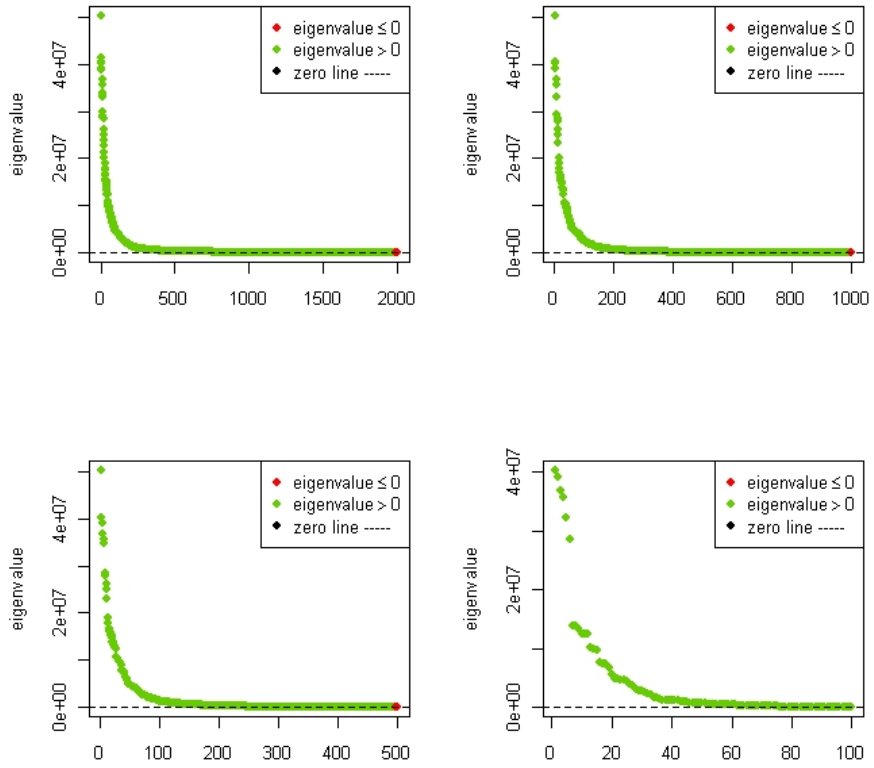

Figure 3.3: Plots illustrating the sorted eigenvalues of target $\mathbf{G}$ for the top 2000, 1000, 500 and 100 genes in the data set Golub_Merge.

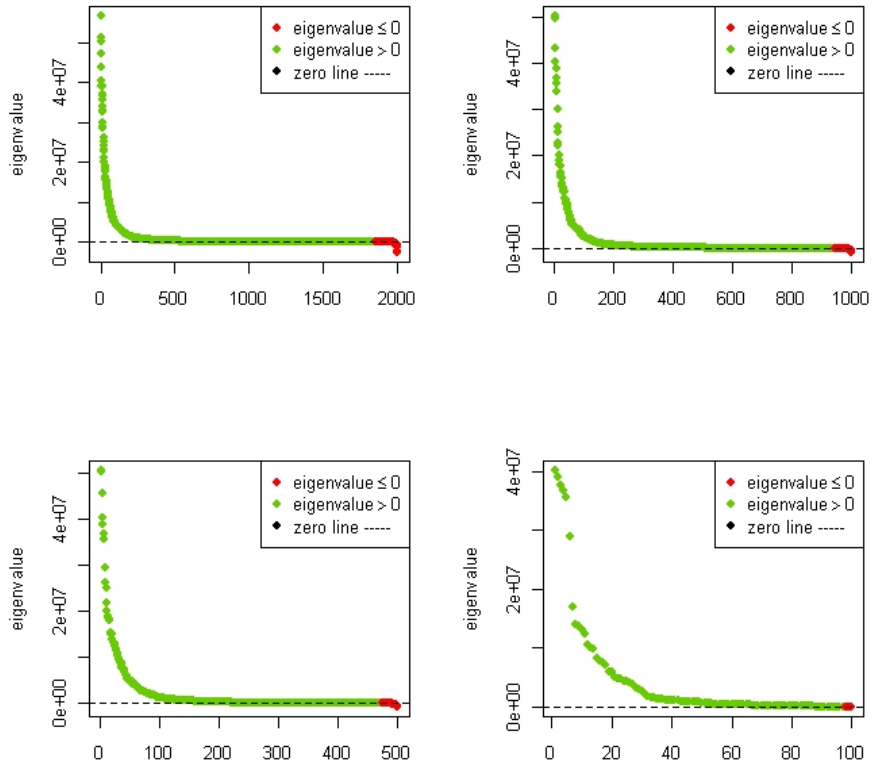

Figure 3.4: Plots illustrating the sorted eigenvalues of target **G*** for the top 2000, 1000, 500 and 100 genes in the data set Golub_Merge.

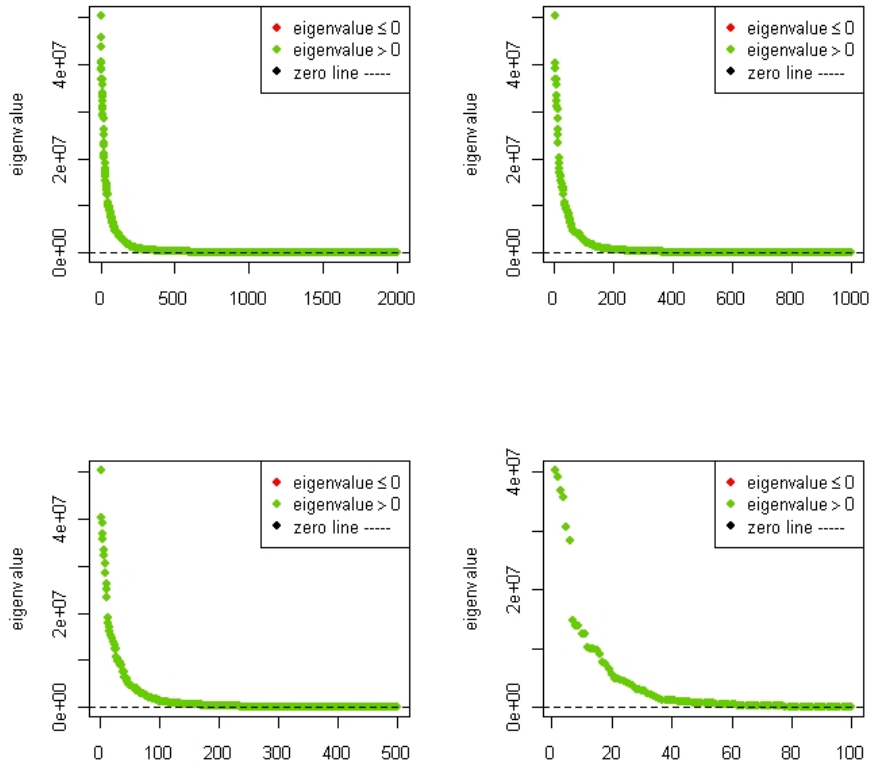

Figure 3.5: Plots illustrating the sorted eigenvalues of target **D** for the top 2000, 1000, 500 and 100 genes in the data set Golub_Merge.

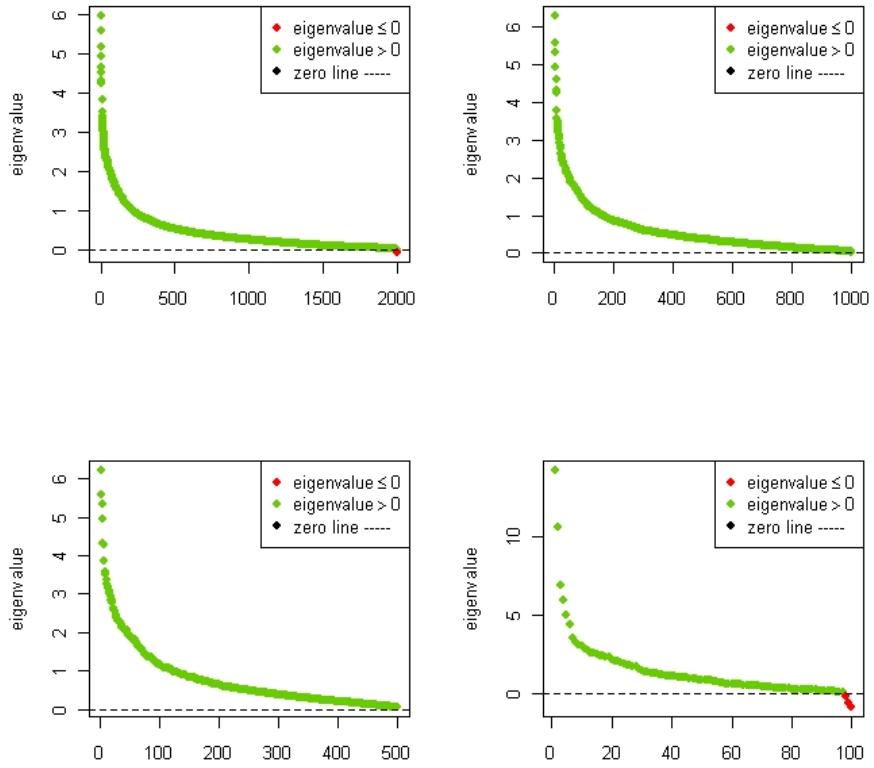

Figure 3.6: Plots illustrating the sorted eigenvalues of target $\mathbf{G}$ for the top 2000, 1000, 500 and 100 genes in the data set ALL_c.

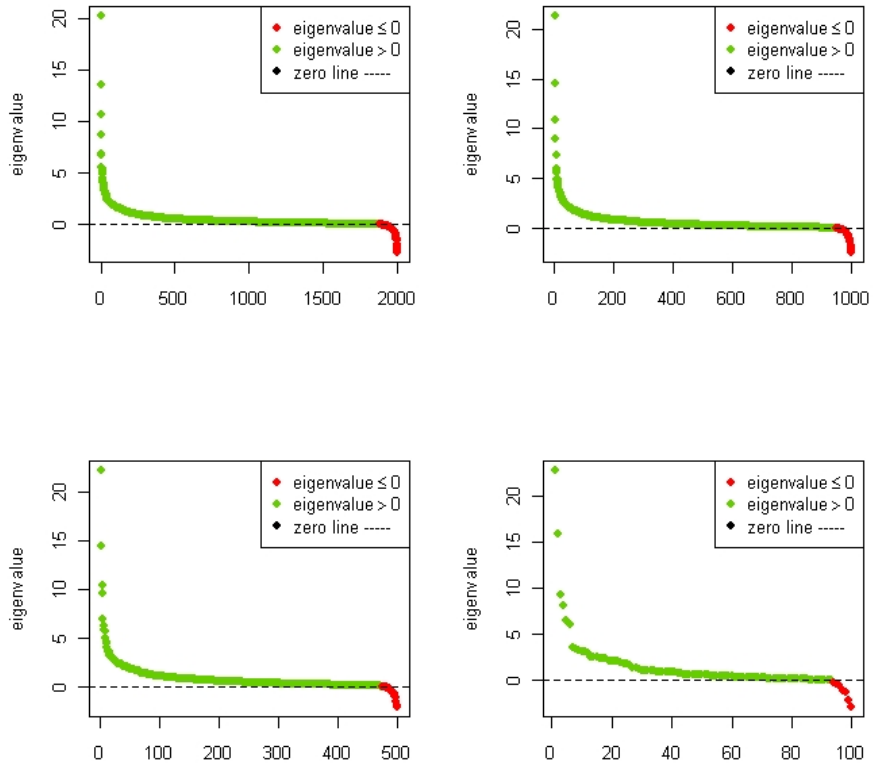

Figure 3.7: Plots illustrating the sorted eigenvalues of target **G*** for the top 2000, 1000, 500 and 100 genes in the data set ALL_c.

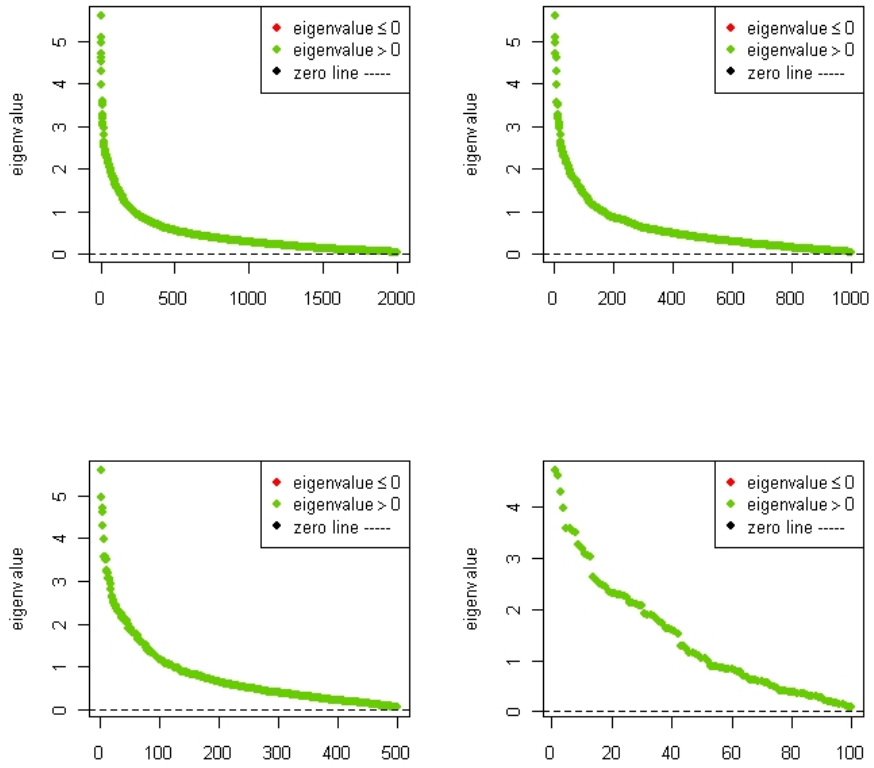

Figure 3.8: Plots illustrating the sorted eigenvalues of target **D** for the top 2000, 1000, 500 and 100 genes in the data set ALL_c.

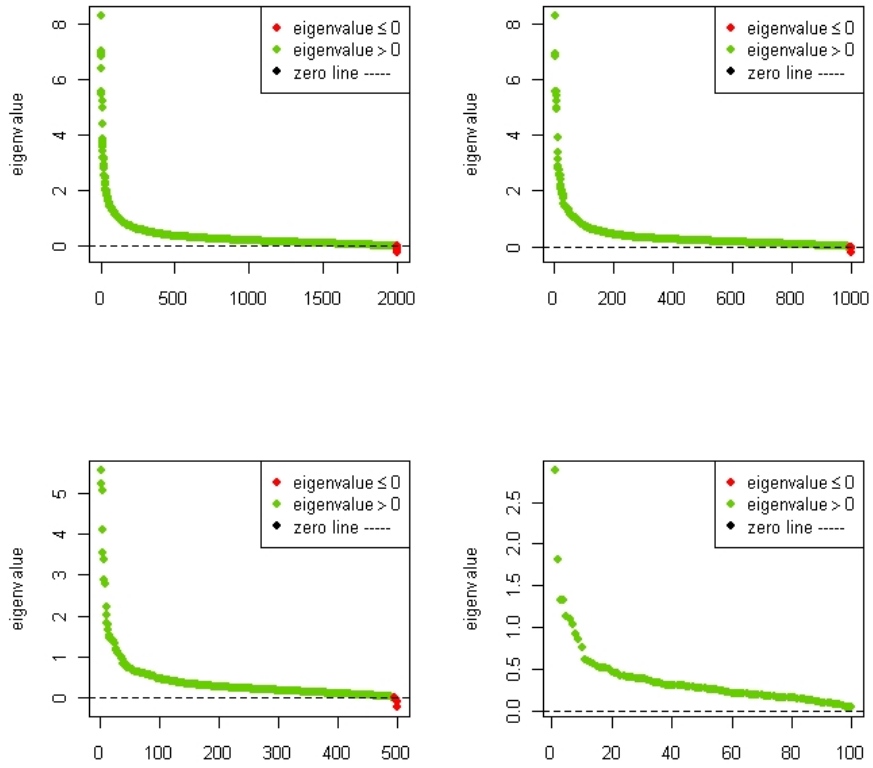

Figure 3.9: Plots illustrating the sorted eigenvalues of target $\mathbf{G}$ for the top 2000, 1000, 500 and 100 genes in the data set sCLLex.

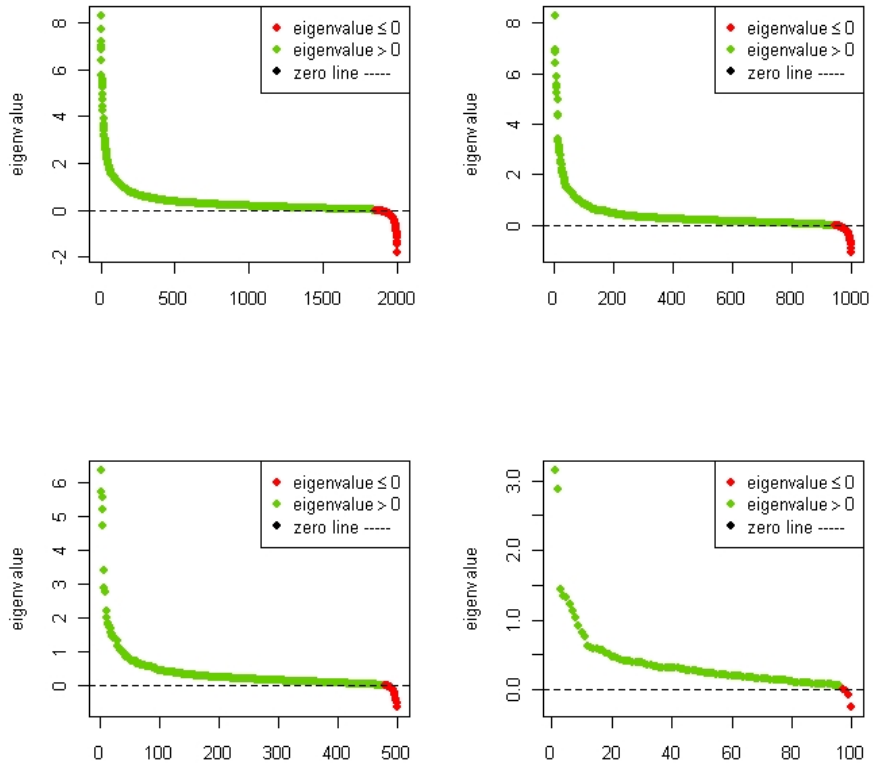

Figure 3.10: Plots illustrating the sorted eigenvalues of target **G*** for the top 2000, 1000, 500 and 100 genes in the data set sCLLex.

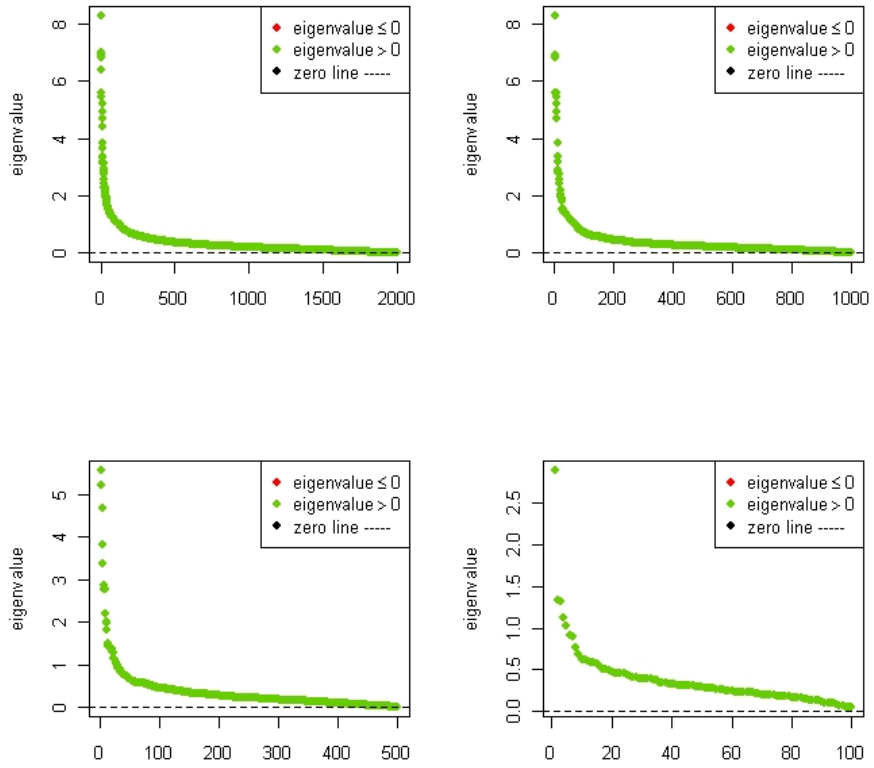

Figure 3.11: Plots illustrating the sorted eigenvalues of target $\mathbf{D}$ for the top 2000, 1000, 500 and 100 genes in the data set sCLLex.

The challenging task is to suggest a method which yields a positive definite covariance target without differing considerably from the original one. Inevitably, not only from the statisticians' point of view the question arises whether such a procedure can be reasonable. On the one hand, we consider important characteristics of the real covariance structure by incorporating external biological knowledge on gene functional groups. On the other hand, we manipulate the resulting covariance target in order to achieve positive definiteness. In our opinion, it is not worthwhile to regularize a covariance estimator by means of a covariance estimator which has to be regularized itself. One may ask provocatively: why should we incorporate external knowledge in the first step, being aware of the fact that we are forced to eliminate - possibly other - knowledge in the second step? In fact, in this thesis we will give some indication of the additional value of incorporating external biological knowledge into the classification process. For comparison purposes, we will employ the diagonal covariance target $\mathbf{D}$. First, however, we briefly present two approaches coping with the problem of indefiniteness. While the first one is applied to the not positive definite covariance target $\mathbf{T} = (t)_{ij}$, the second one is applied further in the shrinkage procedure, namely to the not positive definite shrinkage estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}} = \hat{\lambda}\mathbf{T} + (\mathbf{1} - \hat{\lambda})\mathbf{S}$.

- **The algorithm by Higham**

  One strategy to obtain a positive definite estimator of the covariance matrix is the application of the algorithm by Higham from 1988 to the sample covariance matrix $\mathbf{S} = (s)_{ij}$. The algorithm adjusts all eigenvalues to be larger than some prespecified threshold and thereby guarantees positive definiteness. The algorithm is carried out by the function `make.positive.definite()`, implemented in the open source `R` package `corpcor`. More details concerning the theory behind Higham's algorithm can be found in [24].

- **The inverse by Moore and Penrose**

  The inverse by Moore and Penrose describes a generalization of the standard matrix inverse, i.e. the 'generalized inverse', sometimes referred to as 'pseudoinverse'. The idea was introduced independently by Eliakim Hastings Moore in 1920 and Roger Penrose in 1955 [38]. More precisely, the so-

called 'Moore-Penrose pseudoinverse' can be applied to singular matrices and is based on the singular value decomposition. In our context, the covariance matrix $\mathbf{\Sigma_{SH(IP)}}$ can be decomposed into $\mathbf{\Sigma_{SH(IP)}} = \mathbf{UDV^T}$, whereas $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{D}$ is a square diagonal matrix containing only the positive singular values. The pseudoinverse $\mathbf{\Sigma^{-1}_{SH(IP)}}$ is then defined as $\mathbf{\Sigma^{-1}_{SH(IP)}} = \mathbf{VD^{-1}U^T}$. Note that it only requires the inversion of $\mathbf{D}$. Further, it can be shown that the pseudoinverse $\mathbf{\Sigma^{-1}_{SH(IP)}}$ is the shortest length least squares solution of $\mathbf{\Sigma_{SH(IP)}\Sigma^{-1}_{SH(IP)}} = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. Hence, it reduces to the standard matrix inverse where possible [40], which means that for non-singular matrices the pseudoinverse is equivalent to the standard inverse.
The computation can be carried out by means of the function `pseudoinverse()`, implemented in the open source `R` package `corpcor`. For more details concerning the theory behind the 'Moore-Penrose pseudoinverse' see [38].

Both procedures described above are dissatisfying from a statistician's point of view since the employment of numerical tricks seems to be inevitable. At this point of this work, we do not know yet which covariance targets yield better results in the context of classification. In fact, it is possible that the covariance targets incorporating external biological knowledge on gene functional groups will yield a smaller number of misclassifications, even if their employment requires numerical tricks.
Beyond the optimization of misclassification rates, however, we have to examine the substantial value of the input before interpreting the output. We will deal with this aspect in a critical way in Chapter 4, where we focus on discriminant analysis. Nevertheless, we have to decide for one technique we will use in this work. We will see that the estimator $\hat{\lambda}$ of the optimal shrinkage intensity λ depends on the entries of the covariance target $\mathbf{T} = (t)_{ij}$. Hence, it seems to be less reasonable to employ Higham's algorithm since we want to avoid $\hat{\lambda}$ being numerically manipulated. For this reason, we choose the 'Moore-Penrose pseudoinverse' for the analyses in this thesis, being aware of the fact that the estimate may be unstable due to the lack of observations. Further remarks concerning this issue will be provided in Chapter 4.

3.3 The optimal shrinkage intensity λ

In 3.2 we have studied the first challenge in the shrinkage process from the point of view of statisticians, namely the choice of the covariance target $\mathbf{T} = (t)_{ij}$. In this section we address the selection of the optimal shrinkage intensity referred to as λ. Note that in the literature, both the expression 'shrinkage intensity' and the expression 'regularization parameter' are used for λ. It is obvious that any choice of $\lambda \in [0,1]$ yields a compromise between $\mathbf{S}$ and $\mathbf{T}$, which results in infinitely many possibilities. The objective is obtaining an 'optimal' shrinkage intensity, whereas the term 'optimality' has to be defined. The usual way to obtain λ is determining it rather heuristically, for example by cross-validation [19]. Other well-established methods are based on Markov Chain Monte Carlo (MCMC) and the bootstrap [12]. The property these methods share is that they require computationally expensive procedures which constitutes the main drawback. In this thesis, we concentrate on the analytic determination of λ and its consistent estimation from the data, which were introduced by Ledoit and Wolf in 2003 [31]. This analytic approach is less known by biostatisticians, probably due to the original objective of Ledoit and Wolf who introduced this method in the context of portfolio selection. In 2005, Schäfer and Strimmer proposed the application to genomic data and simplified the consistent estimation of the shrinkage intensity λ [41, 40]. In the following, we first illustrate the analytic derivation of λ. Subsequently, we deal with the consistent estimation of λ since it depends on unobservables and thus cannot be calculated straightforward. We will see that the technique is very general since it is applicable to a wide range of covariance targets $\mathbf{T} = (t)_{ij}$, being constrained due to the positive definiteness requirement.

3.3.1 Analytical derivation of the optimal shrinkage intensity

In a nutshell, the optimal shrinkage intensity λ is considered from a decision theoretic perspective, which in particular means [37, 4]:

• A ***loss function*** $L(\cdot)$ is selected.

Definition 5 (Loss function) *A loss function is a mapping $L(\cdot)$, for which it holds:*

$$\begin{aligned} L(\cdot): \quad \hat{\Theta} \times \Theta &\longrightarrow \mathbb{R} \\ (\hat{\theta}, \theta) &\longmapsto L(\hat{\theta}, \theta), \end{aligned}$$

where $\hat{\Theta}$ is the space of estimates and Θ is the space of true parameters. It usually holds: $\hat{\Theta} = \mathbb{R}$ and $\Theta = \mathbb{R}$.

• λ is chosen such that the expectation of the loss with respect to the data, i.e. the ***risk*** $R(\cdot) = \boldsymbol{E}(L(\cdot))$ of the shrinkage estimator, is minimized:

$$R(\lambda) = E(L(\lambda)) \xrightarrow{\lambda} \min. \tag{3.4}$$

If $L_i(\cdot) = (\sigma_{i_{\mathbf{SH(IP)}}} - \sigma_i)^2$, i.e. the quadratic loss function, it follows:

$$R(\lambda) = E(L(\lambda)) \;=\; E\left(\sum_{i=1}^{p} (\sigma_{i_{\mathbf{SH(IP)}}} - \sigma_i)^2\right) \xrightarrow{\lambda} \min. \tag{3.5}$$

The loss function represents the objective according to which the shrinkage intensity is 'optimal'. Note that all existing shrinkage estimators from finite-sample statistical decision theory as well as the empirical Bayes approach of Frost and Savarino [20] break down in the $n \ll p$ case since the applied loss functions involve the inverse of the covariance matrix. In contrast, Ledoit and Wolf propose a loss function that does not depend on the inverse of the covariance matrix. It is the quadratic loss function, thus the intuitive quadratic measure of distance between the true and the estimated covariance matrices. Note that, in the matrix setting, the quadratic loss is based on the ***Frobenius norm*** [33].

Definition 6 (Frobenius norm) *The Frobenius norm of the $p \times p$ symmetric matrix $\boldsymbol{Z}$ with entries $(z_{ij})_{i,j=1,\ldots,p}$ is defined by:*

$$\|\boldsymbol{Z}\|_F^2 = \sum_{i=1}^{p}\sum_{j=1}^{p} z_{ij}^2.$$

In 3.1, we pointed out the distribution-freeness of the covariance estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ since it is not necessary to specify any underlying distributions. In fact, assuming merely the existence of the first two moments of the distributions of $\mathbf{T} = (t)_{ij}$ and $\mathbf{S} = (s)_{ij}$, it follows for the risk function:

$$\begin{aligned}
R(\lambda) &= E(L(\lambda)) \\
&= E\left(\left\|\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}} - \mathbf{\Sigma}\right\|_{\mathrm{F}}^2\right) \\
&= E\left(\|\lambda\mathbf{T} + (1-\lambda)\mathbf{S} - \mathbf{\Sigma}\|_{\mathrm{F}}^2\right) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} E(\lambda t_{ij} + (1-\lambda)s_{ij} - \sigma_{ij})^2 \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} \underbrace{Var(\lambda t_{ij} + (1-\lambda)s_{ij}) + [E(\lambda t_{ij} + (1-\lambda)s_{ij} - \sigma_{ij})]^2}_{= MSE(\lambda t_{ij} + (1-\lambda)s_{ij})} \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} \lambda^2 Var(t_{ij}) + (1-\lambda)^2 Var(s_{ij}) + 2\lambda(1-\lambda)Cov(t_{ij}, s_{ij}) \\
&\quad + [\lambda E(t_{ij} - s_{ij}) + \underbrace{E(s_{ij} - \sigma_{ij})}_{= Bias(s_{ij})}]^2. \qquad (3.6)
\end{aligned}$$

In 3.1, we pointed out without further explanations that $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ has guaranteed

minimum mean squared error, which results from the quadratic loss function [4]. For scientists who are not familiar with statistical decision theory this might be initially surprising, but the coherence becomes clear in a straightforward way as shown above. Thus it appears why the quadratic loss is the mostly applied loss function: since it results in the mean squared error for biased estimators and in the variance for unbiased ones, it is very beneficial concerning statistical questions. Note further that the quadratic loss function is symmetric, which sometimes might be of relevance. For the interested reader we recommend the lecture notes on statistical decision theory by Augustin, which provide a comprehensive overview of decision theoretic concepts [4].
In order to obtain an optimal shrinkage intensity λ, we now minimize analytically the risk $R(L(\lambda))$ of the form from Eq. 3.6 with respect to λ:

$$\begin{aligned} R'(\lambda) &= \frac{\partial R(\lambda)}{\partial \lambda} \\ &= 2\sum_{i=1}^{p}\sum_{j=1}^{p} \lambda Var(t_{ij}) - (1-\lambda)Var(s_{ij}) + (1-2\lambda)Cov(t_{ij}, s_{ij}) \\ &\quad +\lambda[E(t_{ij}-s_{ij})]^2 + E(t_{ij}-s_{ij})Bias(s_{ij}). \end{aligned} \tag{3.7}$$

$$\begin{aligned} R''(\lambda) &= \frac{\partial R'(\lambda)}{\partial \lambda} \\ &= 2\sum_{i=1}^{p}\sum_{j=1}^{p} \underbrace{Var(t_{ij}) + Var(s_{ij}) - 2Cov(t_{ij}, s_{ij})}_{=Var(t_{ij}-s_{ij})} + [E(t_{ij}-s_{ij})]^2 \\ &= 2\sum_{i=1}^{p}\sum_{j=1}^{p} \underbrace{Var(t_{ij}-s_{ij}) + [E(t_{ij}-s_{ij})]^2}_{>0}. \end{aligned} \tag{3.8}$$

$$
\begin{aligned}
R'(\lambda) &\stackrel{!}{=} 0 \\
&\Leftrightarrow \lambda \left[\sum_{i=1}^{p} \sum_{j=1}^{p} \underbrace{Var(t_{ij}) + Var(s_{ij}) - 2Cov(t_{ij}, s_{ij})}_{= Var(t_{ij} - s_{ij})} + [E(t_{ij} - s_{ij})]^2 \right] \\
&\quad + \sum_{i=1}^{p} \sum_{j=1}^{p} Cov(t_{ij}, s_{ij}) - Var(s_{ij}) + E(t_{ij} - s_{ij}) Bias(s_{ij}) \\
&\quad = 0 \\
&\Leftrightarrow \frac{\sum_{i=1}^{p} \sum_{j=1}^{p} Var(s_{ij}) - Cov(t_{ij}, s_{ij}) - E(t_{ij} - s_{ij}) Bias(s_{ij})}{\sum_{i=1}^{p} \sum_{j=1}^{p} Var(t_{ij} - s_{ij}) + [E(t_{ij} - s_{ij})]^2} \\
&\quad = \lambda. \qquad (3.9)
\end{aligned}
$$

Since $Var(t_{ij} - s_{ij}) = [E(t_{ij} - s_{ij})^2] - [E(t_{ij} - s_{ij})]^2$, it follows for $\lambda = \lambda_{\text{opt}}$:

$$
\lambda = \frac{\sum_{i=1}^{p} \sum_{j=1}^{p} Var(s_{ij}) - Cov(t_{ij}, s_{ij}) - E(t_{ij} - s_{ij}) Bias(s_{ij})}{\sum_{i=1}^{p} \sum_{j=1}^{p} [E(t_{ij} - s_{ij})^2]}. \qquad (3.10)
$$

Note that $R''(\lambda)$ is always positive, i.e. λ is a minimum of the risk function $R'(\lambda)$. Note further that the existence and the uniqueness of λ can be shown, which is illustrated in detail in the literature by Ledoit and Wolf. Moreover, since the sample covariance matrix $\mathbf{S} = (s)_{ij}$ is an unbiased estimator, Eq. 3.10 reduces to:

$$
\lambda = \frac{\sum_{i=1}^{p} \sum_{j=1}^{p} Var(s_{ij}) - Cov(t_{ij}, s_{ij})}{\sum_{i=1}^{p} \sum_{j=1}^{p} [E(t_{ij} - s_{ij})^2]}. \qquad (3.11)
$$

In this chapter, we concentrate on the sample covariance matrix $\mathbf{S} = (s)_{ij}$ as unbiased estimator of the covariance. Therefore, we use Eq. 3.11 for our calculations in the sequel. However, Eq. 3.10 points out that the analytical determination of the optimal shrinkage intensity, for which minimum mean squared error of the resulting shrinkage estimator is achieved, is rather general than restricted to unbiased estimators. In the following, we outline further remarks on the optimal shrinkage intensity λ and how it is chosen:

- We see in Eq. 3.11 that the optimal shrinkage intensity depends on the correlation between the estimation error of $\mathbf{S} = (s)_{ij}$ and of $\mathbf{T} = (t)_{ij}$. Intuitively, if the two are positively correlated, combining them yields a negligible benefit. Conversely, if the two are negatively correlated, a combination of them appears to be beneficial. In other words, if both are positively correlated the weight put on the shrinkage target decreases, whereas it increases if both are negatively correlated. Note that the introduction of this correlation term resolves an inconsistency which arises in empirical Bayesian approaches. Here, the prior is estimated from the sample data, assuming that this prior is independent from the sample data at the same time. Ledoit and Wolf explicitly take into account the correlation between prior and sample information through $\mathrm{Cov}(t_{ij}, s_{ij})$. Thus, they adjust for the two estimators both being inferred from the *same* data.

- Schäfer and Strimmer point out the possibility of generalizing the concept to multiple targets, which means that each target is assigned its own shrinkage intensity. For instance, if the model parameters fall into two natural groups, each could have its own target and thus its own associated shrinkage intensity. Note that, in the extreme case, each parameter could have its own λ.

- Consider the formula for λ from Eq. 3.11. Thus it appears that it is of general nature since the explicit form of the covariance target $\mathbf{T} = (t)_{ij}$ is nowhere used. Ledoit and Wolf point out that the equation stays the same as long as $\mathbf{T}$ is an asymptotically biased estimator of the covariance matrix. In addition, we want to point out that it has to satisfy the positive definiteness requirement. As a result, *any* covariance target leads to a reduction of the mean squared

error, albeit it is very complex to obtain a feasible one in the sense of fulfilling the positive definiteness.

3.3.2 Estimation of the optimal shrinkage intensity

In the first part of this section we studied the analytical approach to the optimal shrinkage intensity λ, for which we derived the following analytical form (see Eq. 3.11):

$$\lambda = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} Var(s_{ij}) - Cov(t_{ij}, s_{ij})}{\sum_{i=1}^{p}\sum_{j=1}^{p}[E(t_{ij} - s_{ij})^2]}.$$

The difficulty of the analytic determinantion is that λ depends on unobservables. According to Ledoit and Wolf, we solve this difficulty by replacing the true optimal λ by a consistent estimator $\hat{\lambda}$. Schäfer and Strimmer point out the weakness of the consistency requirement, since consistency is an asymptotic property and a basic requirement of any sensible estimator. Hence, we follow the suggestion in Schäfer and Strimmer to simplify the consistent estimation of the shrinkage intensity λ by replacing all expectations, variances, and covariances by their unbiased sample counterparts. Thus it follows:

$$\hat{\lambda} = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} \widehat{Var}(s_{ij}) - \widehat{Cov}(t_{ij}, s_{ij})}{\sum_{i=1}^{p}\sum_{j=1}^{p}(t_{ij} - s_{ij})^2}. \tag{3.12}$$

Since in finite samples it is possible that $\hat{\lambda} \notin [0,1]$, i.e. $\hat{\lambda} < 0$ or $\hat{\lambda} > 1$, we truncate the estimated intensity according to both Ledoit and Wolf and Schäfer and Strimmer by using $\hat{\lambda}^* = \max(0, \min(1, \hat{\lambda}))$ in the process of implementation.
Consider once again Eq. 3.12. In order to compute the estimator $\hat{\lambda}^*$ of the optimal shrinkage intensity, it is necessary to estimate the components of the given formula which in particular are:

$\widehat{Var}(s_{ij})$: variances of the individual entries of $\mathbf{S} = (s)_{ij}$

$\widehat{Cov}(t_{ij}, s_{ij})$: covariances between the individual entries of $\mathbf{T} = (t)_{ij}$ and the individual entries of $\mathbf{S} = (s)_{ij}$.

Note that $(t_{ij} - s_{ij})^2$, i.e. the quadratic distance between the individual entries of $\mathbf{T} = (t)_{ij}$ and the individual entries of $\mathbf{S} = (s)_{ij}$, can be calculated in a straightforward way since all required terms are given. In the following, we explicitly address the estimation of $Var(s_{ij})$ and $Cov(t_{ij}, s_{ij})$, whereas the explanations of the next two paragraphs are, with minor modifications, adopted from Schäfer and Strimmer [41].

Useful formulae

Let x_{ki} be the k-th observation of the variable X_i and $\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki}$ its empirical mean. Now set $w_{kij} = (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$ and $\bar{w}_{ij} = \frac{1}{n}\sum_{k=1}^{n} w_{kij}$. Then the unbiased empirical covariance equals:

$$\widehat{Cov}(x_i, x_j) = s_{ij} = \frac{n}{n-1}\bar{w}_{ij}. \tag{3.13}$$

Correspondingly, the variance is:

$$\widehat{Var}(x_i) = s_{ii} = \frac{n}{n-1}\bar{w}_{ii}. \tag{3.14}$$

Estimation of Var($\mathbf{s_{ij}}$)

The empirical unbiased variances of the individual entries of $\mathbf{S} = (s)_{ij}$ are computed in a similar fashion as described above. Thus it follows:

$$\begin{aligned}
\widehat{Var}(s_{ij}) &\underbrace{=}_{\text{Eq. 3.13}} \widehat{Var}\left(\frac{n}{n-1}\bar{w}_{ij}\right) \\
&= \frac{n^2}{(n-1)^2}\,\widehat{Var}(\bar{w}_{ij}) \\
&= \frac{n}{(n-1)^2}\,\widehat{Var}(w_{ij}) \\
&= \frac{n}{(n-1)^3}\,\sum_{k=1}^{n}(w_{kij}-\bar{w}_{ij})^2. \qquad (3.15)
\end{aligned}$$

Correspondingly, it follows for the covariances which become necessary for the next paragraph:

$$\widehat{Cov}(s_{ij}, s_{lm}) = \frac{n}{(n-1)^3}\,\sum_{k=1}^{n}(w_{kij}-\bar{w}_{ij})(w_{klm}-\bar{w}_{lm}). \qquad (3.16)$$

Schäfer and Strimmer point out that moments of higher order than $\widehat{Var}(s_{ij})$ are neglected in estimating the optimal shrinkage intensity $\hat{\lambda}^*$. Moreover, this procedure treats the estimated variances as constants and hence introduces an error which, however, is negligible.

Estimation of $\mathbf{Cov(t_{ij}, s_{ij})}$

The derivation of an estimator $\widehat{Cov}(t_{ij}, s_{ij})$ of the covariances between the individual entries of $\mathbf{T} = (t)_{ij}$ and the individual entries of $\mathbf{S} = (s)_{ij}$ turns out to be rather complex. In fact, $\widehat{Cov}(t_{ij}, s_{ij})$ becomes only relevant for the covariance targets **E** and **F** and thus for the covariance targets **F***, **G** and **G*** introduced in 3.2.2. In [33], Ledoit and Wolf give an expression for $\widehat{Cov}(t_{ij}, s_{ij})$ with regard to the 'constant

correlation model', referred to as target **F** in this work. In our opinion, however, a more detailed derivation is not only beneficial for completeness reasons. In addition, it improves considerably the understanding and thereby diminishes the vagueness of the given formula. For this purpose, we want to contribute by showing that the following formula given by Ledoit and Wolf holds asymptotically:

$$\widehat{Cov}(t_{ij}, s_{ij}) = \frac{\bar{r}}{2} \left(\sqrt{\frac{s_{jj}}{s_{ii}}} \widehat{Cov}(s_{ii}, s_{ij}) + \sqrt{\frac{s_{ii}}{s_{jj}}} \widehat{Cov}(s_{jj}, s_{ij}) \right), \tag{3.17}$$

where $\bar{r}$ is the average of sample correlations of the covariance target. For example, $\bar{r} = 0$ results from the application of target **D** and $\bar{r} = 1$ from the application of target **E**.
Note that by showing Eq. 3.17 it becomes possible to deduce $\widehat{Cov}(t_{ij}, s_{ij})$ for any other presented covariance target. We want to point out that we do not claim neither absolute accuracy nor completeness from a mathematical point of view. In fact, our objective is to present an outline of the derivation of Eq. 3.17.

Let $\mathbf{T} = (t)_{ij}$, $i, j = 1, ..., p$, be the very general covariance target **F** described in 3.2. Let further be $\mathbf{S} = (s)_{ij}$, $i, j = 1, ..., p$, the sample covariance matrix of the observations $x_{k1}, ..., x_{kp}$, $k = 1, ..., n$. According to Ledoit and Wolf it holds: the covariance $Cov(t_{ij}, s_{ij})$ of the individual entries of $\mathbf{T} = (t)_{ij}$ and the individual entries of $\mathbf{S} = (s)_{ij}$ can be estimated by the term $\frac{\bar{r}}{2} \cdot \left(\sqrt{\frac{s_{jj}}{s_{ii}}} \widehat{Cov}(s_{ii}, s_{ij}) + \sqrt{\frac{s_{ii}}{s_{jj}}} \widehat{Cov}(s_{jj}, s_{ij}) \right)$, where $\bar{r}$ is the average of sample correlations of the covariance target. Consider $\widehat{Cov}(t_{ij}, s_{ij})$. This term can be written as $\widehat{Cov}(\bar{r}\sqrt{s_{ii}s_{jj}}, s_{ij})$. In order to show that Eq. 3.17 holds, we carry out the following steps:

1. We consider $\sqrt{s_{ii}s_{jj}}$ as a function $f(s_{ii}, s_{jj}) = \sqrt{s_{ii}s_{jj}}$, where $i, j = 1, ..., p$. We approximate this function by applying Taylor approximation in several variables [1].

2. We replace $\sqrt{s_{ii}s_{jj}}$ by the approximation obtained in 1. Subsequently, we calculate the resulting term for $\widehat{Cov}(\bar{r}\sqrt{s_{ii}s_{jj}}, s_{ij})$.

Definition 7 (Taylor series in the one-dimensional case) *The Taylor series of a real or complex function f(x), whereas f(x) is infinitely differentiable in a neighbourhood of a real or complex number a, is the following power series:*

$$T(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots . \quad (3.18)$$

Equation 3.18 can be written in a more compact form as follows:

$$T(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n, \quad (3.19)$$

where n! denotes the factorial of n and $f^{(n)}(a)$ denotes the n-th derivative of f evaluated at the point a. The zeroth derivative of f is defined to be f itself. Both $(x-a)^0$ and 0! are defined to be 1.

Definition 8 (Taylor series in the multidimensional case) *The Taylor series considered in Definition 7 can be generalized to the multidimensional case, i.e. $\boldsymbol{x} = (x_1, x_2, x_3, ..., x_s)$. The form of $T(\boldsymbol{x}) = T(x_1, x_2, x_3, ..., x_s)$ is as follows:*

$$\begin{aligned} T(\boldsymbol{x}) &= T(x_1, x_2, x_3, ..., x_s) \\ &= \sum_{n_1=0}^{\infty} \cdots \sum_{n_s=0}^{\infty} \frac{(x_1-a_1)^{n_1} \cdots (x_s-a_s)^{n_s}}{n_1! \cdots n_s!} \left(\frac{\partial^{n_1+\dots+n_s} f}{\partial x_1^{n_1} \cdots \partial x_s^{n_s}} \right) (a_1, ..., a_s). \end{aligned} \quad (3.20)$$

A second-order Taylor series expansion of a scalar-valued function of more than one variable can be compactly written as:

$$\begin{aligned} T(\boldsymbol{x}) &= T(x_1, x_2, x_3, ..., x_s) \\ &= f(\boldsymbol{a}) + (\boldsymbol{x}-\boldsymbol{a})^T Df(\boldsymbol{a}) + \frac{1}{2!}(\boldsymbol{x}-\boldsymbol{a})^T\{D^2 f(\boldsymbol{a})\}(\boldsymbol{x}-\boldsymbol{a}) + \cdots, \end{aligned} \quad (3.21)$$

where $\boldsymbol{x} = (x_1, x_2, x_3, ..., x_s)$, $\boldsymbol{a} = (a_1, a_2, a_3, ..., a_s)$ *and* $Df(\boldsymbol{a})$ *is the gradient of* f *evaluated at* $\boldsymbol{x} = \boldsymbol{a}$ *and* $D^2 f(\boldsymbol{a})$ *is the Hessian matrix.*

Knowing the terms and definitions now, we carry out the first step. We consider $\sqrt{s_{ii}s_{jj}}$ as a function $f(s_{ii}, s_{jj}) = \sqrt{s_{ii}s_{jj}}$, where $i, j = 1, ..., p$. We approximate this function by applying Taylor expansion in two variables, namely s_{ii} and s_{jj}. We are interested in $f(s_{ii}, s_{jj}) = \sqrt{s_{ii}s_{jj}}$ in the neighbourhood $(\sigma_{ii}, \sigma_{jj})$. Further, we approximate $f(s_{ii}, s_{jj})$ linearly, i.e. we apply a first order Taylor series expansion. It follows:

$$f(s_{ii}, s_{jj}) \approx f(\sigma_{ii}, \sigma_{jj}) + \frac{\partial f}{\partial s_{ii}}(\sigma_{ii}, \sigma_{jj})\,(s_{ii} - \sigma_{ii}) + \frac{\partial f}{\partial s_{jj}}(\sigma_{ii}, \sigma_{jj})\,(s_{jj} - \sigma_{jj}).$$

Since $f(s_{ii}, s_{jj}) = \sqrt{s_{ii}s_{jj}}$, it holds:

$$\begin{aligned} \sqrt{s_{ii}, s_{jj}} &\approx \sqrt{\sigma_{ii}\sigma_{jj}} + \frac{\partial\sqrt{\sigma_{ii}\sigma_{jj}}}{\partial s_{ii}}\,(s_{ii} - \sigma_{ii}) + \frac{\partial\sqrt{\sigma_{ii}\sigma_{jj}}}{\partial s_{jj}}\,(s_{jj} - \sigma_{jj}) \\ &\approx \sqrt{\sigma_{ii}\sigma_{jj}} + \frac{1}{2}\frac{\sqrt{\sigma_{jj}}}{\sqrt{\sigma_{ii}}}\,(s_{ii} - \sigma_{ii}) + \frac{1}{2}\frac{\sqrt{\sigma_{ii}}}{\sqrt{\sigma_{jj}}}\,(s_{jj} - \sigma_{jj}) \\ &\approx \sqrt{\sigma_{ii}\sigma_{jj}} + \frac{1}{2}\frac{\sqrt{\sigma_{jj}}}{\sqrt{\sigma_{ii}}}s_{ii} - \frac{1}{2}\sqrt{\sigma_{ii}\sigma_{jj}} + \frac{1}{2}\frac{\sqrt{\sigma_{ii}}}{\sqrt{\sigma_{jj}}}s_{jj} - \frac{1}{2}\sqrt{\sigma_{ii}\sigma_{jj}} \\ &\approx \frac{1}{2}\left(\frac{\sqrt{\sigma_{jj}}}{\sqrt{\sigma_{ii}}}s_{ii} + \frac{\sqrt{\sigma_{ii}}}{\sqrt{\sigma_{jj}}}s_{jj}\right). \end{aligned} \quad (3.22)$$

Equation 3.22 yields an approximation for $\sqrt{s_{ii}s_{jj}}$ which we utilize for deriving a term for $\widehat{Cov}(\bar{r}\sqrt{s_{ii}s_{jj}}, s_{ij})$ in the second step:

$$
\begin{aligned}
& \widehat{Cov}(\bar{r}\sqrt{s_{ii}s_{jj}}, s_{ij}) \\
\underbrace{\approx}_{\text{Eq. 3.22}} \quad & \widehat{Cov}\left[\frac{1}{2}\bar{r}\left(\frac{\sqrt{\sigma_{jj}}}{\sqrt{\sigma_{ii}}}s_{ii} + \frac{\sqrt{\sigma_{ii}}}{\sqrt{\sigma_{jj}}}s_{jj}\right), s_{ij}\right] \\
\underbrace{\approx}_{\text{Cov(X+Y,Z)=Cov(X,Z)+Cov(Y,Z)}} \quad & \widehat{Cov}\left(\frac{1}{2}\bar{r}\,\frac{\sqrt{\sigma_{jj}}}{\sqrt{\sigma_{ii}}}s_{ii}, s_{ij}\right) + \widehat{Cov}\left(\frac{1}{2}\bar{r}\,\frac{\sqrt{\sigma_{ii}}}{\sqrt{\sigma_{jj}}}s_{jj}, s_{ij}\right) \\
\underbrace{\approx}_{\text{Cov(aX,bY)=abCov(X,Y)}} \quad & \frac{1}{2}\bar{r}\,\frac{\sqrt{\sigma_{jj}}}{\sqrt{\sigma_{ii}}}\widehat{Cov}(s_{ii}, s_{ij}) + \frac{1}{2}\bar{r}\,\frac{\sqrt{\sigma_{ii}}}{\sqrt{\sigma_{jj}}}\widehat{Cov}(s_{jj}, s_{ij}) \\
\approx \quad & \frac{1}{2}\bar{r}\,\left[\frac{\sqrt{\sigma_{jj}}}{\sqrt{\sigma_{ii}}}\widehat{Cov}(s_{ii}, s_{ij}) + \frac{\sqrt{\sigma_{ii}}}{\sqrt{\sigma_{jj}}}\widehat{Cov}(s_{jj}, s_{ij})\right] \\
\underbrace{\approx}_{\text{E}(\mathbf{S})=\,\Sigma} \quad & \frac{1}{2}\bar{r}\,\left[\frac{\sqrt{s_{jj}}}{\sqrt{s_{ii}}}\widehat{Cov}(s_{ii}, s_{ij}) + \frac{\sqrt{s_{ii}}}{\sqrt{s_{jj}}}\widehat{Cov}(s_{jj}, s_{ij})\right]. \qquad (3.23)
\end{aligned}
$$

Thus it appears that Eq. 3.23 corresponds to Eq. 3.17 which is the formula given by Ledoit and Wolf, whereas both $\widehat{Cov}(s_{ii}, s_{ij})$ and $\widehat{Cov}(s_{jj}, s_{ij})$ can be computed according to Eq. 3.16.

□

3.3.3 Shrinkage of covariances versus shrinkage of correlations

Having studied the choice of the covariance target $\mathbf{T} = (t)_{ij}$ and the analytical derivation of the optimal shrinkage intensity $\lambda \in [0, 1]$ in the previous work, we now constrain our attention on the interdependence of both results from above. In a nutshell, the main issues of 3.2.1 concerning the properties of the covariance targets are as follows:

- The shrinkage targets can be divided into two classes.
- The first class comprises target **A** ('diagonal, unit variance'), target **B** ('diagonal, common variance') and target **C** ('common (co)variance'), which are all extremely low-dimensional (0 to 2 free parameters), thus highly structured. The resulting covariance estimators shrink all components of the sample covariance matrix, i.e. both the diagonal and the off-diagonal entries are shrunken.
- The second class comprises target **D** ('diagonal, unequal variance'), target **E** ('perfect positive correlation') and target **F** ('constant correlation'), which are comparatively parameter-rich. The resulting covariance estimators only shrink the off-diagonal elements of **S**.
- As a consequence, the parameters of the covariance matrix fall into two classes, which both are treated differently in the shrinkage process.

Schäfer and Strimmer point out that this clear separation of the diagonal and the off-diagonal elements suggests, for shrinking purposes, to parameterize the covariance matrix in terms of variances and correlations rather than in variances and covariances, i.e. $\sigma_{ij} = r_{ij}\sqrt{\sigma_{ii}\sigma_{jj}}$. Thus it appears that it is possible to shrink only the correlations rather than the covariances, which is intuitively far more adequate for the covariance targets **D**, **E** and **F**. Moreover, the fact that shrinkage is applied to the correlations has the clear advantage that the off-diagonal elements determining the shrinkage intensity are all on the same scale. Note that this is not the case if we work with covariances; the (co)variance determines the scale, whereas the correlation determines the dimensionless linear structure of connection. In [41, 40], Schäfer and Strimmer propose such a parameterization of target **D** into variances and correlations, which yields the formula $\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(r_{ij})}{\sum_{i \neq j} r_{ij}^2}$ instead of $\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij})}{\sum_{i \neq j} s_{ij}^2}$. In this thesis, we work with modified versions of target **F**, which is one of the co-

variance targets shrinking only the off-diagonal elements of the sample covariance matrix. Hence, we apply the parameterization described above to target **F**, and we obtain the simplified formula for the shrinkage intensity $\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(r_{ij}) - \bar{r} f_{ij}}{\sum_{i \neq j} (r_{ij} - \bar{r})^2}$ (where $f_{ij} = \frac{1}{2}\{\widehat{Cov}(r_{ii}, r_{ij}) + \widehat{Cov}(r_{jj}, r_{ij})\}$) instead of $\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) - \bar{r} f_{ij}}{\sum_{i \neq j} (s_{ij} - \bar{r}\sqrt{s_{ii} s_{jj}})^2}$ (where $f_{ij} = \frac{1}{2}\{\sqrt{\frac{s_{jj}}{s_{ii}}} \widehat{Cov}(s_{ii}, s_{ij}) + \sqrt{\frac{s_{ii}}{s_{jj}}} \widehat{Cov}(s_{jj}, s_{ij})\}$). A complete overview of the covariance targets and their associated estimators of the optimal shrinkage intensity is given in 3.4.

In order to account for the natural grouping of the covariance targets, the expression for $\hat{\lambda}$ has to be modified as follows. The individual covariances s_{ij} have to be replaced by the individual correlations r_{ij}. Thus, in some cases the formula for $\hat{\lambda}$ can be simplified yet, for instance for the covariance target **F** as described above. Note that the calculation of the variance $\widehat{Var}(r_{ij})$ of the empirical correlation coefficients can be estimated similarly as the variance $\widehat{Var}(s_{ij})$ of the empirical covariance coefficients as described in 3.3.2: the concrete way to obtain $\widehat{Var}(r_{ij})$ is applying the formula for $\widehat{Var}(s_{ij})$, i.e. Eq. 3.15, to the *standardized* data matrix. This holds analogously for obtaining $\widehat{Cov}(r_{ij}, r_{lm})$, where applying the formula for $\widehat{Cov}(s_{ij}, s_{lm})$, i.e. Eq. 3.16, to the *standardized* data matrix yields the desired estimator.

3.4 Overview of the covariance targets and the associated estimators of the optimal shrinkage intensity

- **Target A:** 'diagonal, unit variance'; 0 estimated parameters

$$t_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) + \sum_i \widehat{Var}(s_{ii})}{\sum_{i \neq j} s_{ij}^2 + \sum_i (s_{ii} - 1)^2}$$

- **Target B:** 'diagonal, common variance'; 1 estimated parameter: ν

$$t_{ij} = \begin{cases} \nu = \text{avg}(s_{ii}) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) + \sum_i \widehat{Var}(s_{ii})}{\sum_{i \neq j} s_{ij}^2 + \sum_i (s_{ii} - \nu)^2}$$

- **Target C:** 'common (co)variance'; 2 estimated parameters: ν, c

$$t_{ij} = \begin{cases} \nu = \text{avg}(s_{ii}) & \text{if } i = j \\ c = \text{avg}(s_{ij}) & \text{if } i \neq j \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) + \sum_i \widehat{Var}(s_{ii})}{\sum_{i \neq j} (s_{ij} - c)^2 + \sum_i (s_{ii} - \nu)^2}$$

- **Target D:** 'diagonal, unequal variance'; p estimated parameters: s_{ii}

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij})}{\sum_{i \neq j} s_{ij}^2}$$

- **Target E:** 'perfect positive correlation'; p estimated parameters: s_{ii}

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) - f_{ij}}{\sum_{i \neq j} (s_{ij} - \sqrt{s_{ii} s_{jj}})^2} \quad \text{where} \quad f_{ij} = \frac{1}{2} \left\{ \sqrt{\frac{s_{jj}}{s_{ii}}} \widehat{Cov}(s_{ii}, s_{ij}) + \sqrt{\frac{s_{ii}}{s_{jj}}} \widehat{Cov}(s_{jj}, s_{ij}) \right\}$$

- **Target F:** 'constant correlation'; $p+1$ estimated parameters: s_{ii}, $\bar{r}$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}\sqrt{s_{ii}s_{jj}} & \text{if } i \neq j \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) - \bar{r} f_{ij}}{\sum_{i \neq j} (s_{ij} - \bar{r}\sqrt{s_{ii}s_{jj}})^2}$$

- **Target F*:** 'two constant correlations between genes: a negative and a positive one'; $p+2$ estimated parameters: s_{ii}, $\bar{r}_-$, $\bar{r}_+$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}_-\sqrt{s_{ii}s_{jj}} & \text{if } i \neq j, cor(i,j) < 0 \\ \bar{r}_+\sqrt{s_{ii}s_{jj}} & \text{if } i \neq j, cor(i,j) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) - I(cor(i,j)<0)\ \bar{r}_- f_{ij} - I(cor(i,j)>0)\ \bar{r}_+ f_{ij}}{\sum_{i \neq j} (s_{ij} - I(cor(i,j)<0)\ \bar{r}_-\sqrt{s_{ii}s_{jj}} - I(cor(i,j)>0)\ \bar{r}_+\sqrt{s_{ii}s_{jj}})^2}$$

- **Target G:** 'constant correlation between connected genes'; $p+1$ estimated parameters: s_{ii}, $\bar{r}$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}\sqrt{s_{ii}s_{jj}} & \text{if } i \neq j, i \sim j \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) - \sum_{i \sim j} \bar{r} f_{ij}}{\sum_{i \neq j} (s_{ij} - I(i \sim j)\ \bar{r}\sqrt{s_{ii}s_{jj}})^2}$$

- **Target G*:** 'two constant correlations between connected genes: a negative and a positive one'; $p+2$ estimated parameters: s_{ii}, $\bar{r}_-$, $\bar{r}_+$

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \bar{r}_- \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j, i \sim^{\ominus} j \\ \bar{r}_+ \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j, i \sim^{\oplus} j \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\lambda} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) - \sum_{i \sim^{\ominus} j} \bar{r}_- f_{ij} - \sum_{i \sim^{\oplus} j} \bar{r}_+ f_{ij}}{\sum_{i \neq j} (s_{ij} - I(i \sim^{\ominus} j)\, \bar{r}_- \sqrt{s_{ii} s_{jj}} - I(i \sim^{\oplus} j)\, \bar{r}_+ \sqrt{s_{ii} s_{jj}})^2}$$

where ν : average of sample variances

c : average of sample covariances

$\bar{r}$: average of sample correlations (for all genes in target **F** and only for the connected genes in target **G**)

$\bar{r}_-$: average of negative sample correlations (for all genes in target **F*** and only for the connected genes in target **G***)

$\bar{r}_+$: average of positive sample correlations (for all genes in target **F*** and only for the connected genes in target **G***).

Chapter 4

Linear discriminant analysis using $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$

Starting from the shrinkage estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ studied in the previous chapter, we now address its use in the special case of linear discriminant analysis. So far, we have only dealt with $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ under the assumption that the n observed $(1 \times p)$ predictor vectors $\boldsymbol{x}_k^T = (x_{k1}, ..., x_{kp})$, $k = 1, ..., n$, come from one homogeneous population. Within the scope of (linear) discriminant analysis, however, where the predictor vectors fall into groups or classes, the previous procedure is no more convenient.
In the first section of this chapter, we present and discuss two possible approaches to the pooled version of $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$, being referred to as $\hat{\mathbf{\Sigma}}^*_{\mathbf{SH(IP)}}$ in the remainder of this work. Subsequently, we apply the linear discriminant analysis 'via the SH(IP)' to the real-life data sets described in 1.3 and examine both the binary and the c-nary case, where $c > 2$. Finally, we discuss our method's results from different points of view and give some indication of the additional value of incorporating biological knowledge into the classification process in the way we proposed in this thesis.

4.1 $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ in the case of linear discriminant analysis

Let us consider Section 2.1. We have seen that the pooled empirical $(p \times p)$ covariance matrix $\mathbf{S}_{\mathbf{pool}}$ has the following form (see Eq. 2.11):

$$\mathbf{S_{pool}} = \frac{1}{n-c}\sum_{r=1}^{c}\sum_{k=1}^{n_r}(\boldsymbol{x}_{rk}-\bar{\boldsymbol{x}}_r)(\boldsymbol{x}_{rk}-\bar{\boldsymbol{x}}_r)^T,$$

where n_r is the number of observations in class r, $r = 1, ..., c$, $\bar{\boldsymbol{x}}_r$ is the $(p \times 1)$ mean vector for class r and $\boldsymbol{x}_{rk}$ is the $(p \times 1)$ vector of predictor variables corresponding to the k-th observation in class r. It can be easily seen that $\mathbf{S_{pool}}$ can be written as a weighted sum of the within-class covariance matrices, which in turn are estimated by the standard empirical covariance matrix [18, 48]:

$$\begin{aligned}
\mathbf{S_{pool}} &= \frac{1}{n-c}\sum_{r=1}^{c}\sum_{k=1}^{n_r}(\boldsymbol{x}_{rk}-\bar{\boldsymbol{x}}_r)(\boldsymbol{x}_{rk}-\bar{\boldsymbol{x}}_r)^T \\
&= \frac{1}{n-c}\sum_{r=1}^{c}\underbrace{\sum_{k=1}^{n_r}(\boldsymbol{x}_{rk}-\bar{\boldsymbol{x}}_r)(\boldsymbol{x}_{rk}-\bar{\boldsymbol{x}}_r)^T}_{(n_r-1)\mathbf{S}^{(r)}} \\
&= \frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)\mathbf{S}^{(r)}, \qquad (4.1)
\end{aligned}$$

where $\mathbf{S}^{(r)}$ denotes the standard unbiased $(p \times p)$ empirical covariance matrix for class r. However, in the high-dimensional setting where $\mathbf{S_{pool}}$ is no more suitable we need a pooled version of a regularized shrinkage estimator.

In Chapter 3, we have introduced the shrinkage estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ in a general framework, i.e. the observations were assumed to come from one homogeneous population. Thus, $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ can be regarded as the high-dimensional counterpart of the standard empirical covariance matrix $\mathbf{S}$ in this work. Here, however, we want to find a high-dimensional counterpart of the pooled empirical covariance matrix $\mathbf{S_{pool}}$ which means formulating a pooled version of $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$. Note that henceforth the term $\hat{\mathbf{\Sigma}}^*_{\mathbf{SH(IP)}}$ will stand for this pooled version unless otherwise emphasized. In summary:

	Standard covariance estimator	Pooled covariance estimator
$p \ll n$	$\mathbf{S} = \frac{1}{n-1}\sum_{k=1}^{n}(\boldsymbol{x}_k-\bar{\boldsymbol{x}})(\boldsymbol{x}_r-\bar{\boldsymbol{x}})^T$	$\mathbf{S_{pool}} = \frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)\mathbf{S}^{(r)}$
$n \ll p$	$\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}} = \hat{\lambda}\mathbf{T} + (1-\hat{\lambda})\mathbf{S}$	$\hat{\mathbf{\Sigma}}^*_{\mathbf{SH(IP)}} = \boxed{?}$

In the following, we propose two approaches to obtain $\hat{\mathbf{\Sigma}}^{*}_{\mathbf{SH(IP)}}$. While the first one could be considered to be naive, the second one turns out to be very tedious.

4.1.1 Approach 1: Pooling the within-class shrinkage estimators

Let us consider Eq. 4.1. Intuitively, it may be obvious to compute the pooled version of $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ according to the standard procedure demonstrated above. In particular, it follows:

$$\begin{aligned}
\hat{\mathbf{\Sigma}}^{*}_{\mathbf{SH(IP)}} &= \frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)\hat{\mathbf{\Sigma}}^{(r)}_{\mathbf{SH(IP)}} \\
&\underbrace{=}_{\text{Eq. 3.3}} \frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)\left[\hat{\lambda}_r\mathbf{T}^{(r)}+(1-\hat{\lambda}_r)\mathbf{S}^{(r)}\right] \\
&= \frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)\left[\mathbf{S}^{(r)}+\hat{\lambda}_r\underbrace{\left\{\mathbf{T}^{(r)}-\mathbf{S}^{(r)}\right\}}_{\mathbf{D}_r}\right], \qquad (4.2)
\end{aligned}$$

where $\hat{\mathbf{\Sigma}}^{(r)}_{\mathbf{SH(IP)}}$ denotes the shrinkage estimator for class r, $r = 1, ..., c$, $\mathbf{S}^{(r)}$ is the standard unbiased empirical covariance matrix for class r and $\mathbf{T}^{(r)}$ denotes the within-class covariance target for class r. Moreover, $\lambda_r \in [0, 1]$ is the shrinkage intensity for class r, i.e. the shrinkage intensities are calculated separately for each class. Note that Eq. 4.2 characterizes λ_r from another interesting point of view: it can be regarded as the weight put on the difference $\mathbf{D}_r$ between the covariance target $\mathbf{T}^{(r)}$ for class r and the sample covariance matrix $\mathbf{S}^{(r)}$ for class r.

Intuitively, it is clear that pooling the within-class shrinkage estimators $\hat{\mathbf{\Sigma}}^{(r)}_{\mathbf{SH(IP)}}$ as described above does not correspond to a pooled $\hat{\mathbf{\Sigma}}^{*}_{\mathbf{SH(IP)}}$ with only one shrinkage intensity. While both $\mathbf{T}^{(r)}$ and $\mathbf{S}^{(r)}$, $r = 1, ..., c$, can be pooled according to the well-known procedure in Eq. 4.1, pooling the estimated shrinkage intensities $\hat{\lambda}_r$ is far more complex. We will deal with this subject in 4.1.2. Nevertheless, the approach we presented is straightforward and thus convenient in practice.

4.1.2 Approach 2: Deriving the pooled shrinkage estimator with one shrinkage intensity

In 4.1.1, we pointed out the difficulty arising in case one is interested in a pooled version of the shrinkage estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$. This difficulty results from the fact that the shrinkage intensity has to be estimated and that the estimated $\hat{\lambda}$ we would obtain for the pooled shrinkage estimator $\hat{\mathbf{\Sigma}}^*_{\mathbf{SH(IP)}}$ seems not to correspond to pooling the $\hat{\lambda}_r$, $r = 1, ..., c$, according to the accepted procedure in Eq. 4.1. Since this does not hold for $\mathbf{T}^{(r)}$ and $\mathbf{S}^{(r)}$, we constrain our attention on the shrinkage intensity, being referred to as $\lambda_{\mathbf{pool}}$. Note that this part can be skipped since, in this work, the approach of choice will be the one presented in 4.1.1. For clarity's sake, we recommend to continue with 4.2 and to come back to this part at a later point.

In the following, we extend the estimation procedure for the optimal shrinkage intensity introduced by Ledoit and Wolf [31, 33, 32] and presented in 3.3.2 in this work from the standard to the pooled case. In particular, we draft the development of an estimator $\hat{\lambda}_{\mathbf{pool}}$ for $\lambda_{\mathbf{pool}}$. For this purpose, let us first list the results from 3.3.2, being helpful for the subsequent calculations:

a) **Eq. 3.11:** Analytical form of the optimal shrinkage intensity when $\mathbf{S} = (s)_{ij}$ is unbiased

$$\lambda = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} Var(s_{ij}) - Cov(t_{ij}, s_{ij})}{\sum_{i=1}^{p}\sum_{j=1}^{p}[E(t_{ij} - s_{ij})^2]}$$

b) **Eq. 3.12:** Consistent estimator of the optimal shrinkage intensity

$$\hat{\lambda} = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} \widehat{Var}(s_{ij}) - \widehat{Cov}(t_{ij}, s_{ij})}{\sum_{i=1}^{p}\sum_{j=1}^{p}(t_{ij} - s_{ij})^2}$$

c) **Eq. 3.13:** Alternative notation of the unbiased empirical covariance

Let x_{ki} be the k-th observation of the variable X_i and $\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki}$ its empirical mean. Now set $w_{kij} = (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$ and $\bar{w}_{ij} = \frac{1}{n}\sum_{k=1}^{n} w_{kij}$.

Then it holds:

$$\widehat{Cov}(x_i, x_j) = s_{ij} = \frac{n}{n-1}\bar{w}_{ij}$$

d) **Eq. 3.14:** Alternative notation of the unbiased empirical variance

$$\widehat{Var}(x_i) = s_{ii} = \frac{n}{n-1}\bar{w}_{ii}$$

e) **Eq. 3.15:** Estimator for $Var(s_{ij})$

$$\widehat{Var}(s_{ij}) \quad = \quad \frac{n}{(n-1)^3} \sum_{k=1}^{n} (w_{kij} - \bar{w}_{ij})^2$$

f) **Eq. 3.16:** Estimator for $Cov(s_{ij}, s_{lm})$

$$\widehat{Cov}(s_{ij}, s_{lm}) \quad = \quad \frac{n}{(n-1)^3} \sum_{k=1}^{n} (w_{kij} - \bar{w}_{ij})(w_{klm} - \bar{w}_{lm})$$

g) **Eq. 3.17:** Estimator for $Cov(t_{ij}, s_{ij})$

$$\widehat{Cov}(t_{ij}, s_{ij}) \quad = \quad \frac{\bar{r}}{2} \left(\sqrt{\frac{s_{jj}}{s_{ii}}} \widehat{Cov}(s_{ii}, s_{ij}) + \sqrt{\frac{s_{ii}}{s_{jj}}} \widehat{Cov}(s_{jj}, s_{ij}) \right),$$

where $\bar{r}$ is the average of sample correlations of the covariance target.

Let us now consider the pooled empirical covariance matrix $\mathbf{S}_{\mathbf{pool}}$. Then it follows for $\widehat{Var}(s_{ij_{\mathbf{pool}}})$:

$$
\begin{aligned}
& \widehat{Var}(s_{ij_{\mathbf{pool}}}) \\
\underbrace{=}_{\text{Eq. 3.2}} \quad & \widehat{Var}\left(\frac{1}{n-c}\sum_{r=1}^{c}\sum_{k=1}^{n_r}(x_{rki}-\bar{x}_{ri})(x_{rkj}-\bar{x}_{rj})\right) \\
= \quad & \widehat{Var}\left(\frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)s_{ij}^{(r)}\right) \\
\underbrace{=}_{\text{Eq. 3.13}} \quad & \widehat{Var}\left(\frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)\frac{n_r}{n_r-1}\bar{w}_{ijr}\right) \\
= \quad & \frac{1}{(n-c)^2}\widehat{Var}\left(\sum_{r=1}^{c}n_r\bar{w}_{ijr}\right) \\
\underbrace{=}_{*} \quad & \frac{1}{(n-c)^2}\left(\widehat{Var}(n_1\bar{w}_{ij1})+\ldots+\widehat{Var}(n_c\bar{w}_{ijc})\right) \\
= \quad & \frac{1}{(n-c)^2}\left(n_1^2\widehat{Var}(\bar{w}_{ij1})+\ldots+n_c^2\widehat{Var}(\bar{w}_{ijc})\right) \\
= \quad & \frac{1}{(n-c)^2}\left(n_1\widehat{Var}(w_{ij1})+\ldots+n_c\widehat{Var}(w_{ijc})\right) \\
= \quad & \frac{1}{(n-c)^2}\left(\frac{n_1}{n_1-1}\sum_{k=1}^{n_1}(w_{kij1}-\bar{w}_{ij1})^2+\ldots+\frac{n_c}{n_c-1}\sum_{k=1}^{n_c}(w_{kijc}-\bar{w}_{ijc})^2\right) \\
\underbrace{=}_{\text{Eq. 3.15}} \quad & \frac{1}{(n-c)^2}\sum_{r=1}^{c}\underbrace{\frac{n_r}{n_r-1}\sum_{k=1}^{n_r}(w_{kijr}-\bar{w}_{ijr})^2}_{(n_r-1)^2\widehat{Var}(s_{ij}^{(r)})} \\
= \quad & \frac{1}{(n-c)^2}\sum_{r=1}^{c}(n_r-1)^2\widehat{Var}(s_{ij}^{(r)}), \qquad (4.3)
\end{aligned}
$$

where $\bar{w}_{ijr}=\frac{1}{n_r}\sum_{k=1}^{n_r}w_{kijr}$, $w_{kijr}=(x_{kir}-\bar{x}_{ir})(x_{kjr}-\bar{x}_{jr})$ and $s_{ij}^{(r)}$ is the (ij)-th entry of the standard unbiased empirical covariance matrix for class r, $r=1,...,c$.
* Note that moments of higher order are neglected.

It appears from Eq. 4.3 that $\widehat{Var}(s_{ij_{\mathbf{pool}}})$ can be written by means of the empirical variances of the individual entries of the within-class covariance matrices $\mathbf{S}^{(r)}$ as defined by Eq. 3.15. Correspondingly, it follows for the covariances:

$$\widehat{Cov}(s_{ij_{\mathbf{pool}}}, s_{lm_{\mathbf{pool}}}) \underbrace{=}_{\text{Eq. 3.16}} \frac{1}{(n-c)^2} \sum_{r=1}^{c} (n_r - 1)^2 \widehat{Cov}(s_{ij}^{(r)}, s_{lm}^{(r)}). \quad (4.4)$$

Facing the second term to be estimated, i.e. $Cov(t_{ij_{\mathbf{pool}}}, s_{ij_{\mathbf{pool}}})$, we find that the calculations from above similarly apply, but yield a more complicated form. Eventually, we conclude that $\hat{\lambda}_{\mathbf{pool}}$ cannot be written as a weighted sum of the $\hat{\lambda}_r$, $r = 1, ..., c$. For the purpose of convenience, we prefer the approach presented in 4.1.1 in this thesis.

4.2 Application to real-life data

So far, we have studied the linear discriminant analysis from a rather theoretic point of view. In this section we focus on real cancer microarray data sets as described in 1.3 and examine the classification performance of the method proposed in this work. We extract the biological knowledge on gene functional groups from the database KEGG which we introduced in 2.3 and furthermore discuss the additional value of incorporating biological knowledge into the classification process.

4.2.1 Denotations and technical remarks

For the purpose of clarity and reproducibility, we first give an outline of the denotations we will use and of the methodical or technical details behind the results in 4.2.2 and 4.2.3, respectively. Let us consider these aspects in the order of their appearance in the whole classification procedure:

- **Data preparation**

  We use the two-class data sets Golub_Merge and sCLLex as well as the six-class data set ALL_a and the four-class data set ALL_b described in 1.3. We set aside the two-class data set ALL_c for the following reason: the two classes result from pooling together the ten classes from the original data set ALL. This drastic pooling, however, is attended by a severe loss of information. Hence, since we have two other two-class data sets available, we omit the data set ALL_c.

  Note that for computational reasons, we do not employ all variables (genes) of each data set, but perform a variable selection before. A classical mistake is to select variables (genes) as a preliminary step based on the whole data set and to build classification rules based on this reduced set of variables (genes). However, variable selection should be considered as a part of the construction of classification rules. Consequently, it should be carried out for each learning set separately, thus in each iteration of the classification procedure. Further details and studies concerning this topic can be found in [2, 7, 35, 42, 49]. As briefly depicted in 1.3, the `R` package `CMA` offers various methods performing variable selection for each learning set separately [43]. In particular, we use the method `GeneSelection()`. We choose an ordinary two sample t.test as concrete variable selection method. We generate the learning and test samples by employing the method `GenerateLearningsets()` and use a stratified five-fold cross-validation as evaluation scheme for the two-class data sets and a three-fold cross-validation otherwise, repeated ten times in order to achieve more stable results [10]. The concrete `R` code for the data preparation is available on the attached CD.

- **Linear discriminant analysis**

  We have implemented the variants of linear discriminant analysis proposed in this thesis, i.e. we have implemented the linear discriminant analysis using the shrinkage estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SH(IP)}}$ according to the scheme illustrated below, where $\mathbf{T} = \{\text{target } \mathbf{D} \wedge \text{target } \mathbf{G} \wedge \text{target } \mathbf{G}^* \wedge \text{target } \mathbf{F} \wedge \text{target } \mathbf{F}^*\}$:

$$\hat{\mathbf{\Sigma}}^{*}_{\mathbf{SH(IP)}} = \frac{1}{n-c}\sum_{r=1}^{c}(n_r-1)\underbrace{\left[\hat{\lambda}_r\mathbf{T}^{(r)} + (1-\hat{\lambda}_r)\mathbf{S}^{(r)}\right]}_{\hat{\mathbf{\Sigma}}^{(r)}_{\mathbf{SH(IP)}}}$$

see: Eq.3.3, Eq.3.12, Eq.3.13, Eq.3.14, Eq.3.15, Eq.3.16, Eq.3.17, Eq.4.2

$$\Downarrow$$

$$\hat{\mathbf{\Sigma}}^{*\,\mathbf{-1}}_{\mathbf{SH(IP)}} = \text{pseudoinverse}\Big(\hat{\mathbf{\Sigma}}^{*}_{\mathbf{SH(IP)}}\Big)$$

see: part 3.2.4 of Section 3.2

$$\Downarrow$$

$$\hat{d}_r(\boldsymbol{x}) = -\tfrac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}}_r)^T\,\hat{\mathbf{\Sigma}}^{*\,\mathbf{-1}}_{\mathbf{SH(IP)}}\,(\boldsymbol{x}-\bar{\boldsymbol{x}}_r)+\log(\hat{p}(r))$$

see: Eq.2.8, Eq.2.10

In words, we first compute the pooled shrinkage estimator from Eq. 4.2. Subsequently, we employ the (pseudo)inverse of this pooled shrinkage estimator in order to perform the linear discriminant analysis, whereas we use the function `pseudoinverse()` from the `R` package `corpcor` to obtain the pseudoinverse. As a result, this yields five variants of LDA 'via the SH(IP)', differing only in terms of the covariance target $\mathbf{T}$. Note that henceforth, we use the following abbreviations: rlda.TD, rlda.TG, rlda.TG*, rlda.TF and rlda.TF*. The `R` program carrying out these different variants of LDA will be outlined in Appendix A and may be inspected on the attached CD. It has been implemented such that it can be incorporated into the framework of the CMA package [43]. Thereby, the variants of LDA proposed in this thesis can be called in the CMA method `classification()` which carries out the classification by means of the learning and test sets as defined above.

- **Prediction accuracy and comparison of methods**

  Once the classification has been carried out for all iterations, i.e. for all learning and test sets, the CMA method `evaluation()` offers the calculation of a multiplicity of prediction accuracy measures [43]. In this thesis, we focus on the average misclassification rate, i.e. the average test error obtained for the test sets described above. Moreover, we ascertain both the sensitivity and the

specificity for the two-class data sets according to the explanations in 2.1.3 and 2.2.1.

In order to decide whether our variants of LDA work well on real data sets we have to compare it to existing classification methods. In this work, we choose both the diagonal linear discriminant analysis (DLDA) and the nearest shrunken centroids method (NSC) as competitors. While we perform a variable selection in the former, this is not necessary in the latter since the NSC method eliminates a lot of non-contributing variables (genes) itsself [15, 22]. Note that it is possible to call both methods in the CMA method `classification()`. Note further that for the NSC method the shrinkage parameter Δ is optimized over the grid $\{0.1, 0.25, 0.5, 1, 2, 5\}$. Additionally, we do not only constrain our attention on the comparison with the two competitors from above, but also focus on the comparison between the five variants of LDA proposed in this thesis. For this purpose, let us consider these methods. We point out that rlda.TG and rlda.TG* incorporate biological knowledge on gene functional groups. In contrast, the methods rlda.TD, rlda.TF and rlda.TF* do not embed external knowledge from databases. For instance, it is thus possible to contrast rlda.TG with rlda.TF and rlda.TG* with rlda.TF*, which gives some indication of the additional value of incorporating biological knowledge into the classification process. Further, contrasting rlda.TD with the other variants allows general statements about the additional value of accounting for correlations between genes. We will deal with these aspects in 4.2.2 and 4.2.3, respectively.

4.2.2 The binary case: $c = 2$

Consider the explanations in 4.2.1. Let us now report - for the two-class data sets Golub_Merge and sCLLex - the results obtained for the different variants of LDA 'via the SH(IP)' as well as for the competitors DLDA and NSC method. For each variant of LDA the top 50, 100, 200 and 500 genes are employed and the results are compared. Note that the NSC method is carried out once for the whole data set since it eliminates non-contributing genes itsself. For clarity's sake, we mark the best and the second best result by $\star$ and $\bullet$, respectively.

Application to the Golub_Merge data

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	50	0.043 ●	0.916	0.979
rlda.TG	50	0.045	0.912	0.979
rlda.TG*	50	0.043 ●	0.916	0.979
rlda.TF	50	0.043 ●	0.932 ●	0.971
rlda.TF*	50	0.254	0.652	0.796
dlda	50	0.057	0.844	0.996 ●
nsc	7 129	0.021 ⋆	0.940 ⋆	1.000 ⋆

Table 4.1: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 50 genes (except for nsc) of the two-class data Golub_Merge (n=72).

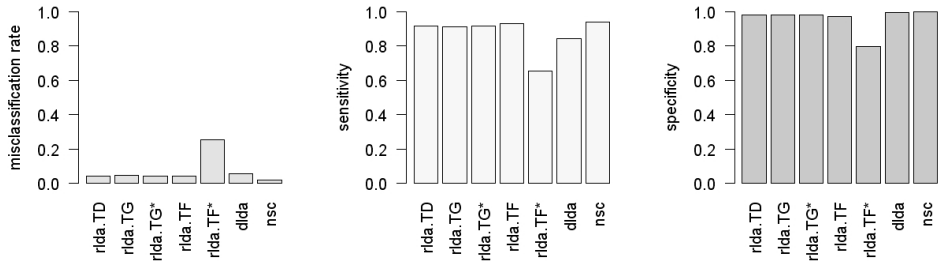

Figure 4.1: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 50 genes (except for nsc) of the two-class data Golub_Merge.

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	100	0.028 ●	0.960 ★	0.979 ●
rlda.TG	100	0.029	0.956 ●	0.979 ●
rlda.TG*	100	0.033	0.944	0.979 ●
rlda.TF	100	0.034	0.960 ★	0.969
rlda.TF*	100	0.382	0.516	0.672
dlda	100	0.042	0.880	1.000 ★
nsc	7 129	0.021 ★	0.940	1.000 ★

Table 4.2: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 100 genes (except for nsc) of the two-class data Golub_Merge (n=72).

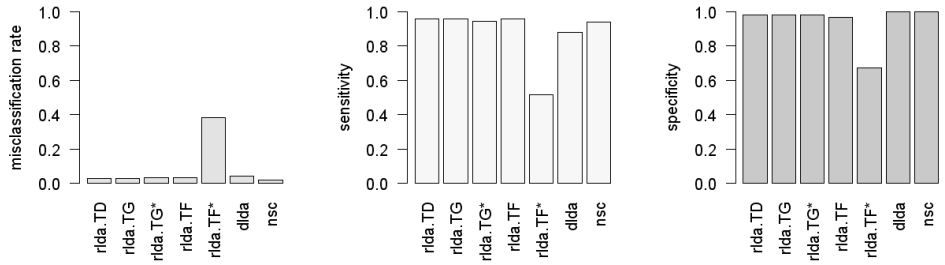

Figure 4.2: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 100 genes (except for nsc) of the two-class data Golub_Merge.

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	200	0.028 ●	0.960 ⋆	0.979
rlda.TG	200	0.028 ●	0.960 ⋆	0.979
rlda.TG*	200	0.091	0.876	0.927
rlda.TF	200	0.028 ●	0.960 ⋆	0.979
rlda.TF*	200	0.384	0.584	0.632
dlda	200	0.035	0.908	0.996 ●
nsc	7 129	0.021 ⋆	0.940 ●	1.000 ⋆

Table 4.3: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 200 genes (except for nsc) of the two-class data Golub_Merge (n=72).

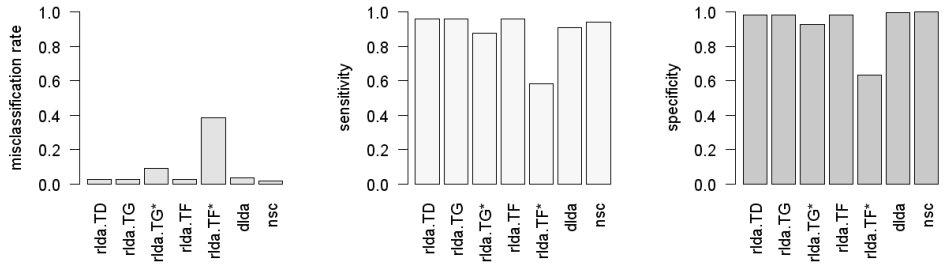

Figure 4.3: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 200 genes (except for nsc) of the two-class data Golub_Merge.

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	500	0.032	0.948 ●	0.979
rlda.TG	500	0.032	0.944	0.981
rlda.TG*	500	0.228	0.744	0.788
rlda.TF	500	0.030 ●	0.952 ⋆	0.979
rlda.TF*	500	0.417	0.584	0.583
dlda	500	0.031	0.916	0.998 ●
nsc	7 129	0.021 ⋆	0.940	1.000 ⋆

Table 4.4: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 500 genes (except for nsc) of the two-class data Golub_Merge (n=72).

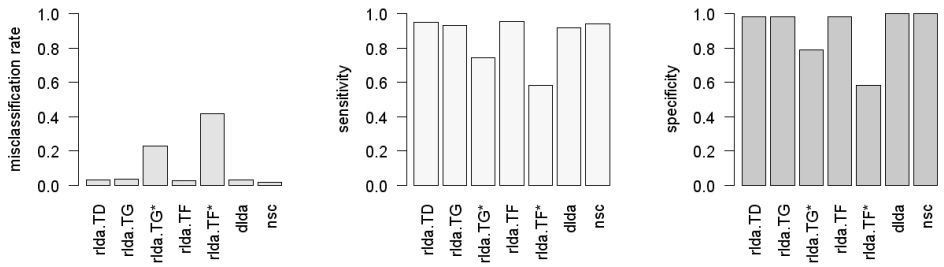

Figure 4.4: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 500 genes (except for nsc) of the two-class data Golub_Merge.

Results:

- In each data setting, i.e. for the top 50, 100, 200 and 500 selected genes, the methods rlda.TD, rlda.TG and rlda.TF produce similar results. The slight differences often are in the range of error fluctuation.
- The methods rlda.TD, rlda.TG and rlda.TF perform well with regard to all prediction measures. At least two of them outperform, even though marginally, the competitors NSC method and DLDA as well as the other variants rlda.TG* and rlda.TF* of LDA 'via the SH(IP)' in terms of the sensitivity in three of the four data settings.
- The competitor NSC method outperforms the other methods with regard to the mis-

classification rate and the specificity. The second competitor DLDA is basically as competitive as the NSC method in terms of the specificity, but performs only moderately otherwise.

- The methods rlda.TG* and rlda.TF* tend to produce the worst results. While rlda.TG* performs moderately for the small numbers of selected genes, rlda.TF* yields the highest misclassification rate and also the lowest sensitivity and specificity in all data settings.

Application to the data sCLLex data

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	50	0.244 ●	0.480	0.913
rlda.TG	50	0.244 ●	0.480	0.913
rlda.TG*	50	0.253	0.450	0.913
rlda.TF	50	0.247	0.460	0.920 ●
rlda.TF*	50	0.416	0.580 ⋆	0.593
dlda	50	0.204 ⋆	0.530 ●	0.953 ⋆
nsc	12 625	0.333	0.380	0.833

Table 4.5: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 50 genes (except for nsc) of the two-class data sCLLex (n=22).

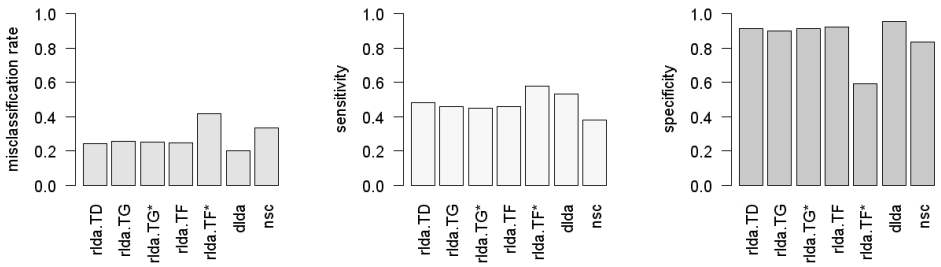

Figure 4.5: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 50 genes (except for nsc) of the two-class data sCLLex.

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	100	0.249	0.450	0.920 ●
rlda.TG	100	0.224 ⋆	0.520 ⋆	0.920 ●
rlda.TG*	100	0.264	0.450	0.897
rlda.TF	100	0.248	0.450	0.920 ●
rlda.TF*	100	0.468	0.490 ●	0.553
dlda	100	0.228 ●	0.480	0.933 ⋆
nsc	12 625	0.333	0.380	0.833

Table 4.6: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 100 genes (except for nsc) of the two-class data sCLLex (n=22).

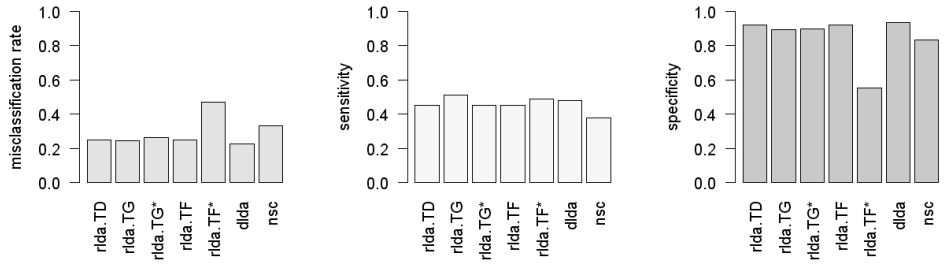

Figure 4.6: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 100 genes (except for nsc) of the two-class data sCLLex.

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	200	0.265	0.420	0.913
rlda.TG	200	0.267	0.430	0.903
rlda.TG*	200	0.284	0.510 ●	0.833
rlda.TF	200	0.249 ●	0.440	0.927 ●
rlda.TF*	200	0.533	0.560 ⋆	0.410
dlda	200	0.228 ⋆	0.480	0.933 ⋆
nsc	12 625	0.333	0.380	0.833

Table 4.7: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 200 genes (except for nsc) of the two-class data sCLLex (n=22).

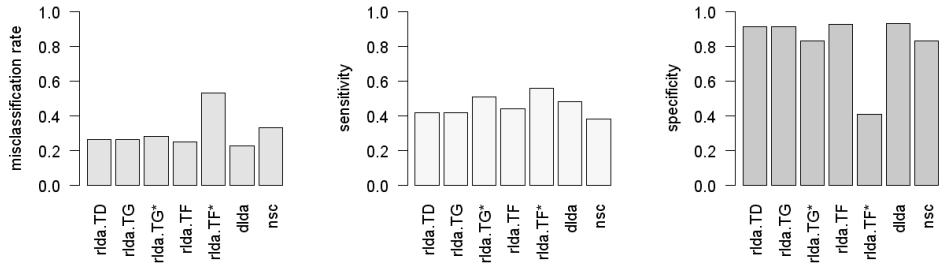

Figure 4.7: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 200 genes (except for nsc) of the two-class data sCLLex.

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	500	0.222 ●	0.470	0.953 ⋆
rlda.TG	500	0.218 ⋆	0.480	0.953 ⋆
rlda.TG*	500	0.279	0.550 ⋆	0.813
rlda.TF	500	0.218 ⋆	0.480	0.953 ⋆
rlda.TF*	500	0.444	0.510 ●	0.580
dlda	500	0.264	0.450	0.893 ●
nsc	12 625	0.333	0.380	0.833

Table 4.8: Overview of the 10 × five-fold CV error (the average misclassification rate over all 10×5=50 test sets), the sensitivity and the specificity obtained for each variant of LDA using the top 500 genes (except for nsc) of the two-class data sCLLex (n=22).

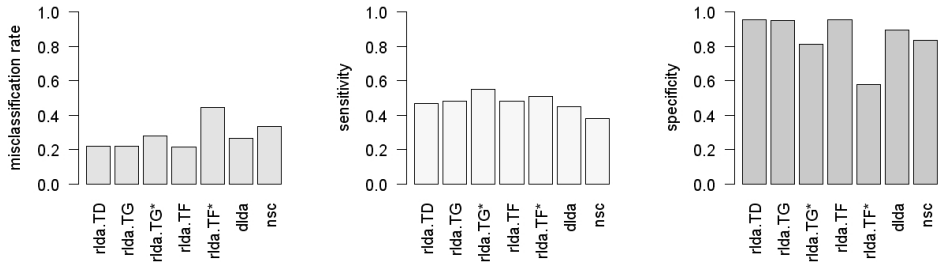

Figure 4.8: Graphical illustration of the misclassification rate, the sensitivity and the specificity for each variant of LDA using the top 500 genes (except for nsc) of the two-class data sCLLex.

Results:

- The classification results we obtain with this data set are relatively bad throughout all methods. This is likely to arise from the fact that the data set sCLLex only contains n=22 observations.
- In each data setting, i.e. for the top 50, 100, 200 and 500 selected genes, the methods rlda.TD, rlda.TG and rlda.TF produce similar results. The differences often are in the range of error fluctuation.
- The methods rlda.TD, rlda.TG and rlda.TF perform relatively well with regard to all

prediction measures in each data setting and even outperform the other methods in some situations. Especially for p=500, these three methods outperform the competitors NSC method and DLDA as well as the other variants rlda.TG* and rlda.TF* of LDA 'via the SH(IP)' in terms of the miclassification rate and the specificity.

- The competitor DLDA outperforms the other methods with regard to the misclassification rate and the specificity in two of the four data settings and works relatively well otherwise. The second competitor NSC method leads to the worst sensitivity and performs only slightly better otherwise.
- Although rlda.TF* produces the worst results in general, it outperforms in terms of the sensitivity in two of the four data settings. The method rlda.TG* performs relatively well with regard to the sensitivity, but is not competitive otherwise.

4.2.3 The c-nary case: $c > 2$

Consider the explanations in 4.2.1. In the following, we present - for the six-class data set ALL_a and for the four-class data set ALL_b - the results obtained for the different variants of LDA 'via the SH(IP)' as well as for the competitors DLDA and NSC method. For each variant of LDA the top 50, 100 and 200 genes are employed and the results are compared. Note that the NSC method is carried out once for the whole data set since it eliminates non-contributing genes itsself. For clarity's sake, we mark the best and the second best result by ⋆ and •, respectively.

Application to the ALL_a data

Method	p (# genes)	10 × three-fold CV error
rlda.TD	50	0.365
rlda.TG	50	0.362 •
rlda.TG*	50	0.494
rlda.TF	50	0.362 •
rlda.TF*	50	0.807
dlda	50	0.349 ⋆
nsc	12 625	0.384

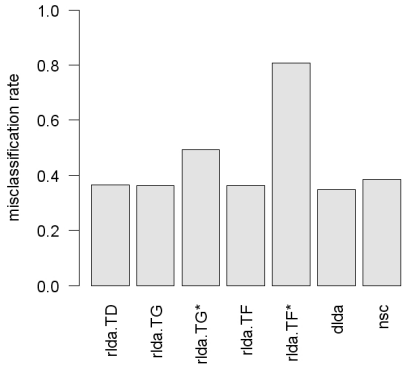

Figure 4.9: Overview and graphical illustration of the 10 × three-fold CV error (the average misclassification rate over all 10×3=30 test sets) obtained for each variant of LDA using the top 50 genes (except for nsc) of the six-class data ALL_a (n=128).

Method	p (# genes)	10 × three-fold CV error
rlda.TD	100	0.363
rlda.TG	100	0.361 ●
rlda.TG*	100	0.542
rlda.TF	100	0.362
rlda.TF*	100	0.806
dlda	100	0.351 ⋆
nsc	12 625	0.384

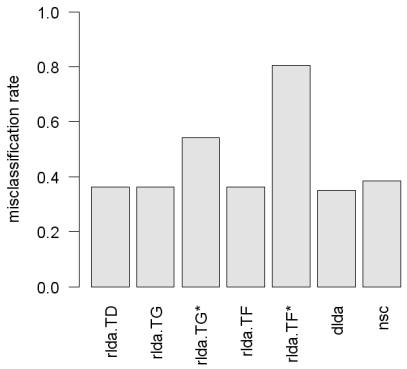

Figure 4.10: Overview and graphical illustration of the 10 × three-fold CV error (the average misclassification rate over all 10×3=30 test sets) obtained for each variant of LDA using the top 100 genes (except for nsc) of the six-class data ALL_a (n=128).

Method	p (# genes)	10 × three-fold CV error
rlda.TD	200	0.373
rlda.TG	200	0.372
rlda.TG*	200	0.583
rlda.TF	200	0.371 ●
rlda.TF*	200	0.840
dlda	200	0.357 ⋆
nsc	12 625	0.384

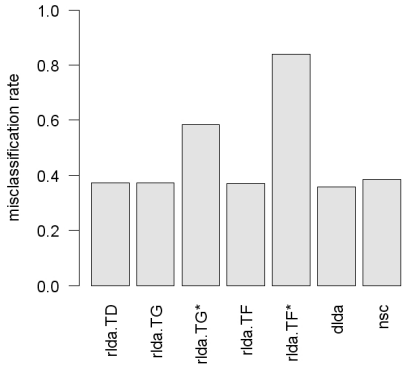

Figure 4.11: Overview and graphical illustration of the 10 × three-fold CV error (the average misclassification rate over all 10×3=30 test sets) obtained for each variant of LDA using the top 200 genes (except for nsc) of the six-class data ALL_a (n=128).

Results:

- The classification results we obtain with this data set are relatively bad throughout all methods. This, however, is the standard case for c-class data sets ($c > 2$).
- In each data setting, i.e. for the top 50, 100 and 200 selected genes, the methods rlda.TD, rlda.TG and rlda.TF produce similar results. The slight differences often are in the range of error fluctuation. The three methods rlda.TD, rlda.TG and rlda.TF outperform, even though slightly, the other methods except the DLDA in all data settings.
- The competitor DLDA marginally outperforms the other methods in each data setting. The second competitor NSC method classifies only moderately.
- The methods rlda.TG* and rlda.TF* produce the worst results, rlda.TF* yields even more misclassifications than correct classifications in all data settings.

Application to the ALL_b data

Method	p (# genes)	10 × three-fold CV error
rlda.TD	50	0.250 ●
rlda.TG	50	0.255
rlda.TG*	50	0.313
rlda.TF	50	0.250 ●
rlda.TF*	50	0.639
dlda	50	0.236 ★
nsc	12 625	0.250 ●

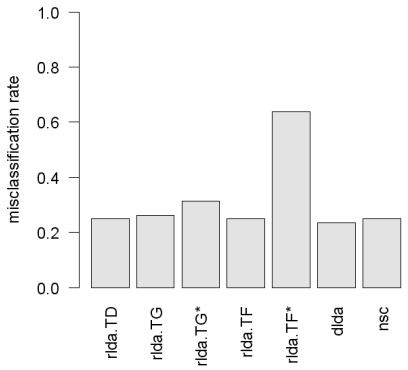

Figure 4.12: Overview and graphical illustration of the 10 × three-fold CV error (the average misclassification rate over all 10×3=30 test sets) obtained for each variant of LDA using the top 50 genes (except for nsc) of the four-class data ALL_b (n=128).

Method	p (# genes)	10 × three-fold CV error
rlda.TD	100	0.266
rlda.TG	100	0.269
rlda.TG*	100	0.344
rlda.TF	100	0.261
rlda.TF*	100	0.709
dlda	100	0.231 ⋆
nsc	12 625	0.250 ●

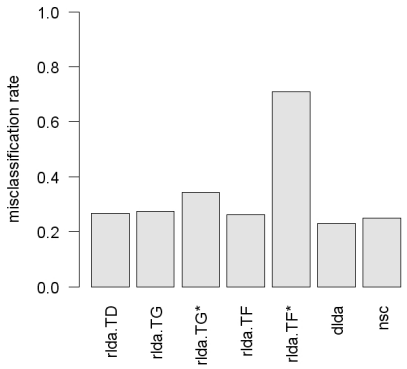

Figure 4.13: Overview and graphical illustration of the 10 × three-fold CV error (the average misclassification rate over all 10×3=30 test sets) obtained for each variant of LDA using the top 100 genes (except for nsc) of the four-class data ALL_b (n=128).

Method	p (# genes)	10 × three-fold CV error
rlda.TD	200	0.280
rlda.TG	200	0.281
rlda.TG*	200	0.446
rlda.TF	200	0.277
rlda.TF*	200	0.768
dlda	200	0.238 ⋆
nsc	12 625	0.250 ●

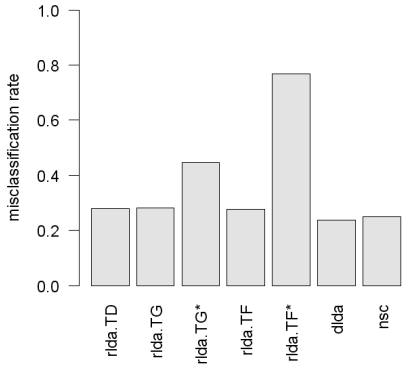

Figure 4.14: Overview and graphical illustration of the 10 × three-fold CV error (the average misclassification rate over all 10×3=30 test sets) obtained for each variant of LDA using the top 200 genes (except for nsc) of the four-class data ALL_b (n=128).

Results:

- The classification results we obtain with this data set are relatively bad throughout all methods. This, however, is the standard case for c-class data sets ($c > 2$).
- In each data setting, i.e. for the top 50, 100 and 200 selected genes, the methods rlda.TD, rlda.TG and rlda.TF produce similar results. The slight differences often are in the range of error fluctuation. The three methods rlda.TD, rlda.TG and rlda.TF perform well, but do not outperform the other methods.
- The competitor DLDA outperforms, even though marginally, the other methods in each data setting. The second competitor NSC method performs slightly better than the variants of LDA 'via the SH(IP)'.
- The methods rlda.TG* and rlda.TF* produce the worst results, rlda.TF* yields even more misclassifications than correct classifications in all data settings.

4.3 Discussion

Having reported the results obtained for the different variants of LDA 'via the SH(IP)' and its competitors DLDA and NSC method using the real-life gene expression data sets described in 1.3, we now constrain our attention on the extensive discussion of the results from above.
Generally, it appears that a decrease of the sample size n and an increase of the number c of classes leads to worse results. For instance, the effect of the sample size can be illustrated by comparing the misclassification rate, the sensitivity and the specificity for the two-class data sets Golub_Merge and sCLLex. Intuitively, these effects are clear and have been observed frequently in previous studies. Further, we find that the methods rlda.TD, rlda.TG and rlda.TF produce similar results in each data setting for all data sets. This finding also applies for the standard deviations which can be inspected in Appendix B. The slight differences often are in the range of error fluctuation. This unexpected result gives some indication of the additional value of incorporating external biological knowledge into the covariance target and of accounting for correlations between genes in general. Both rlda.TG and rlda.TF assume correlations between genes while rlda.TG additionally incorporates external biological knowledge, see 3.2. Neither rlda.TG nor rlda.TF performs considerably better than rlda.TD which employs a diagonal covariance target. In the following,

we point out **possible** reasons explaining this result. Note that rlda.TG equals rlda.TD if the selected genes do not belong to any gene functional group in the database KEGG. Theoretically, this can occur since more than 50 % of the genes from the data sets we used are in no gene functional group [1]. Hence, the reason for rlda.TG and rlda.TD producing similar results might be that they have basically the same form in the data settings we have chosen. On the other hand, rlda.TD equals rlda.TF if the average of sample correlations , i.e. $\bar{r}$, takes the value zero. In 3.2, we have found that $\bar{r}$ in fact is very close to zero.
The methods rlda.TD, rlda.TG and rlda.TF perform well in binary and c-nary classification problems in each data setting. In some situations, they even outperform the competitors DLDA and NSC method. Although the margin of improvement often is slight, it is nevertheless remarkable that the three methods are still competitive in the c-nary case. On the contrary, rlda.TG* and rlda.TF* tend to produce the worst results. These methods seem to suffer severely from the fact that the associated covariance targets are likely to be indefinite. Consequently, the computation of the Moore-Penrose pseudoinverse is unstable which leads to bad results. Concerning the competitors DLDA and NSC method, we find that they work well and often outperform the other methods, but are also outperformed themselves by the methods rlda.TD, rlda.TG and rlda.TF in some situations. Especially when the number of classes increases, the DLDA shows advantage over the other methods. It outperforms, albeit marginally, the other competitor NSC method and the variants of LDA 'via the SH(IP)' in each data setting. The NSC method appears to weaken the more classes a data set consists of. This confirms the findings by Guo et al. [22].

In conclusion, let us sum up the crucial statements: **1.** According to our results, there is no additional value of incorporating external biological knowledge in the way we did in this thesis and of accounting for correlations between genes in general. The method rlda.TD turns out to suffice. Note that we have only considered data settings with $p \leq 500$. **2.** The method rlda.TD can be as competitive as the NSC method in binary classification problems and appears to perform better in c-nary classification problems. Both methods require the determination of a shrinkage parameter. While

[1] In the data set Golub_Merge 4 172 out of 7 129 genes ($\hat{=}$ 58.5%) are in no gene functional group. In the data sets ALL and sCLLex 8 040 out of 12 625 genes ($\hat{=}$ 63.7%) are in no gene functional group.

the shrinkage parameter in the NSC method is determined using a cross-validation procedure which is computationally very expensive, the shrinkage parameter in the rlda.TD is determined analytically. **3.** For the purpose of prognosis, more studies are still needed. A simulation study might be beneficial for the comparison of the NSC method and the rlda.TD.

Chapter 5

Summary and Outlook

In this thesis, we have studied a variant of regularized linear discriminant analysis incorporating biological knowledge on gene functional groups from the database KEGG. This chapter's objective is to give a summary of the methodological path we followed as well as the results we achieved. The crucial statements are stressed and an outlook to our future work in this field - being within the scope of current scientific focus - is given.

In the introductory Chapter 1, we lead to the topic and gave a brief guideline through this thesis. We further provided an overview of the five microarray gene expression data sets we used throughout this thesis. For the purpose of knowledge extraction, such data sets have to fulfill the requirement of being compatible with the database KEGG.

In Chapter 2, we presented the scientific scope on which this thesis is built. In particular, we started with explaining the idea behind discriminant analysis which nowadays can be seen as a generic term for a multiplicity of methods. In this thesis, we focused on the linear discriminant analysis (LDA) resulting from the assumption of equal within-class covariance matrices. We further discussed its generalization to the high-dimensional setting where the number p of variables by far exceeds the number n of observations. Here, the major challenge is to modify the traditional pooled empirical covariance estimator such that the resulting estimator has the required properties of being well-conditioned and invertible. In 2.2.2, we briefly

mentioned some approaches coping with high-dimensionality and pointed out the shrinkage principle they are based on. Moreover, we addressed the issue of measuring the prediction accuracy. According to several studies using a repeated K-fold cross-validation procedure turns out to be beneficial in the $n \ll p$ case [9]. Basic insights into the database KEGG where prior biological information is encoded by graphs were given in 2.3. For simplicity's sake, we assumed that a KEGG pathway forms a gene functional group. We completed Chapter 2 with a review of the approaches by Guillemot et al. [17] and Tai and Pan [46], both differentially incorporating prior knowledge into the regularized linear discriminant analysis. Guillemot et al. assume the availability of one single graph including all genes from a given data set which, however, is not available in practice. Tai and Pan group the genes from a given data set according to their biological functions into a block-diagonal structure. Hence, the genes occuring in multiple gene functional groups are omitted or duplicated in order to ensure the between-group independence. In our opinion, however, neither omitting nor duplicating should be the strategy of choice.

In Chapter 3, we first detached our explanations from the special case of linear discriminant analysis and introduced a new covariance estimation procedure we referred to as **SHIP**: **SH**rinking and **I**ncorporating **P**rior knowledge. The resulting covariance estimator $\hat{\mathbf{\Sigma}}_{\mathbf{SHIP}}$ is based on the shrinkage estimator introduced by Ledoit and Wolf [32] and picked up by Schäfer and Strimmer [41], being enhanced by consideration of prior knowledge on gene functional groups. In order to incorporate this knowledge into the shrinkage estimator, we proposed two modified versions of target $\mathbf{F}$ from Schäfer and Strimmer where genes that occur in the same gene functional group have constant correlation. As a consequence, we do not need to omit or duplicate any genes like in Tai and Pan. While the first version employs one constant correlation (i.e. the average of sample correlations between connected genes), the second one employs a negative and a positive constant correlation. Although the latter appears to be more adequate in the context of biological interpretation, it turns out to be strongly indefinite and thus does not fulfill the positive definiteness requirement. In 3.3, we studied in detail the analytic determination of the optimal shrinkage intensity λ, whereas optimality is considered from a decision theoretic perspective with a quadratic loss function. In fact, the analytic determination of λ constitutes a clear advantage over common approaches like cross-validation de-

manding computationally very expensive procedures. Subsequently, we dealt with the consistent estimation of λ. For this purpose, we followed the explanations by Ledoit and Wolf and Schäfer and Strimmer and additionally gave a detailed proof of the formula. We have implemented the shrinkage estimator 'via the SH(IP)', i.e. the shrinkage estimator where one can choose between different covariance targets, in the language `R`.

Chapter 4 addressed a variant of regularized linear discriminant analysis which generalizes the idea of the shrinkage estimator introduced in Chapter 3. We demonstrated in detail how the ideas from Chapter 3 can technically be included into the framework of linear discriminant analysis. In particular, we discussed two possible approaches to the pooled version of the shrinkage estimator from above. While the first one consists in pooling the within-class shrinkage estimators, the second one aims at establishing the theoretical framework for the pooled shrinkage estimator with one shrinkage intensity in the sense of Ledoit and Wolf. In this thesis, we employed the first approach. Using the real-life data presented in Chapter 1, we further examined and compared the classification performance of five variants of the LDA 'via the SH(IP)', from which two incorporate biological knowledge on gene functional groups and three do not. They have been implemented in the language `R`. We chose the diagonal linear discriminant analysis (DLDA) and the nearest shrunken centroids method (NSC) [15] as competitors. According to our results, there seems to be no additional value of incorporating external biological knowledge in the way we did in this thesis and of accounting for correlations between genes in general. We found that the variant of LDA 'via the SH(IP)' employing the diagonal covariance target $\mathbf{D}$ from Schäfer and Strimmer suffices. Moreover, it was shown that it can be as competitive as the NSC method in binary classification problems and appears to perform better in c-nary classification problems. Although more studies are still needed in order to compare both methods, the fact that the shrinkage intensity in the method we proposed is determined analytically and thus leads to a minimum mean squared error of the resulting estimator appears to be a clear advantage.

In the following, let us consider some subjects to our future research within this field. Some of these subjects are directly based on the ideas pursued in this thesis.

Outlook 1

In 2.3, it was shown that KEGG pathways are represented as graphs in which the edges stand for the chemical reactions or relations and the vertices stand for the genes taking part in these reactions or relations [28]. According to Tai and Pan [46], we assumed that a KEGG pathway forms a gene functional group. In the latter all genes are related to each other via their function. Thus, we concluded that for each pair of genes in a KEGG graph there exists at least one path whose length can be defined as the number of edges lying between these two genes. Our idea was the additional incorporation of the length of the shortest path between two genes from one gene functional group into target **G** proposed in 3.2.2. The underlying hypothesis was that genes being close to each other in the pathway are more likely to co-express. We defined the resulting covariance target as follows.

Target H:

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \frac{\bar{r}\sqrt{s_{ii}s_{jj}}}{l(g_i,g_j)} & \text{if } i \neq j, i \sim j \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{r}$ is the average of sample correlations between connected genes and $l(g_i, g_j)$ denotes the length of the shortest path between the genes g_i and g_j, $i \neq j$, $i, j = 1, ..., p$. We found, however, that this approach fails for the following reason. The genes included in a KEGG graph, i.e. the graph's vertices, are not necessarily associated via any path. More precisely, the number of edges being connected to such a gene might be zero. Figur 5.1 (see below) shows the graphical representation of the KEGG pathways hsa04510, hsa04664, hsa04010 and hsa04640 and illustrates well the difficulties possibly arising in the computation of target **H** as proposed above. For instance, the pathway hsa04640 represents the extreme case since is consists of 87 vertices and zero edges.

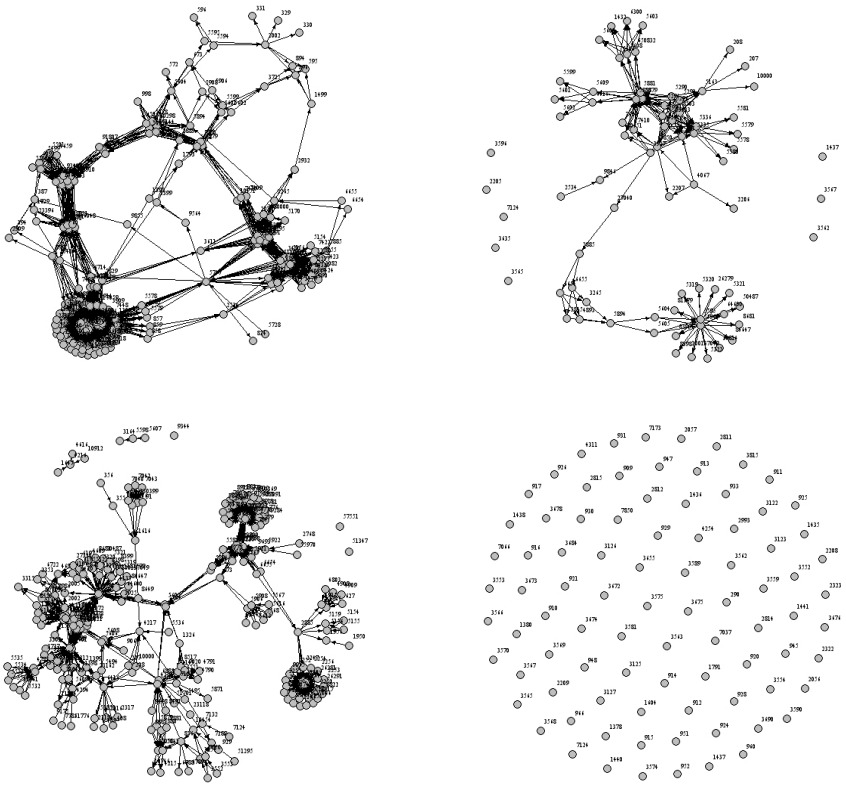

Figure 5.1: Graphical representation of the KEGG pathways hsa04510, hsa04664, hsa04010 and hsa04640 (from top left to bottom right).

An alternative to target **H** imposing an additional restriction might be as follows.

Target I:

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j \\ \frac{\bar{r}\sqrt{s_{ii}s_{jj}}}{l(g_i,g_j)} & \text{if } i \neq j, i \sim j, l(g_i, g_j) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{r}$ is the average of sample correlations between connected genes and $l(g_i, g_j)$

denotes the length of the shortest path between the genes g_i and g_j, $i \neq j$. Moreover, a rigorous approach we though try to avoid would be to consider only the genes for which the length as defined above is positive. Nevertheless, we have not fully understood yet the detailed biological background of gene functional groups, pathways and their graphical representation. We are currently trying to gain more understanding of this topic in order to improve the handling of biological knowledge within the regularized linear discriminant analysis. In fact, future research in biostatistics will demand more biological and medical expertise on the part of statisticians.

Outlook 2

In this thesis, we have tried to incorporate biological knowledge on gene functional groups by shrinking the empirical covariance matrix towards a non-diagonal covariance structure. For instance, we have considered target $\mathbf{G}$ in 3.2.2 where genes occuring in at least one same gene functional group have constant correlation. We found in 4.2 that non-diagonal covariance targets do not lead to an improvement of the classification accuracy. It was shown that the method rlda.TD which employs a diagonal covariance target performs well and can be as competitive as the NSC method. Hence, the question of developing a diagonal covariance target incorporating prior knowledge on gene functional groups arises. In a next step, we will study a modified version of the diagonal target $\mathbf{B}$ outlined in 3.2.1 which is characterized by common variances. One idea might be to group the genes according to their functions and to compute the mean variance for each gene functional group separately such that the genes occuring in the same group have common variance. For G disjoint gene functional groups, this procedure would yield G variances. Since, however, the gene functional groups do overlap in practice, one might apply the following strategy. Let gene g occur in three gene functional groups. Then we propose its variance to be computed as the average of the three mean variances obtained for the gene functional groups gene g occurs in. In our opinion, it might be worthwhile to work on such a covariance target in order to subsequently include it into the framework of LDA 'via the SH(IP)'.

Outlook 3

In 4.1, we proposed two possible approaches to the pooled version of the shrinkage estimator from Chapter 3. In the second approach we tried to establish the theoretical framework for the pooled shrinkage estimator with one shrinkage intensity in the sense of Ledoit and Wolf. Obviously the situation becomes more difficult than in the first approach which consists in pooling the within-class shrinkage estimators. Nevertheless, it is worthwhile to establish the full theoretical framework and to implement the resulting pooled shrinkage estimator in order to examine whether the second approach yields a better classification performance. We are currently working on this topic.

Outlook 4

Moreover, it might be beneficial to incorporate prior biological knowledge on gene functional groups into the variable selection process. One possibility could be to select solely the genes occuring in at least one gene functional group. In case the number of genes is still too large, one could subsequently perform variable selection using the reduced set of genes. In this thesis, we have performed variable selection without considering the knowledge extracted from KEGG. The motivation behind this approach was being cautious in using this prior knowledge.

Outlook 5

In this thesis, we extracted the biological knowledge on gene functional groups using the database KEGG. Additionally, it might be interesting to employ one of the various other existing databases in order to compare the results. Besides KEGG [28] the well-known databases are Biocarta [6], BioCyc [29], Gene Ontology [3], GenMAPP [21], Reactome [27] and TransPath [30].

Chapter 6

Conclusion

The result of our endeavors to improve the prediction accuracy by incorporating prior biological knowledge on gene functional groups is - from one of the various points of view of this thesis - disappointing. Even though we have suggested two different possibilities to embed external knowledge on gene functional groups into the regularized linear discriminant analysis, there is no evidence for neither of them to be the clear winner in comparison with well-known methods such as the diagonal linear discriminant analysis (DLDA) and the nearest shrunken centroids method (NSC) which both are applicable in the high-dimensional setting. The fact that our approach may improve the results' interpretability is only a cold comfort given that the price for this slightly better interpretability is a more technical procedure due to the application of the database KEGG. On the other hand, initially unexpected results opened up the gate to interesting directions of our future research in this field. From the current point of view, the next step should be developing a diagonal covariance target incorporating prior knowledge on gene functional groups in order to subsequently include it into the framework of LDA 'via the SH(IP)'.

In conclusion, we believe that the field of class prediction methods incorporating prior biological knowledge from databases is on the rise. Due to the tremendous progress on the biological side it is though a challenge to handle this knowledge appropriately. Indeed, future research in biostatistics - and especially in the field of statistical genetics - will demand a large degree of biological and medical expertise on the part of statisticians.

Appendix A

Computational aspects

The statistical analyses presented in this thesis were carried out by means of the statistical software and language `R 2.9.1`. Most of these analyses also can be carried out by using former versions of `R`. The extraction of biological knowledge from KEGG, however, requires the loading of packages which were built under recent versions. In order to ensure a proper operability of these packages, we thus recommend to use at least `R 2.9.1`. We have both employed existing methods and newly implemented the methods proposed in this thesis, e.g. the variants of LDA 'via the SH(IP)' examined in Chapter 4.
For clarity's sake, we will solely give brief descriptions of the most important procedures we have implemented. The complete programming codes, inclusively detailed examples for the extraction of knowledge from KEGG as well as the methods employed in Chapter 3, can be found in the folder 'R.code' on the attached CD. Subsequently, we present an example demonstrating how to perform the different variants of LDA proposed in this thesis in `R`.

A.1 Description of the software

In a nutshell, we present an outline of the most important methods we have implemented. Note that - according to the explanations in Section 2.3 - we assume that a KEGG pathway is a gene functional group and vice versa.

- `check.path()`

  The auxiliary function `check.path()` checks whether two pathway lists `p1` and `p2` share at least one name. Here, a pathway list represents a list of pathways/gene functional groups in which a certain gene is included. It can be obtained from KEGG. Correspondingly, `check.path()` checks whether two genes have *at least* one pathway in common.
  Input:
  1. A pathway list `p1`.
  2. A pathway list `p2`.
  Output:
  `0` or `1`. The value `1` means that the two genes have at least one pathway in common. The value `0` means that the two genes have no pathway in common. In case both genes are in no pathway, i.e. `p1 = NA` and `p2 = NA`, this is not considered as common pathway.

- `target.help()`

  The auxiliary function `target.help()` uses `check.path()` and creates a matrix indicating whether there is a connection between two genes (i.e. whether the two genes have at least one pathway in common).
  Input:
  A gene list `genesINpaths` which can be obtained from KEGG. For details see the file 'pathway.extraction.KEGG.r' in the folder 'KEGG.examples' of the folder 'R.code' on the attached CD. Each entry of `genesINpaths` is itself a list of pathway names specifying the pathways in which a gene is included. If a gene is not included in any pathway, the entry is `NA`.
  Output:
  A matrix with the entries `0` and `1`. `0` means that the two genes have no pathway in common. `1` means that the two genes have at least one pathway in common.

- `targetG()`

  The function `targetG()` uses `target.help()` and creates the covariance target $\mathbf{G}$ as introduced in Chapter 3.

Input:

1. A gene list `genesINpaths`. Each entry of `genesINpaths` is itself a list of pathway names specifying the pathways in which a gene is included. If a gene is not included in any pathway, the entry is `NA`.

2. The data matrix `x`.

Output:

1. The covariance target $\mathbf{G}$.

2. The mean correlation `cora` over the genes that have at least one pathway in common.

- `targetGstar()`

  The function `targetGstar()` uses `target.help()` and creates the covariance target $\mathbf{G}^*$ as introduced in Chapter 3, i.e. we allow two mean correlations (a positive and a negative one) in order to pay attention to the fact that genes can be negatively correlated within the same pathway.

  Input:

  1. A gene list `genesINpaths`. Each entry of `genesINpaths` is itself a list of pathway names specifying the pathways in which a gene is included. If a gene is not included in any pathway, the entry is `NA`.

  2. The data matrix `x`.

  Output:

  1. The covariance target $\mathbf{G}^*$.

  2. The mean correlation `cora.pos` over the genes that have at least one pathway in common and that are positively correlated.

  3. The mean correlation `cora.neg` over the genes that have at least one pathway in common and that are negatively correlated.

- `choose.target()`

  The function `choose.target()` is able to create three types of target matrices, i.e. target $\mathbf{D}$, target $\mathbf{F}$ and target $\mathbf{F}^*$ from Chapter 3. The choice of the concrete covariance target is controlled by the argument `type`.

Input:

1. The data matrix `x`.

2. The denotation `type` of the concrete target matrix to be computed.

Output:

1. The covariance target chosen by the argument `type`.

2. The mean correlation `cora` over all correlations. For `type =` "`targetF*`" `cora` is a vector consisting of `cora.pos` and `cora.neg`.

- `shrink.estim()`

  The function `shrink.estim()` is able to compute the different variants of the shrinkage estimator 'via the SH(IP)' from Chapter 3. Since only the correlations are shrunken, the standardized data matrix is employed as proposed by Schäfer and Strimmer [41], see 3.3.3. The method `shrink.estim()` is created for the target matrices $\mathbf{D}$, $\mathbf{F}$, $\mathbf{F}^*$, $\mathbf{G}$ and $\mathbf{G}^*$ which can be obtained by using the functions `targetG()`, `targetGstar()` or `choose.target()`.

  Input:

  1. The data matrix `x`.

  2. An object `tar` created by `targetG()`, `targetGstar()` or `choose.target()`.

  Output:

  1. One variant of the shrinkage estimator 'via the SH(IP)', depending on the argument `tar`.

  2. The shrinkage intensity lambda.

- `rlda.iter()`

  The function `rlda.iter()` carries out the LDA based on the Bayes classification rule with normally distributed predictors for *one* iteration. The possible covariance estimators are pooled variants of the shrinkage estimator 'via the SH(IP)' obtained by using `shrink.estim()` (see Section 4.1).

  Input:

  1. The learning set `Xlearn` (see Section 2.1.3).

  2. The test set `Xtest` (see Section 2.1.3).

  3. The vector `Ylearn` of class observations belonging to the learning set.

4. The argument `type` indicating which target matrix is used for computing the shrinkage estimator. One can choose between `type = "targetD"`, `type = "targetF"`, `type = "targetF*"`, `type = "targetG"`, `type = "targetG*"` and `type = "standard"` (especially for $n > p$, computes the standard pooled covariance matrix).

5. A gene list `genesINpaths` (see above). It is relevant for `type = "targetG"` and `type = "targetG*"`, i.e. for the targets incorporating biological knowledge from KEGG. For the other types where a gene list is not necessary, one can set `genesINpaths=NA`.

Output:

The predicted classes for `Xtest`.

- `rldaCMA()`

  The method `rldaCMA()` has been incorporated into the framework of the CMA package [43]. It employs `rlda.iter()` in each iteration and can be called as classifier in the CMA method `classification()` which subsequently carries out the class prediction using the learning and test sets as generated by the CMA method `GenerateLearningsets()`.

  Input:

  1. The complete data matrix `X`.
  2. The vector `y` of class observations belonging to `X`.
  3. The indices `learnind` specifying the learning and test sets which in turn are generated by the CMA method `GenerateLearningsets()`.
  4. The argument `type` indicating which target matrix is used for computing the shrinkage estimator (see `rlda.iter()`).
  5. A gene list `genesINpaths` (see `rlda.iter()`).

  Output:

  The predicted classes for all iterations, i.e. for all test sets.

A.2 Using the software

In the following, we present an example in order to demonstrate how to use the software from A.1. We use the two-class data set Golub_Merge. We presume that the necessary programming codes as supplemented on the attached CD are properly loaded before the code of the example is used. Loading all required packages and the programming codes can be done by using the file 'initialization.r' in the folder 'R.code'.

1. Initialization

```
source("initialization.r")
```

2. Data preparation

```
# a) We load and prepare the data set Golub_Merge.

library(golubEsets)
data(Golub_Merge)
show(Golub_Merge)
phenodata  <- pData(Golub_Merge)
Y          <- phenodata$ALL.AML
X          <- exprs(Golub_Merge)
X          <- t(X)


# b) We load the annotation package for the data set Golub_Merge.

library(hu6800.db)


# c) We extract the "biological knowledge" for the data set Golub_Merge.

genelist <- as.list(hu6800PATH)


# d) We generate the learning and test sets employing the CMA package and
#    use a stratified five-fold cross-validation as scheme.
```

```
set.seed(1234)
learnset <- GenerateLearningsets(y=Y,method="CV",fold=5,niter=10,strat=TRUE)


# e) We perform a gene selection in each learning set using the CMA package.

geneselect <- GeneSelection(X=X,y=Y,learningsets=learnset,method="t.test")
```

3. Linear discriminant analysis 'via the SH(IP)'

```
# a) We carry out the classification using rldaCMA, the method we developed
#    in the Chapters 3 and 4. First, we choose type="TargetD". Second, we
#    choose type="TargetG" for illustration purposes. The argument nbgene=50
#    indicates that a variable selection is performed in each iteration and
#    the best 50 genes are employed.

classifyTD <- classification(X=X,y=Y, learningsets=learnset, type="TargetD",
              genesINpaths=NA, genesel=geneselect, nbgene=50,
              classifier=rldaCMA)

classifyTG <- classification(X=X,y=Y, learningsets=learnset, type="TargetG",
              genesINpaths=genelist, genesel=geneselect, nbgene=50,
              classifier=rldaCMA)


# b) We examine the classification performance using the CMA method
#    evaluation(). We choose the prediction accuracy measures average
#    misclassification rate over all iterations, average sensitivity over
#    all iterations and average specificity over all iterations.

evalTD.m  <- evaluation(classifyTD, measure="misclassification")
evalTD.s  <- evaluation(classifyTD, measure="sensitivity")
evalTD.sp <- evaluation(classifyTD, measure="specificity")

evalTG.m  <- evaluation(classifyTG, measure="misclassification")
evalTG.s  <- evaluation(classifyTG, measure="sensitivity")
evalTG.sp <- evaluation(classifyTG, measure="specificity")
```

4. Some outputs

```
# a) show(Golub_Merge) leads to the annotation package to be loaded from
#    http://www.bioconductor.org/.

show(Golub_Merge)
ExpressionSet (storageMode: lockedEnvironment)
assayData: 7129 features, 72 samples
  element names: exprs
phenoData
  sampleNames: 39, 40, ..., 33  (72 total)
  varLabels and varMetadata description:
    Samples: Sample index
    ALL.AML: Factor, indicating ALL or AML
    ...: ...
    Source: Source of sample
    (11 total)
featureData
 featureNames: AFFX-BioB-5_at,AFFX-BioB-M_at,...,Z78285_f_at(7129 total)
 fvarLabels and fvarMetadata description: none
experimentData: use 'experimentData(object)'
 pubMedIds: 10521349
Annotation: hu6800


# b) The genelist has the following form (we consider only the first,
#    the second and the sixth element (gene) for illustration). It
#    corresponds to the argument genesINpaths.

genelist[c(1:2,6)]
$A28102_at
[1] "04080"

$AB000114_at
[1] NA

$AB000409_at
[1] "04010" "04910"
```

```
# c) The evaluation of classifyTD and classifyTG yields the following
#    results (we consider only the misclassification rate):

evalTD.m
evaluated method: 'rldaCMA'
scheme used :'iterationwise'
performance measure: 'misclassification'
mean performance is 0.043
with a standard error of 0.007

evalTG.m
evaluated method: 'rldaCMA'
scheme used :'iterationwise'
performance measure: 'misclassification'
mean performance is 0.045
with a standard error of 0.008
```

Appendix B

Additional remarks

In Section 4.2, we found that the methods rlda.TD, rlda.TG and rlda.TF produce similar results in each data setting for all data sets we employed, i.e. for the two-class data sets Golub_Merge and sCLLex as well as for the six-class data set ALL_a and for the four-class data set ALL_b. Thus, we obtained similar results with regard to the prediction measures misclassification rate, sensitivity and specificity, whereas each given prediction measure is the average prediction measure over all test sets. For the sake of completeness and accuracy, the standard deviation should be examined. In the following, we present - for the methods rlda.TD, rlda.TG and rlda.TF - the results from 4.2 and the corresponding standard deviations for each data set. Apparently the similarity of the results also applies for the standard deviations which confirms the findings from 4.2.

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	50	0.043 (± 0.007)	0.916 (± 0.018)	0.979 (± 0.006)
rlda.TG	50	0.045 (± 0.008)	0.912 (± 0.019)	0.979 (± 0.006)
rlda.TF	50	0.043 (± 0.007)	0.932 (± 0.016)	0.971 (± 0.007)
rlda.TD	100	0.028 (± 0.006)	0.960 (± 0.011)	0.979 (± 0.006)
rlda.TG	100	0.029 (± 0.006)	0.956 (± 0.013)	0.979 (± 0.006)
rlda.TF	100	0.034 (± 0.006)	0.960 (± 0.011)	0.969 (± 0.007)
rlda.TD	200	0.028 (± 0.006)	0.960 (± 0.011)	0.979 (± 0.006)
rlda.TG	200	0.028 (± 0.006)	0.960 (± 0.011)	0.979 (± 0.006)
rlda.TF	200	0.028 (± 0.006)	0.960 (± 0.011)	0.979 (± 0.006)
rlda.TD	500	0.032 (± 0.006)	0.948 (± 0.014)	0.979 (± 0.006)
rlda.TG	500	0.032 (± 0.006)	0.944 (± 0.014)	0.981 (± 0.006)
rlda.TF	500	0.030 (± 0.006)	0.952 (± 0.013)	0.979 (± 0.006)

Method	p (# genes)	10 × five-fold CV error	Sensitivity	Specificity
rlda.TD	50	0.244 (± 0.026)	0.480 (± 0.057)	0.913 (± 0.028)
rlda.TG	50	0.244 (± 0.026)	0.480 (± 0.057)	0.913 (± 0.028)
rlda.TF	50	0.247 (± 0.024)	0.460 (± 0.057)	0.920 (± 0.026)
rlda.TD	100	0.249 (± 0.026)	0.450 (± 0.056)	0.920 (± 0.026)
rlda.TG	100	0.224 (± 0.027)	0.520 (± 0.057)	0.920 (± 0.026)
rlda.TF	100	0.248 (± 0.025)	0.450 (± 0.056)	0.920 (± 0.026)
rlda.TD	200	0.265 (± 0.025)	0.420 (± 0.056)	0.913 (± 0.027)
rlda.TG	200	0.267 (± 0.025)	0.430 (± 0.057)	0.903 (± 0.026)
rlda.TF	200	0.249 (± 0.023)	0.440 (± 0.055)	0.927 (± 0.024)
rlda.TD	500	0.222 (± 0.025)	0.470 (± 0.058)	0.953 (± 0.021)
rlda.TG	500	0.218 (± 0.025)	0.480 (± 0.061)	0.953 (± 0.021)
rlda.TF	500	0.218 (± 0.025)	0.480 (± 0.057)	0.953 (± 0.021)

Table B.1: Overview of the 10 × five-fold CV error, the sensitivity and the specificity obtained for the methods rlda.TD, rlda.TG and rlda.TF using the top 50, 100, 200 and 500 genes of the two-class data Golub_Merge (n=72) (top) and sCLLex (n=22) (bottom). In brackets the standard deviation is given.

Method	p (# genes)	10 × three-fold CV error
rlda.TD	50	0.365 (± 0.010)
rlda.TG	50	0.362 (± 0.010)
rlda.TF	50	0.362 (± 0.011)
rlda.TD	100	0.363 (± 0.009)
rlda.TG	100	0.361 (± 0.010)
rlda.TF	100	0.362 (± 0.010)
rlda.TD	200	0.373 (± 0.009)
rlda.TG	200	0.372 (± 0.010)
rlda.TF	200	0.371 (± 0.010)

Method	p (# genes)	10 × three-fold CV error
rlda.TD	50	0.250 (± 0.009)
rlda.TG	50	0.255 (± 0.009)
rlda.TF	50	0.250 (± 0.009)
rlda.TD	100	0.266 (± 0.010)
rlda.TG	100	0.269 (± 0.009)
rlda.TF	100	0.261 (± 0.010)
rlda.TD	200	0.280 (± 0.011)
rlda.TG	200	0.281 (± 0.010)
rlda.TF	200	0.277 (± 0.011)

Table B.2: Overview of the 10 × three-fold CV error obtained for the methods rlda.TD, rlda.TG and rlda.TF using the top 50, 100 and 200 genes of the six-class data ALL_a (n=128) (top) and the four-class data ALL_b (n=128) (bottom). In brackets the standard deviation is given.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, New York, 1970.

[2] C. Ambroise and G. J. McLachlan. Selection bias in gene extraction in tumour classification on basis of microarray gene expression data. *Proceedings of the National Academy of Science*, 99:6562–6566, 2002.

[3] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, 25:25–29, 2000.

[4] T. Augustin. Entscheidungstheorie. Vorlesungsskript, 2007.

[5] H. Binder and M. Schumacher. Incorporating pathway information into boosting estimation of high-dimensional risk prediction models. *BMC Bioinformatics*, 10:18, 2009.

[6] BioCarta. http://www.biocarta.com/.

[7] A.-L. Boulesteix. *Dimension Reduction and Classification with High-Dimensional Microarray Data*. PhD thesis, Fakultät für Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universität München, 2005.

[8] A.-L. Boulesteix and C. Strobl. Optimal classifier selection and negative bias in error rate estimation: An empirical study on high-dimensional prediction.

BMC Medical Research Methodology (accepted). Technical Report, Department of Statistics, Ludwig-Maximilians Univ., 58, 2009.

[9] A.-L. Boulesteix, C. Strobl, T. Augustin, and M. Daumer. Evaluating microarray-based classifiers: an overview. *Cancer Informatics*, 6:77–97, 2008.

[10] U. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20:374–380, 2004.

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

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

[13] B. Efron and C. Morris. Stein's Paradox in Statistics. *Scientific American*, 236:119–127, 1977.

[14] F. Rapaport et al.. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:35, 2007.

[15] R. Tibshirani et al.. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99:6567–6572, 2002.

[16] T. R. Golub et al.. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286:531–537, 1999.

[17] V. Guillemot et al.. Graph-Constrained Discriminant Analysis of functional genomics data. IEEE International Conference on Bioinformatics and Biomedicine Worshops, 2008.

[18] L. Fahrmeir, A. Hamerle, and G. Tutz. *Multivariate statistische Verfahren.* Walter de Gruyter, Berlin, 1996.

[19] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175, 1989.

[20] P. A. Frost and J. E. Savarino. An empirical Bayes approach to portfolio selection. *Journal of Financial and Quantitative Analysis*, 21:293–305, 1986.

[21] GenMAPP. http://www.genmapp.com/.

[22] Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8:86–100, 2007.

[23] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[24] N. J. Higham. Computing A Nearest Symmetric Positive Semidefinite Matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

[25] John P. A. Ioannidis. Microarrays and molecular research: noise discovery? *The Lancet*, 365:454–455, 2005.

[26] John P. A. Ioannidis. Is Molecular Profiling Ready for Use in Clinical Decision Making? *Oncologist*, 12:301–311, 2007.

[27] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, D:428–432, 2005.

[28] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28:27–30, 2000.

[29] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahren, S. Tsoka, N. Darzentas, V. Kunin, and N. Lopez-Bigas. Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research*, 33(19):6083–6089, 2005.

[30] M. Krull, S. Pistor, N. Voss, A. Kell, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapov, C. Choi, O. Kel-Margoulis, and E. Wingender. TRANSPATH: an information resource for storing and visualizing signal pathways and their pathological aberrations. *Nucleic Acids Research*, D:546–551, 2006.

[31] O. Ledoit and M. Wolf. Improved Estimation of the Covariance Matrix of

Stock Returns with an Application to Portfolio Selection. *Journal of Empirical Finance*, 10:603–621, 2003.

[32] O. Ledoit and M. Wolf. A Well-Conditioned Estimator for Large-Dimensional Covariance Matrices. *Journal of Multivariate Analysis*, 88:365–411, 2004.

[33] O. Ledoit and M. Wolf. Honey, I Shrunk the Sample Covariance Matrix. *Journal of Portfolio Management*, 31:110–119, 2004.

[34] C. Li and H. Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24:1175–1182, 2008.

[35] D. Nguyen and D. M. Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18:39–50, 2002.

[36] U. S. National Library of Medicine. http://www.pubmed.gov/. 2009.

[37] R. Opgen-Rhein and K. Strimmer. Accurate Ranking of Differentially Expressed Genes by a Distribution-Free Shrinkage Approach. *Statistical Applications in Genetics and Molecular Biology*, 6:9, 2007.

[38] R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

[39] M. Schena. *Microarray analysis*. Wiley-Liss, New York, 2003.

[40] J. Schäfer. *Small-Sample Analysis and Inference of Networked Dependency Structured from Complex Genomic Data*. PhD thesis, Fakultät für Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universität München, 2005.

[41] J. Schäfer and K. Strimmer. A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. *Statistical Applications in Genetics and Molecular Biology*, 4:32, 2005.

[42] R. Simon, M. D. Rademacher, K. Dobbin, and L. M. McShane. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute*, 95:14–18, 2003.

[43] M. Slawski, M. Daumer, and A.-L. Boulesteix. CMA - a comprehensive Bioconductor package for supervised classification with high dimensional data. *BMC Bioinformatics*, 9:439, 2008.

[44] M. Slawski, W. zu Castell, and Gerhard Tutz. Feature Selection Guided by Structural Information. *Technical Report, Department of Statistics, Ludwig-Maximilians Univ.*, 51, 2009.

[45] C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, 1955.

[46] F. Tai and W. Pan. Incorporating prior knowledge of gene functional groups into regularized discriminant analysis of microarray data. *Bioinformatics*, 23:3170–3177, 2007.

[47] R Development Core Team. R: A language and environment for statistical computing. Munich, Germany. ISBN 3-900051-12-7. http://www.R-project.org/. 2009.

[48] G. Tutz. Multivariate Verfahren. Vorlesungsskript, 2007.

[49] I. A. Wood, P. M. Visscher, and K. L. Mengersen. Classification based upon gene expression data: bias and precision of error rates. *Bioinformatics*, 23:1363–1370, 2007.

[50] H. Zhou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67:301–320, 2005.

List of Figures

List of Tables

Erklärung

Hiermit versichere ich, dass ich, Monika Jelizarow, die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

München, den 21. Dezember 2009 .....

Monika Jelizarow