# Discriminant Analysis for Longitudinal Data with Application in Medical Diagnostics

Mareike Kohlmann

München 2010

# Discriminant Analysis for Longitudinal Data with Application in Medical Diagnostics

**Mareike Kohlmann**

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

in Zusammenarbeit mit dem
Biostatistics Department der Roche Diagnostics GmbH
in Penzberg

vorgelegt von
Mareike Kohlmann
aus Bad Mergentheim

München, den 7. Oktober 2010

Erstgutachter: Prof. Dr. Leonhard Held

Zweitgutachter: Prof. Dr. Ludwig Fahrmeir

Tag der mündlichen Prüfung: 8. Dezember 2010

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Leonhard Held. His guidance, encouragement and patience have provided a fruitful basis for my dissertation. His commitment made the long-distance collaboration possible. I am grateful to my second supervisor Ludwig Fahrmeir for his constructive advice and his continuous local support in Munich. I also wish to thank Manuel Eugster and Friedrich Leisch who taught me how to create R packages and made the programming part very enjoyable.

The Biostatistics Department at Roche Diagnostics in Penzberg gave me the fantastic opportunity to undertake my doctoral studies in conjunction with industry. I very much appreciate the pleasant and supportive working atmosphere as well as the financial support. Especially Veit Peter Grunert, Andrea Geistanger, Friedemann Krause and Christoph Berding contributed to such a successful cooperation.

It is a pleasure to thank my friends, former and present colleagues. My special thanks go to Stefanie Kalus who provided me with a great all-embracing support - from invaluable statistical advice to personal friendship. A big thank to my office-mates Tanja Schweitzer and Christiane Dargatz for good company and much fun. I also wish to thank Gerhard Krennrich who inspired me in numerous discussions about statistics in the final phase of the dissertation. An enormous thank you to Alexander Speed for his belief in me and his excellent and dedicated linguistic assistance in English.

My parents deserve special mention for their unlimited support in all its forms. Sonja, Simon and Lukas, thanks for being supportive and caring siblings. I owe my loving thanks to Patrick. He has been a great source of patience, understanding and mental support at any time.

# Abstract

Classification methods for longitudinal data bear the potential to identify classifiers that are superior to those based on cross-sectional data. Recently, the univariate longitudinal quadratic discriminant analysis (longQDA) was proposed for such purposes. Its key idea is to use marginal means and covariance matrices of linear mixed models as group-specific plug-in estimators for the discriminant rule. This dissertation investigates some of the unaddressed issues as model selection and several multivariate extensions. A complementary software implementation in R is presented which fulfills state-of-the-art design and user requirements. Longitudinal biomarker data from diagnostic studies that are assessed for their potential to classify patients as therapy-resistant or not serve as motivating applications.

First, we compare two model selection criteria for determining the most appropriate univariate linear mixed model structure for each group and quantify the corresponding bias of an incorrect decision. The first criterion selects the model structure that yields the best classification performance. The second selects the model with the minimal Bayesian information criterion and performs better in our simulation study. The bias of an incorrect decision turns out to be higher for longer data profiles and more complex longitudinal models with random effects. Subsequently, we present multivariate extensions of longQDA. Two multivariate mixed model classes with a parsimonious parametrization are proposed: multivariate random effects models and covariance pattern models with a Kronecker product structure. With a special set-up of the data, estimation algorithms implemented for the univariate case are used for the first model class. The restricted maximum likelihood estimation of Kronecker product models is accomplished by a numerical constraint optimization algorithm.

Finally, we introduce the R package `longQDA` for executing quadratic discriminant analysis with longitudinal data. Beyond the statistical methodology presented in this dissertation, the entire process of data analysis up to the reporting of the results is supported. The software implementation follows the modern object-orientated concept with S4 classes and fulfills conceptual requirements such as a user-friendly handling, a good run-time performance and easy extensibility. The latter quality criterion is demonstrated for two features: the functionalities for multivariate data settings and its use in simulation studies.

# Zusammenfassung

Klassifikationsmethoden für Longitudinaldaten bergen das Potenzial bessere Klassifikatoren zu identifizieren als es mit Querschnittsdaten möglich wäre. Die univariate longitudinale quadratische Diskriminanzanalyse (longQDA) wurde kürzlich dafür vorgeschlagen. Die Hauptidee ist, marginale Mittelwerte und Kovarianzmatrizen von linearen gemischten Modellen als gruppen-spezifische Plug-in-Schätzer für die Diskriminanzregel zu verwenden. Diese Dissertation untersucht noch nicht bearbeitete Themen wie die Modellselektion und verschiedene multivariate Erweiterungen. Sie stellt eine Software-Implementierung in R vor, die moderne Design- und User-Anforderungen erfüllt. Als motivierende Anwendungsbeispiele werden longitudinale Biomarkerdaten aus diagnostischen Studien verwendet. Es soll deren Potential bewertet werden, Patienten als therapieresistent oder nicht resistent zu klassifizieren.

Zuerst werden zwei Modellselektionskriterien verglichen, um die Struktur des linearen gemischten Modells für jede Gruppe zu bestimmen und die entsprechende Verzerrung einer inkorrekten Entscheidung zu quantifizieren. Das erste Kriterium wählt die Modellstruktur, die die beste Klassifikations-Performance liefert. Das zweite wählt das Modell mit dem minimalen Bayesschen Informationskriterium und erweist sich in einer Simulationsstudie als das bessere Kriterium. Die Verzerrung durch eine inkorrekte Entscheidung ist für längere Datenprofile und komplexere longitudinale Modelle mit zufälligen Effekten höher. Anschließend werden multivariate Erweiterungen der longQDA vorgestellt. Zwei multivariate Klassen der gemischten Modelle mit sparsamer Parametrisierung werden vorgeschlagen: multivariate Modelle mit zufälligen Effekten und Modelle mit einer Kronecker-Produkt-Struktur in der Kovarianzmatrix. Mit einer speziellen Datensatzstruktur können Schätzalgorithmen, die für den univariaten Fall entwickelt wurden, für die erstgenannte Modellklasse verwendet werden. Die restringierte Maximum-Likelihood-Schätzung der Kronecker-Produkt-Modelle erfolgt mit einem numerischen Optimierungsalgorithmus unter Nebenbedingungen.

Zuletzt wird das R-Paket `longQDA` für die quadratische Diskriminanzanalyse mit Longitudinaldaten präsentiert. Über die statistische Methoden dieser Dissertation hinaus unterstützt es den gesamten Datenanalyse-Prozess bis hin zur Dokumentation der Ergebnisse. Die Software-Implementierung beruht auf dem modernen objektorientierten Konzept von S4 und überzeugt durch Benutzerfreundlichkeit, eine gute Laufzeit-Performanz und einfache Erweiterungsmöglichkeiten. Das letztgenannte Qualitätskriterium wird anhand von zwei Features dargestellt, der Funktionalität für multivariate Daten und dem Einsatz in Simulationsstudien.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

> *It appears to me a most excellent thing for the physician to cultivate Prognosis; for by foreseeing and foretelling, in the presence of the sick, the present, the past, and the future, [...], he will be the more readily believed to be acquainted with the circumstances of the sick; so that men will have confidence to intrust themselves to such a physician. And he will manage the cure best who has foreseen what is to happen from the present state of matters.*
>
> Hippokrates (300 B.C.)

Already in ancient times, correct patient prognoses were indicative for professional medical work of high quality. The developments in the diagnostics field are nowadays so advanced that we operate at a more microbiological level: Body liquids or genes are searched for so called biomarkers to aid in diagnosis, prognosis or treatment issues. This is a typical area for the statistical application of supervised learning algorithms such as discriminant analysis. One aim is to determine the biomarker that discriminates best between patient groups, another one is to predict the group membership for future patients: Once a model has been finalized and the discriminant function has been derived, how well can we predict to which group a particular patient belongs?

In most applications in the literature (e.g. in Pepe, 2003; Zhou et al., 2002), biomarkers are measured only once for each patient. This may be a feasible approach for the most common case where the diagnosis of a patient is of interest. But if the effect of a therapy or any other physical process or time-dependent response is of interest, we are faced with a time-changing phenomenon. Then it is more appropriate to consider longitudinal profiles instead of using just one single measurement of a biomarker: We expect that characteristics of a (disease) process are reflected in turn in the profiles of a predictive biomarker. In fact, the definition of a biomarker implies repeated measurements over time: "A biomarker is a molecule that indicates an alteration of the physiological state of an individual in relation to health or disease state, drug treatment, toxins, and other challenges of the environment. By this definition, a biomarker is not static, it is changing over time." (Zolg and Langen, 2004). Research questions about change are of interest. Consequently, time needs to be handled as a predictor and longitudinal data analysis is the method of choice.

Marshall and Barón (2000) and Tomasko et al. (1999) introduced the longitudinal quadratic discriminant analysis (longQDA) for such purposes. It extends the Bayesian classification method quadratic discriminant analysis (QDA, McLachlan, 1992) in that the group parameters for the discriminant rule are estimated by linear mixed models (Verbeke and Molenberghs, 2000; Weiss, 2005). Also known as multilevel models or hierarchical linear models, these models are especially adequate for longitudinal data with their inherent time dependencies. For non-linear time structures which are more probable for long biomarker profiles, the estimation may be easily adapted (De la Cruz-Mesia and Marshall, 2006).

The longitudinal data structure induces an issue that is irrelevant in discriminant analysis for cross-sectional data: the selection of the most appropriate model for the biomarkers. The high flexibility of linear mixed models allows for many model structures, especially for the covariance structure. For the standard modelling setting where the sole aim is the estimation of the linear mixed model and eventually the testing of hypotheses, Singer and Willett (2003) stress the importance of the selection of the respective covariance structure. This issue is even of higher importance for longQDA. The estimation of the mean and of the covariance parameters are of equal interest as both are plugged in the discriminant function. In the standard setting, however, when the estimation of the model is the sole analysis goal, the focus is more on the mean structure than on the covariance parameters which determine "only" the precision and consequently the significance of the statistical tests. Furthermore, there is an interest in the effects of an incorrect model structure on the assessment of the biomarker performance. A comprehensive simulation study was carried out to address these issues dependent on the length of the biomarker profiles. Model selection was either based on various performance measures, i.e. the final results of longQDA, or on the Bayesian information criterion when estimating the mixed models, independent of the discriminant analysis task. This issue is either not addressed in the literature (Marshall and Barón, 2000; Brant et al., 2003) or was only based on the minimal error rate (Tomasko et al., 1999; Wernecke et al., 2004).

A further topic of this dissertation is the extension to multivariate longQDA where multivariate mixed models yield group-specific sample estimators. Note that in this context, univariate and multivariate always refers to the number of biomarkers, the outcomes of the mixed models. Strictly speaking, so called univariate mixed models are multivariate in the usual sense since a vector of repeated measurements is modelled, but all of the same outcome. Multivariate mixed models are presented in the following which extend the univariate mixed models which turned out to be appropriate for our biomarker settings. These are mainly multivariate random coefficients models with at least one random intercept. Regarding alternative models, the suitability of multivariate covariance pattern models are elaborated. The appealing idea of these models is to parameterise the covariance matrix parsimoniously as a Kronecker Product of the auto- and the cross-covariance. Both variants of multivariate mixed models are applied to data from a diagnostic biomarker study. Simulated data serve as examples for achieving a superior performance by combining longitudinal biomarkers.

Beside the methodological development, statisticians are more and more faced with software development. A wide-spread application of statistical methods is closely related to the availability of comfortable state-of-the art software implementations. Therefore, our `R` package called `longQDA` implements the longitudinal quadratic discriminant analysis with Monte Carlo cross validation, all performance measures and full flexibility in model selection. The software implementation follows object-orientated programming paradigms and fulfills requirements as user friendliness and easy extensibility and supports the reporting of analysis results. In a wider scope, it serves as an example for the implementation of a statistical method in compliance with requirements of users with statistical knowledge.

To summarize, the main focus of this dissertation is primarily on the following points:

- extend the knowledge in model selection issues in univariate longQDA: assess the performance not only by standard diagnostic measures such as the area under the receiver operating characteristic (ROC) curve but also by the predictive Brier score and contrast their usability in model selection with the Bayesian information criterion (BIC)
- propose a multivariate longQDA based on multivariate random effects models or covariance pattern models with a parsimonious Kronecker Product structure
- provide a software solution as an R package, which is comfortable to use and easy to extend due to the S4 implementation

The structure of this thesis follows those topics. The next chapter introduces longitudinal quadratic discriminant analysis to the reader and establishes the terminology in the diagnostic context. Performance measures are discussed to assess the classification potential of biomarkers. One application data set with a univariate longitudinal biomarker is presented and analysed by means of longQDA. Then the main topics given above are elaborated in Chapter 3-5, respectively. Chapter 6 concludes with the summary and the outlook.

# 2 Univariate Longitudinal QDA and Performance Measures

This chapter covers the main subject of the dissertation. First and foremost, this is the longitudinal QDA (longQDA). It extends the classic QDA for cross-sectional data in that the plug-in estimators for the discriminant rule are the resulting marginal means and covariances of modeling the longitudinal data by mixed models. Section 2.1 introduces univariate longQDA and especially mixed models that are appropriate for longitudinal biomarkers. In Section 2.2, several measures are presented to assess the predictive classification performance of biomarkers. This includes performance measures as the area under the receiver operating characteristic (ROC) curve which are most common in diagnostics as well as the Brier score and its subcomponents which are currently recommended for various statistical applications (Gneiting and Raftery, 2007; Gerds et al., 2008). Some notes follow about Monte Carlo cross validation which is a procedure for robust estimation. In Section 2.3, the univariate longQDA was applied to real data of HIV patients (taken from May and DeGruttola, 2007). This serves as one example for commonly encountered data in longitudinal diagnostic studies and illustrates the benefit for classification when measuring one biomarker longitudinally, i.e. several times during the study.

## 2.1 Univariate Longitudinal QDA

As in a QDA setting for cross-sectional data, there are two patient groups $k = 0, 1$ defined by the Gold Standard (reference classification) and the indicator function $z_{i[1]}$ defines the membership of patient $i$ to group $k = 1$ where $i = 1, \dots, n_{[1]}$. Thus, there are $N = n_{[0]} + n_{[1]}$ patients in the study. As all applications deal with biomarkers indicative for therapy response, group 1 consists always of resistant patients and group 0 of non-resistant patients. The key characteristic for longQDA is that biomarkers are not only measured once, but several times during a longitudinal diagnostic study, leading to one biomarker vector $\boldsymbol{w}_i = (w_{i1}, \dots, w_{ip})^T$ for patient $i$. Given group $k$, a $p$-variate conditionally normal density $f(\boldsymbol{w}_i | z_{i[1]} = k)$ is assumed for the $p$-times repeatedly measured biomarker $\boldsymbol{w}_i$, i.e. $\boldsymbol{w}_i | z_{i[1]} = k \sim N(\boldsymbol{\mu}_{i[k]}, \boldsymbol{V}_{i[k]})$. To account for the repeated biomarker measurements, an obvious approach is to estimate the mean $\boldsymbol{\mu}_{i[k]}$ and the covariance matrix $\boldsymbol{V}_{i[k]}$ by a linear mixed model. Let $\boldsymbol{w}_{((n_{[k]}\cdot p)\times 1)} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_{[k]}})^T$ be the $(n_{[k]} \cdot p) \times 1$-dimensional vector containing observations of $i = 1, \ldots, n_{[k]}$ patients with measurements taken at $j = 1, \ldots, p$ visits at time $\boldsymbol{t}_{ij}$ for group $k$. To ease readability, the group index $k$ indicating that the biomarker density is defined conditionally on $z_{i[1]} = k \; \forall i = 1, \ldots, n_{[k]}$ is omitted in all mixed models from now on. For patient $i$, a linear mixed model is defined for the univariate response variable $\boldsymbol{w}_i$ in each group as

$$\boldsymbol{w}_{i(p\times 1)} = \boldsymbol{X}_{i(p\times u)}\boldsymbol{\beta}_{(u\times 1)} + \boldsymbol{Z}_{i(p\times s)}\boldsymbol{b}_{i(s\times 1)} + \boldsymbol{\epsilon}_{i(p\times 1)} \tag{2.1}$$

where $\boldsymbol{b}_i \sim N(\boldsymbol{0}_{(s\times 1)}, \boldsymbol{D}_{(s\times s)}), \boldsymbol{\epsilon}_i \sim N(\boldsymbol{0}_{(p\times 1)}, \boldsymbol{R}_{i(p\times p)})$. The parameters $\boldsymbol{b}_i$ and $\boldsymbol{\epsilon}_i$ are independent and their covariance matrices are assumed to be positive definite. Further, $\boldsymbol{X}_i$ is a $(p \times u)$ matrix of known covariates, modelling how the biomarker evolves over time for subject $i$, and $\boldsymbol{\beta}$ is a $u$-dimensional vector of unknown population-specific regression parameters.

The marginal normal distribution of $\boldsymbol{w}_i$ has the mean $\boldsymbol{\mu}_i = \boldsymbol{X}_i\boldsymbol{\beta}$ and the covariance matrix $\boldsymbol{V}_i = \boldsymbol{Z}_i\boldsymbol{D}\boldsymbol{Z}_i^T + \boldsymbol{R}_i$. Combining the $n_{[k]}$ patient-specific regression models (Eq. (2.1)), the model is given by

$$\boldsymbol{w} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{\epsilon} \tag{2.2}$$

where the vectors $\boldsymbol{w}$, $\boldsymbol{b}$, $\boldsymbol{\epsilon}$ and the matrix $\boldsymbol{X}$ are obtained from stacking the vectors $\boldsymbol{w}_i$, $\boldsymbol{b}_i$, $\boldsymbol{\epsilon}_i$ and the matrix $\boldsymbol{X}_i$, respectively, underneath each other. The matrix $\boldsymbol{Z}$ is block-diagonal with blocks of $\boldsymbol{Z}_i$ on the main diagonal and zeros elsewhere. Due to the commonly assumed independence between patients in mixed models, $\boldsymbol{R}$ is a covariance matrix with a block-diagonal structure of $n_{[k]}$ sub-matrices $\boldsymbol{R}_i$. This is consequently also true for $\boldsymbol{V}$, hence $\boldsymbol{w} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{V})$.

Biomarkers are measured at the beginning of the study and at pre-defined scheduled visits thereafter. As individual visit times $\boldsymbol{t}_i$ usually vary from that time schedule and have different time lags in between, mixed models that can incorporate non-equidistant measurements need to be selected. The selection comprises random coefficients models that assume a random intercept for each patient (RI) or a random intercept and a linear random slope (RIS) or extend the RI model by assuming a continuous AR(1) structure for the residual matrix (RICAR1). Pure covariance pattern models were not considered, the focus was on mixed model structures (MMS) that allow for variations at an individual level, at least at baseline. The number of measurements is usually rarely larger than ten and thus models with more complicated stochastic dependencies as proposed by Taylor et al. (1994) or Munoz et al. (1992) were not considered.

For clarity, here are the parameters of the considered covariance structure in univariate longQDA:

RI: $s = 1$, i.e. $\boldsymbol{Z}_i = (1, \dots, 1)^T$, $\boldsymbol{D} = d_{\mathrm{I}}^2$,
$\boldsymbol{R}_i = \sigma^2\, \boldsymbol{I}$

RIS: $s = 2$, i.e. $\boldsymbol{Z}_i = \begin{pmatrix} 1 & t_{i1} \\ \vdots & \vdots \\ 1 & t_{ip} \end{pmatrix}$, $\boldsymbol{D} = \begin{pmatrix} d_{\mathrm{I}}^2 & d_{\mathrm{IS}} \\ d_{\mathrm{IS}} & d_{\mathrm{S}}^2 \end{pmatrix}$,
$\boldsymbol{R}_i = \sigma^2\, \boldsymbol{I}$

RICAR1: $s = 1$, i.e. $\boldsymbol{Z}_i = (1, \dots, 1)^T$, $\boldsymbol{D} = d_{\mathrm{I}}^2$,
$\boldsymbol{R}_i = (r_{jj'})_i = \sigma^2[\phi^{(t_{ij} - t_{ij'})} 1(j \neq j') + 1(j = j')]$
with $0 < \phi < 1$ and $j, j' = 1, \dots, p$.

The matrix $\boldsymbol{I}$ is the $p$-dimensional identity matrix, $d_{\mathrm{I}}^2$ resp. $d_{\mathrm{S}}^2$ is the variance of the random intercept resp. of the linear slope and $d_{\mathrm{IS}}$ is the covariance between the intercept and the linear slope.

Note that the RIS model should always be parameterized to allow for a correlation between the random intercepts and the random slopes as the correlation parameter changes due to a time shift and thus should not be constrained (Weiss, 2005, p.260f.).

The aim in quadratic discriminant analysis is to find a quadratic discriminant function $m(\boldsymbol{w}_i)$ for the classification of patients into groups defined by a Gold Standard. The discriminant function $m(\boldsymbol{w}_i)$ for longQDA is the same as for QDA, the log-ratio of $f(\boldsymbol{w}_i | z_{i[1]} = 1)$ to $f(\boldsymbol{w}_i | z_{i[1]} = 0)$. Future patients are classified according to the following discriminant rule: If $m(\boldsymbol{w}_i) \geq \tau'$, where $\tau'$ is a selected cut-off value, then patient $i$ with biomarker measurements $\boldsymbol{w}_i$ is assigned to group $k = 1$. If $m(\boldsymbol{w}_i) < \tau'$, the patient is assigned to group $k = 0$.

Now let $\pi_{[1]}$ be the prior probability that a patient belongs to the group with biomarker density $f(\boldsymbol{w}_i | z_{i[1]} = 1)$ prior to observing $\boldsymbol{w}_i$, and set $\pi_{[1]} = 1 - \pi_{[0]}$. For normal distributions, the optimal discriminant rule, in the sense of minimizing the expected probability of misclassification, is given by $\tau' = \log(\pi_{[0]}/\pi_{[1]})$. This is also known as the Bayesian decision rule (Johnson and Wichern, 2002). As the posterior probability is proportional to the product of the likelihood and the prior probability (resp. equal to after multiplication by the normalising constant), the classification rule is based on the posterior probability $p_{i[1]}$ with

$$p_{i[1]} = P(\hat{z}_{i[1]} = 1 | \boldsymbol{w}_i) = \frac{\exp\Big[-\frac{1}{2}(\boldsymbol{w}_i - \boldsymbol{\mu}_{[1]})^T\, \boldsymbol{V}_{i[1]}^{-1}\, (\boldsymbol{w}_i - \boldsymbol{\mu}_{[1]})\Big] \cdot \Big[|\boldsymbol{V}_{i[1]}|\Big]^{-\frac{1}{2}} \cdot \pi_{[1]}}{\sum_{k=0}^{1} \exp\Big[-\frac{1}{2}(\boldsymbol{w}_i - \boldsymbol{\mu}_{[k]})^T\, \boldsymbol{V}_{i[k]}^{-1}\, (\boldsymbol{w}_i - \boldsymbol{\mu}_{[k]})\Big] \cdot \Big[|\boldsymbol{V}_{i[k]}|\Big]^{-\frac{1}{2}} \cdot \pi_{[k]}}. \qquad (2.3)$$

The posterior probabilities $\boldsymbol{p}_{[1]} = (p_{1[1]}, \dots, p_{n_{[1]}[1]})^T$ constitute a univariate measure which summarizes the predictive classification information of the longitudinal biomarker profiles. They form the basis for the overall assessment of the biomarker performance which is covered in the next section.

To prevent an overoptimistic estimation of the biomarker performance, Monte Carlo cross validation (MCCV) with sampling without replacement (Shao, 1993) is applied. The principle of any cross validation procedure is to split the data repeatedly in two disjunct samples called training and test set. The training set is used for estimation purposes whereas the performance is then independently assessed with the test set. The repetition of this procedure reduces any possible systematic bias due to a possibly non-representative partitioning of the data by chance. The results are summarized over all repetitions. The approach is termed Monte Carlo cross validation as the samples are taken randomly, with a fixed number of repetitions (e.g. 50, independent of the size of the sample). Due to the random drawing, it is assured that data of one patient are used for training as well as for testing. For longitudinal data, all observations of one patient were sampled at once to preserve the correlation structure (Goldstein, 2003). Thus, patients and not single observations were sampled. The mixed model estimators for the discrimination rule were evaluated in the training sets and the rule was applied to the samples in the test sets yielding estimations for the performance measures. By the way, to preserve the group proportions, it is important to split the data sample into the training and test set for each group separately. Results are reported in terms of median estimates over all MCCV samples, and the empirical 10 and 90 percentiles are used as non-parametric confidence intervals (for theoretical justifications see Efron and Tibshirani, 1993).

## 2.2 Classification Performance Measures

To assess the performance of a biomarker to predict the group membership, we aim for a small number of performance measures providing the most comprehensive summary. For this purpose, measures that are either based on the receiver operating characteristic (ROC) curve (Pepe, 2003; Zhou et al., 2002) or on the Brier score (see e.g. Gneiting and Raftery, 2007) are presented in the following and contrasted with respect to their properties.

First, the common classification approach in diagnostics is elucidated. In the typical setting, a medical test is expected to give either a positive or a negative test result. To fulfill this requirement of a binary test result, continuous test results need to be dichotomized so that a test is said to be positive if the result exceeds or is equal to a selected threshold $\tau^*$. When biomarker data are assessed by classification methods as longQDA which yield a group-specific posterior probability for each patient, the probabilities for one group may take over the role of the continuous test results. By definition, these are the probabilities for group $k = 1$ as the test positivity is generally defined for the group with larger test results. In other words, $\tau^*$ is the parameter which determines at what value the posterior probability is dichotomized to yield a binary prediction $\hat{z}_{i[k]}$ corresponding to a positive resp. negative test result.

Usually, medical tests are imperfect in that the test results resp. the posterior probabilities of both groups overlap yielding misclassifications. This means that patients with false positive test results have posterior probabilities that are larger than or equal to $\tau^*$ but are members of group 0. Patients with false negative results are characterized by posterior probabilities which are smaller than $\tau^*$ but are members of group 1. Therefore, the group-specific probabilities that a patient is correctly classified may serve as performance measures.

The sensitivity is defined as $P(p_{i[1]} \geq \tau^* | \, z_{i[1]} = 1)$ and the specificity as $P(p_{i[1]} < \tau^* | \, z_{i[1]} = 0)$. They are also denoted as true positive and true negative fraction. These measures are estimated by the sample proportion of correctly classified patients of all patients in group 1 resp. group 0 according to the Gold Standard. This approach implies a decision given a 0-1-loss function (Friedman et al., 2009, Section 2.4) weighting all misclassifications equally by 1. Both measures assess the local (and not the global) performance of a diagnostic test as they are threshold-specific. Varying $\tau^*$ on the scale of the posterior probabilities yields all possible threshold-specific sensitivity and specificity pairs. A local performance criterion for a diagnostic test may be the maximisation of sensitivity and specificity. However, each measure can only be maximised at the expense of the other if the densities of the posterior probabilities overlap. Decreasing the thresholds $\tau^*$ increases the sensitivity. When the specificity is favoured over the specificity, $\tau^*$ should be decreased. The ROC curve is a graph that displays this trade-off (Figure 2.1).

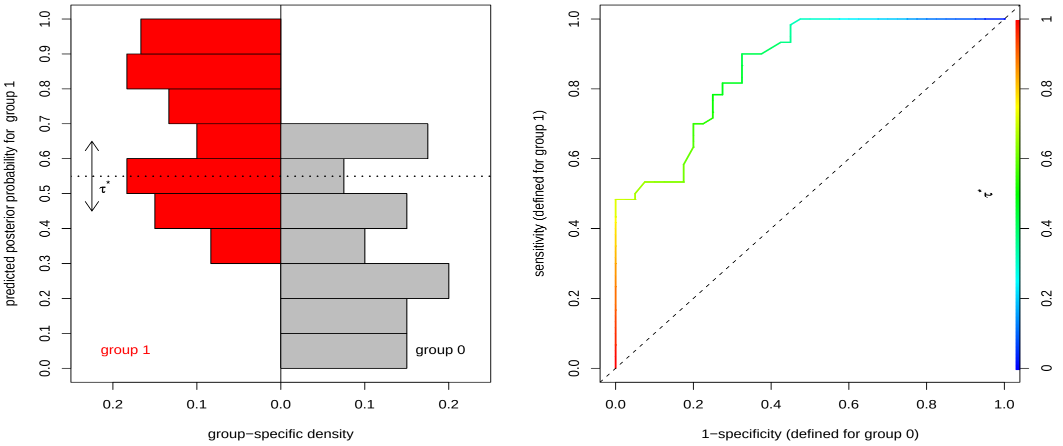


Figure 2.1: Histograms of simulated posterior probabilities for group 1 (left) and corresponding ROC curve (right).

The sensitivities are plotted on the y-axis as a function of the false positive fraction (1-specificity) over the entire range of possible thresholds. To assess the performance at a global level, the most commonly used summary measure in diagnostic settings is the area under the ROC curve (AUC). The AUC equals one for perfect classification and 0.5 for biomarkers with no discriminatory power. The AUC corresponds to an average performance over all possible decisions. This is a quite artificial construct with limited expressiveness.

A further performance measure is the Brier score (BS) which will be considered in details. In the two-group case, it is defined as

$$\mathrm{BS} = 1 - \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_{i[1]} - z_{i[1]})^2.$$

The score measures the discrepancy between the observed real group membership and the estimated posterior probability of the classification method using a squared error loss function. To ease the comparability of the performance measures, the BS is scaled as such that a maximum score of 1 is indicative for the best classification performance. It reaches its minimum at 0 in the unlikely case that all patients are misclassified with a group-specific probability equal to zero. The BS equals one minus the Gold Standard's variance $\sigma^2_{z_{[1]}}$ for a constant prediction with a probability equal to the prior $\pi_{[1]}$. This provides a threshold of the BS and assesses a prediction based solely on the a priori knowledge.

Ikeda et al. (2002) study the relationship between AUC and BS but could not identify a linkage between the two. This is not surprising as there are differences regarding the content. The AUC and the sensitivity and specificity are performance measures that indicate how effective the classification rule is in assigning a patient to the correct group and are therefore called accuracy measures by Hand (2001). The performance is only evaluated in terms of correct and misclassified patients without respect to the probability with which the patient is classified into a certain group. The BS, however, provides information about the relative severities of misclassifications by placing different weights on misclassified as well as on correctly classified patients. This score overcomes the shortcomings of the AUC which gives no indication of the degree of confidence one should place in a classification. A distinction should be made, for example, between the classification of one patient with a high probability for one group, say 0.99, and another one with a much lower probability near 0.5. In an assessment by the AUC, however, both are treated the same.

A more detailed view based on the decompositions of the Brier score proposed by Sanders (1963), Murphy (1973) and Yates (1982) illustrate that the BS enables a more comprehensive assessment of the biomarker performance. For Sanders' decomposition, the posterior probabilities for one group (here for $k = 1$) are categorized into say 5 cells $c = 1, \ldots, 5$ (The number of cells depends on the desired precision and the size of the patient sample.). The patients in cell $c$ are defined by $\mathcal{J}_c := \{i = 1, \ldots, N | \, p_{i(1)} \in \mathcal{M}_c\}$ with $\mathcal{M}_c = [\frac{c-1}{5}; \frac{c}{5})$, $c = 1, \ldots, 4$ and $\mathcal{M}_5 = [\frac{4}{5}; 1]$. Thus, $|\mathcal{J}_c| = n_c$ and $N = \sum_{c=1}^{5} n_c$. With this discretisation of the posterior probabilities, the BS is partitioned as follows:

$$\begin{aligned} \text{BS} &= 1 - \frac{1}{N}\sum_{c=1}^{5}\sum_{i\in\mathcal{J}_c}(\hat{p}_{i[1]} - z_{i[1]})^2 \\ &= 1 - \Big(\frac{1}{N}\sum_{c=1}^{5} n_c \bar{z}_{[1]c}(1 - \bar{z}_{[1]c}) + \frac{1}{N}\sum_{c=1}^{5} n_c(\hat{\bar{p}}_{[1]c} - \bar{z}_{[1]c})^2\Big) \\ &= 1 - (\text{BS}_D + \text{BS}_C) \end{aligned}$$

where $\bar{z}_{[1]c}$ is the proportion of $n_c$ patients who are members of group 1 defined by the Gold Standard and $\bar{p}_{[1]c}$ is the corresponding expected proportion. The latter is estimated by the mean posterior probability of cell $c$, averaged over $n_c$ patients. The measure $\text{BS}_D$ assesses the discrimination performance and $\text{BS}_C$ the calibration performance (a.k.a. precision). According to Gurney and Swensen (1995), calibration values indicate the ability of the classification rule to assign numeric probabilities, discrimination values indicate the ability to distinguish outcomes (e.g. resistant vs. non-resistant patients). Murphy (1973) partitions the discrimination component of Sanders further:

$$\text{BS}_D = \bar{z}_{[1]}(1 - \bar{z}_{[1]}) - \frac{1}{N}\sum_{c=1}^{5} n_c(\bar{z}_{[1]c} - \bar{z}_{[1]})^2$$

where $\bar{z}_{[1]}$ is the proportion of patients in group 1 in the sample. He proposes this decomposition as the variance of the Gold Standard is determined "by nature" and does not depend on the performance of the biomarker which should actually be assessed. The second term is called Murphy's discrimination component $\text{BS}_{D_M}$ and is preferred over $\text{BS}_D$. Its upper bound is the Gold Standard's variance $\sigma^2_{z_{[1]}}$, which is the first term of $\text{BS}_D$.

It is especially important to take the calibration into account when decisions are made on an individual basis, whether a clinician wants to know with which probability a patient is classified into one of the groups or whether classification rules based on different model assumptions or biomarkers are compared by the statistician. In both cases, the posterior probabilities should be interpretable in a frequentistic sense. Diamond (1992) points out that the optimization of one component often goes at the expense of the other due to different implied scales of the measures: The discrimination supports a categorical predicted result (or one that can be at least discretized in two categories without much loss), whereas the calibration is maximised at the expense of discrimination. Therefore good calibration does not imply good discrimination, and vice versa. For example, a classification rule that predicts the membership to group 1 with a constant probability equal to the group proportion $\bar{z}_{[1]}$, i.e. without adding any additional information to improve the prediction, has a good calibration but a bad discrimination. One example for a rule with perfect discrimination but poor calibration would be the following. Given group proportions of 2:1, a rule predicts with uniformly distributed $p_{i[1]} \in [0; 0.2]\ \forall i \in 1, \dots n_{(0)}$ and $p_{i[1]} \in [0.21; 1]\ \forall i \in 1, \dots n_{(1)}$, resulting in two separated probability densities. The BS takes both performance aspects into account whereas the AUC is only a measure for the discriminative power of a classifier.

A related limitation of the AUC becomes obvious when considering the quality requirement for a classification rule (also known as scoring rule) to be strictly proper (Gneiting and Raftery, 2007; Hand, 1997). A strictly proper classification rule is characterized by attaining its maximum expected value given the posterior probability distribution if and only if the true probabilities for the patient's group membership are used. The BS has the desirable property to be a strictly proper rule but the AUC does not. The AUC fulfills only the weaker condition of non-strictly properness: The true probabilities may yield the same performance as others, not necessarily a higher one. For example, the AUC also reaches its maximum if all patients are classified into the correct group, but with uncertain posterior probabilities close to 0.5.

Another decomposition of the Brier Score (Yates, 1982) is based on the covariance decomposition and does not require a discretisation of the posterior probabilities. It provides performance measures which are more common in statistics. According to the well known decomposition of the mean squared error MSE = variance + bias$^2$, the decomposition is

$$\begin{aligned} \mathrm{BS} &= 1 - \left(\sigma^2_{z_{[1]}} + \sigma^2_{p_{[1]}} - 2\sigma_{z_{[1]},p_{[1]}} + (\bar{p}_{[1]} - \bar{z}_{[1]})^2\right) \\ &= 1 - \Big(\bar{z}_{[1]}(1-\bar{z}_{[1]}) + \Delta\sigma^2_{p_{[1]}} + \sigma^2_{p_{[1]},min} - 2(\bar{p}_{[1|\,z_{[1]}=1]} \\ &\quad -\bar{p}_{[1|\,z_{[1]}=0]})\bar{z}_{[1]}(1-\bar{z}_{[1]}) + (\bar{p}_{[1]} - \bar{z}_{[1]})^2\Big) \end{aligned}$$

where $\Delta\sigma^2_{p_{[1]}} = \sigma^2_{p_{[1]}} - \sigma^2_{p_{[1]},min}$ is the "excess" variability with the conditional minimum variance $\sigma^2_{p_{[1]},min} = (\bar{p}_{[1|\,z_{[1]}=1]} - \bar{p}_{[1|\,z_{[1]}=0]})^2\bar{z}_{[1]}(1-\bar{z}_{[1]})$, $\bar{p}_{[1]}$ is the estimated overall mean posterior probability for group $k = 1$; $\bar{p}_{[1|\,z_{[1]}=1]}$ and $\bar{p}_{[1|\,z_{[1]}=0]}$ are, respectively, the mean posterior probabilities for group $k = 1$ resp. $k = 0$ defined by the Gold Standard. Regarding the variance of the posterior probabilities, we should aim for a classification rule with a minimal variance $\sigma^2_{p_{[1]}}$. But the only way $\sigma^2_{p_{[1]}}$ reaches its minimum possible value zero is when the posterior probabilities are the same for all patients and then the covariance $\sigma_{z_{[1]},p_{[1]}}$ is zero as well. Thus the proper objective to minimize $\sigma^2_{p_{[1]}}$ is to condition on the relationship between the Gold Standard and the biomarker $\sigma_{z_{[1]},p_{[1]}}$. The conditional minimum prediction variance given $\sigma_{z_{[1]},p_{[1]}}$ is $\sigma^2_{p_{[1]},min}$. The ratio between the excess variation and the conditional minimum prediction variance $\mathrm{Q}_{\mathrm{Var}} = \Delta\sigma^2_{p_{[1]}}/\sigma^2_{p_{[1]},min}$

should be small and is inversely proportional to the 2-sample t-test for comparing the group-specific predictions $\bar{p}_{[1|\,z_{[1]}=1]}$ and $\bar{p}_{[1|\,z_{[1]}=0]}$ (Spiegelhalter, 1986). Besides $\mathrm{Q_{Var}}$, the covariance $\sigma_{z_{[1]},p_{[1]}}$ between the Gold Standard and the posterior probabilities serves as performance measure. It is advisable to standardize this covariance to yield the point biserial correlation $\rho_{z_{[1]},p_{[1]}}$. Note the smaller range compared to correlation coefficients for two continuous variables. The range depends on $\bar{z}_{[1]}$, the limits are given in Gradstein (1986). The separation of the posterior probability distributions of the two groups may be assessed by $\mathrm{Diff}_{\bar{p}_{[1]}} = \bar{p}_{[1|\,z_{[1]}=1]} - \bar{p}_{[1|\,z_{[1]}=0]}$. The bias $(\bar{p}_{[1]} - \bar{z}_{[1]})$ indexes a performance characteristic labelled calibration-in-the-large, $Cal_L$, which reflects the ability of the biomarker to match mean posterior probabilities to the proportion of patients in group $k = 1$ defined by the Gold Standard. It is interpreted as the "baseline knowledge" about group membership.

All in all, the Brier score with its derived performance measures allows for a more profound assessment of the biomarker performance than the ROC curve related measures by taking the actual location and shape of the group-specific posterior probability distributions in account in various ways. The AUC is reported as it is the most commonly used measure in diagnostic medicine. Other authors (Spiegelhalter, 1986; Harrell, 2006) make similar compromises regarding their choice of performance measures.

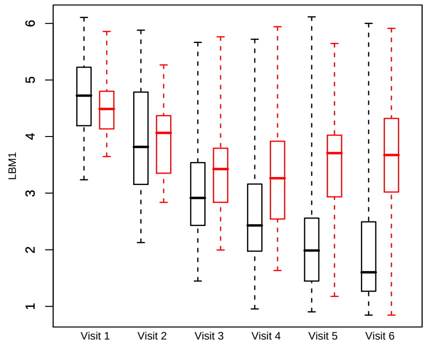
## 2.3 Application of Longitudinal QDA: HIV Therapy Resistance Data

The application of longQDA is illustrated with data of one biomarker that is indicative for HIV therapy resistance. The data were recently presented in the literature (May and DeGruttola, 2007) and are freely available. The aim is to classify patients based on their viral load either as resistant or non-resistant to specific HIV treatments (Nevirapine, Delavirdine and Efavirenz) and thus to replace the need of phenotype or genotype data. The group with the worse condition is denoted by $k = 1$, hence resistant patients are members of group $k = 1$ and non-resistant patients of group $k = 0$. The presence of resistance is assumed to be perfectly determined by the presence of the reported K103N mutation of HIV (although this is possibly idealistic for a test assay based on phenotype or genotype data). Using this Gold Standard, there are 292 patients with no resistance and 64 who exhibited the mutation. The viral load is measured by the amount of HIV RNA and up to 6 measurements (at baseline, after 2, 4, 8, 16 and 24 weeks) are available for each patient. The analysis was restricted to patients with complete data to enable a fair comparison of the results between single visits with the same data base. Biomarker data of 59 non-resistant and 26 resistant patients were considered. This implies a non-informative drop-out and a missing at random mechanism which was assumed for simplicity. Otherwise, other methods need to be applied (Hogan et al., 2004; Thiébaut et al., 2005).

For all QDA and longQDA analyses, 50-fold MCCV was applied and patients were randomly allocated to the training or the test set according to a group-wise ratio of 2:1. The priors were fixed to the group proportions of the entire dataset (not the one reduced to the complete cases), assuming that these are good estimations for the "prevalence" of therapy resistance. So $\pi_{[1]}$ was 0.18 and hence the lower bound for the BS 0.852, the range of the biserial correlation $[-0.683; 0.683]$.

The biomarker measurements were log10-transformed (labelled LBM1). The resulting group-specific profiles on this scale are depicted in Figure 2.4 below (ignore the colouring). Profiles of non-resistant patients decrease quadratically whereas those of the resistant group have a rather unchanged, more linear shape from the third visit on. At baseline, the non-resistant patients have a wider range of initial biomarker values but the variation during the treatment is higher for the resistant ones. Besides, an atypical subgroup is present in the non-resistant group whose profiles do not decline over time and resemble rather the profiles of patients who were classified as resistant by the Gold Standard.

A comparison of the biomarker levels at group level show that from the fifth visit on, the ranges of the boxplots do not overlap at all (Figure 2.2(a)), indicating a good separation between the groups. The variances of the viral load measurements are especially high from the third visit on in both groups.



(a) Boxplots are given for the scheduled visit; for non-resistant patients in black, for resistant patients in red.

| Visit | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | ⋱ | 0.59 | 0.50 | 0.42 | 0.27 | 0.26 |
| 2 | 0.55 | ⋱ | 0.66 | 0.50 | 0.46 | 0.38 |
| 3 | 0.57 | 0.45 | ⋱ | 0.85 | 0.74 | 0.71 |
| 4 | 0.35 | 0.36 | 0.73 | ⋱ | 0.88 | 0.85 |
| 5 | 0.33 | 0.42 | 0.67 | 0.83 | ⋱ | 0.95 |
| 6 | 0.22 | 0.36 | 0.56 | 0.79 | 0.92 | ⋱ |

(b) Autocorrelations for non-resistant patients are given in the upper, for resistant patients in the lower triangle.

Figure 2.2: Boxplots and autocorrelations of log-transformed HIV RNA

The empirical autocorrelations were estimated for all patients with complete measurements (Table 2.2(b)). Also for correlations of visits more than one lag apart, they are of medium or large size, higher than 0.6 from the third visit on. At the first and second visit, this is only true for those with lag 1. Both groups have autocorrelations of similar size but they are more or less higher in the non-resistant group (one exception: coefficient of first and fifth visit). As the time dependencies in the biomarker data are of considerable size, they should be adequately taken into account by using mixed models to estimate the group-specific parameters.

### 2.3.1 Biomarker Performance at Single Visits by QDA

Before applying longQDA, the data are further explored by examining the performance of the biomarker at single visits. This provided insight into which visits contribute most to the longitudinal performance. Except for the calibration performance, an improvement in performance is observed over time (Table 2.1(a)). For the calibration it is vice versa, the worst is achieved at visit 5 or 6. The performance of the 3rd visit is for most of the measures as weak as at baseline and there are only small performance differences between visit 5 and 6.

(a) Comparing QDA with single visits.

| Visit | AUC | BS | $\text{BS}_C^{[-]}$ | $\text{BS}_{D_M}$ | $\rho_{z_{[1]},p_{[1]}}$ | $\ln(\text{Q}_{\text{Var}}^{[-]})$ |
|---|---|---|---|---|---|---|
| 1 | .55 [.43; .66] | .79 [.77; .80] | .003 [.001; .024] | .01 [.00; .01] | .09 [−.04; .25] | 8 [7; 11] |
| 2 | .69 [.53; .78] | .81 [.78; .82] | .009 [.010; .028] | .02 [.01; .05] | .30 [ .07; .41] | 5 [5; 7] |
| 3 | .61 [.51; .73] | .79 [.76; .81] | .007 [.001; .045] | .01 [.00; .03] | .14 [−.04; .39] | 6 [5; 9] |
| 4 | .66 [.56; .78] | .81 [.78; .82] | .011 [.003; .039] | .03 [.01; .05] | .27 [ .10; .42] | 5 [4; 7] |
| 5 | .71 [.60; .85] | .83 [.78; .85] | .030 [.007; .096] | .08 [.05; .12] | .41 [ .23; .59] | 3 [3; 4] |
| 6 | .77 [.64; .87] | .84 [.77; .86] | .026 [.009; .102] | .08 [.05; .13] | .49 [ .27; .62] | 3 [2; 3] |

(b) Comparing longQDA with 3, 4, 5 and 6 visits.

| # of Visits | AUC | BS | $\text{BS}_C^{[-]}$ | $\text{BS}_{D_M}$ | $\rho_{z_{[1]},p_{[1]}}$ | $\ln(\text{Q}_{\text{Var}}^{[-]})$ |
|---|---|---|---|---|---|---|
| 3 | .66 [.55; .77] | .80 [.74; .83] | .033 [.008; .091] | .04 [.02; .07] | .24 [.04; .43] | 4 [4; 6] |
| 4 | .65 [.55; .74] | .80 [.75; .83] | .027 [.005; .072] | .03 [.01; .06] | .26 [.04; .42] | 4 [3; 7] |
| 5 | .74 [.62; .84] | .82 [.76; .86] | .024 [.008; .080] | .06 [.03; .09] | .40 [.23; .57] | 3 [2; 4] |
| 6 | .75 [.66; .85] | .80 [.72; .86] | .056 [.020; .135] | .08 [.05; .12] | .44 [.28; .60] | 2 [1; 3] |

Table 2.1: Selected performance measures comparing (long)QDA, given as median with the 10th and 90th percentile. For all performance measures indicated by $^{[-]}$, lower values are indicative for a better performance. For the others, it is vice versa. The results of the bias for $Cal_L$ are similar to those of $BS_C$ as is the difference of the group-specific average posterior probabilities to the discrimination measures. They are therefore omitted in the table.

### 2.3.2 Longitudinal Biomarker Performance by longQDA

As the exploratory analysis showed how the initial biomarker levels vary across patients, the mixed models were confined to those with at least a random intercept. Therefore, RI, RIS and RICAR1 are considered as mixed model structures (MMS). For simplicity, the same MMS is always applied for both groups. Time was included as a fixed and random linear effect in the mixed model and an additional quadratic fixed time effect for longQDA analyses involving at least four visits. When estimated the model without the fixed quadratic time effect, a worse fit resulted. However, the performance of the biomarker did not change remarkably. A restricted maximum likelihood (REML) approach was used to estimate the group-specific covariance parameters. The longitudinal variant of QDA was applied including either the first $g = 3, 4, 5$ or 6 visits.

The results of longQDA with regard to the number of measurements per patient were assessed and the performance of the different mixed model structures (MMS) were compared. The MMS were selected based on the BIC (see (3.1) for the definition) which turned out to be the better approach in our simulation study (presented in Chapter 3). Including the first 3 visits, RI was selected as the best model, whereas the RIS model was the favoured structure for data including 4, 5 or 6 visits. The estimated model parameters of the RIS model based on the complete biomarker data are listed in Table 2.2.

| | $\beta_0$ | $\beta_1$ | $\beta_2$ | $d_{\mathrm{I}}$ | $d_{\mathrm{S}}$ | $d_{\mathrm{IS}}$ | $\sigma^2$ |
|---|---|---|---|---|---|---|---|
| non-resistant patients | 4.5 | -13.2 | 18.7 | 0.46 | 8.15 | 0.32 | 0.32 |
| resistant patients | 4.3 | -7.7 | 13.5 | 0.22 | 4.43 | 0.30 | 0.28 |

Table 2.2: Group-specific estimates based on RIS model for 6 visits

Mean group profiles and biomarker profiles simulated according to the resulting marginal distribution are depicted in Figure 2.3. The simulated profiles resemble the observed ones depicted in Figure 2.4 (ignore colouring) quite well.

The classification performance of the biomarker, dependent on the number of visits included in longQDA, is shown in Table 2.1(b). Nearly all performance measures indicate an increase in performance over time, especially for 5 and 6 measurements. This is consistent with our findings from the analyses based on single visits where the fifth and sixth one performed best. However, the calibration measures are worse when using longer biomarker profiles. In general, the biomarker performance with 5 or 6 measurements is as well only of medium size. This may be due to the subgroup in the non-resistant group who have quite flat profiles similar to the resistant patients. According to their high predicted posterior probabilities for being resistant, this subgroup was identified by longQDA as being rather resistant than non-resistant.

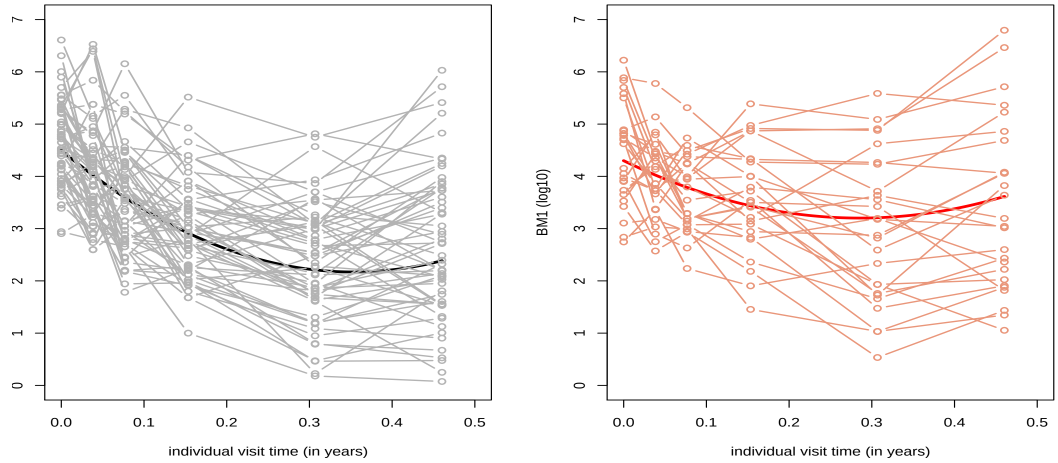


Figure 2.3: Simulated biomarker profiles based on the marginal mixed model parameters for 6 visits (non-resistant patients in black on the left, resistant patients in red on the right). The bold lines are the estimated means.

In Figure 2.4, the biomarker profiles were coloured according to the results gained by longQDA with 6 visits. Overall, the posterior probabilities match well with our previous findings that therapy resistance is characterized by a rather flat profile and non-resistance by a clear decrease in viral load over time. Patients in the resistant group are harder to classify correctly due to their more heterogeneous profiles.

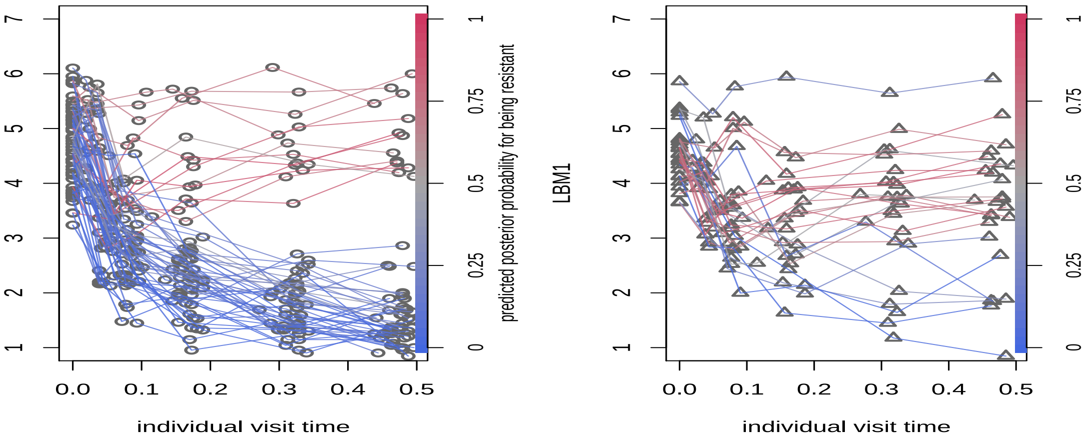


Figure 2.4: Individual observed biomarker profiles coloured according to the predicted posterior probabilities to be resistant, estimated by longQDA with all 6 visits (non-resistant patients on the left, resistant patients on the right).

To put emphasis on the gain in performance by using longitudinal measurements, the increase in classification certainty over time is examined at last and the individual posterior probabilities for being resistant based on 1, 3, 4, 5 and all 6 visits are compared in Figure 2.5. Note that the probability at baseline was only included there as reference point at which the patients were still untreated and should therefore not be used as a prediction for therapy response. Using more and more repeated measurements and waiting some time, the predictions improve: For the "real" non-resistants with a low probability for being resistant, the posterior profiles tend downwards. The resistants, in contrast, exhibit increasing posterior profiles over time, both reflecting an increase in classification certainty. The equivalent estimations of QDA using only the last cross-sectional biomarker measurement (green dots) were included as reference. The posterior probabilities of QDA tend to be more central for the non-resistant patients and lower for the resistant ones. This involves also a lower separation of the probabilities' distributions.

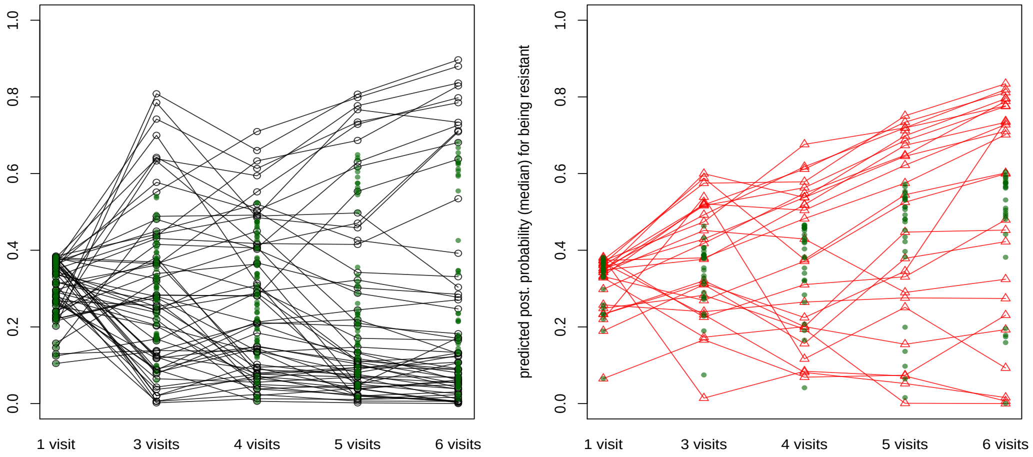


Figure 2.5: Increase in classification certainty with increasing number of measurements (non-resistant patients in black on the left, resistant patients in red on the right). The green dots are the estimated posterior probabilities by QDA when using only the data of the 1st, 3rd, 4th, 5th and 6th visit.

# 3 Model Selection in Longitudinal QDA

## 3.1 Introduction

In this chapter, the importance of selecting an appropriate mixed model structure is examined when estimating the group-specific parameters of the discriminant rule. The model selection is an even more important issue in longQDA than in the case where the sole interest is in the estimation of mixed models: In longQDA, all estimated parameters find their way in the discriminant rule, none of them are nuisance parameters. For simplicity, the issue is elaborated for univariate longQDA. It is assumed that the model selection approach may be transferable to the multivariate case.

In Section 3.2, the effects of fitting various mixed models are examplified with the HIV therapy resistance data of Section 2.3. For real data, the true model structure is unknown. However, in simulations, the true, underlying model structure can be fixed. The detailled setup of the simulation study is described in Section 3.3. In the simulation, two different selection approaches are compared, one where the model was selected based on having the highest performance measures, and another where the selection was based on BIC (Subsection 3.4.1). Within the first approach, all performance measures introduced in Section 2.2 are considered to figure out whether some are more appropriate than others. In Section 3.4.2, the importance of model selection is assessed for longQDA by quantifying the bias of an incorrect model structure. The simulation results are summarized in Section 3.4.3.

## 3.2 Motivation

To demonstrate the effects of different model structures with a practical example and to put emphasis on the model selection issue, the biomarker profiles of the application data set are modelled by RI, RIS, RICAR1 and QDA. Classic QDA ignores the time structure in the data: The order of the visits as well as the actual individual visit times are not taken into account and the covariance matrices are totally unstructured. The abbreviation (M)MS in contrast to MMS refers in the following to model structures which do not necessarily include only mixed models. The estimated performance measures for 5 visits are given in Table 3.1, the estimated posterior probabilities are displayed in Figure 3.1. These results are quite different and depend on the chosen (M)MS, especially at the individual level. So there is a need for a reliable model selection criterion.

| (M)MS | AUC | BS | $\mathrm{BS}_C^{[-]}$ | $\mathrm{BS}_{D_M}$ | $\rho_{z_{[1]},p_{[1]}}$ | $\ln(\mathrm{Q}_{\mathrm{Var}}^{[-]})$ |
|---|---|---|---|---|---|---|
| RI | .74 [.63; .85] | .80 [.70; .87] | .075 [.020; .135] | .08 [.05; .12] | .50 [.32; .65] | 1 [1; 2] |
| RIS | .74 [.62; .84] | .82 [.76; .86] | .024 [.008; .080] | .06 [.03; .09] | .40 [.23; .57] | 3 [2; 4] |
| RICAR1 | .72 [.61; .86] | .81 [.73; .87] | .058 [.017; .135] | .09 [.05; .12] | .45 [.28; .60] | 2 [2; 3] |
| QDA | .68 [.57; .82] | .79 [.73; .85] | .039 [.015; .092] | .05 [.02; .09] | .33 [.28; .56] | 3 [2; 4] |

Table 3.1: Selected performance measures comparing (M)MS of longQDA. They are given as median with 10th and 90th percentile.

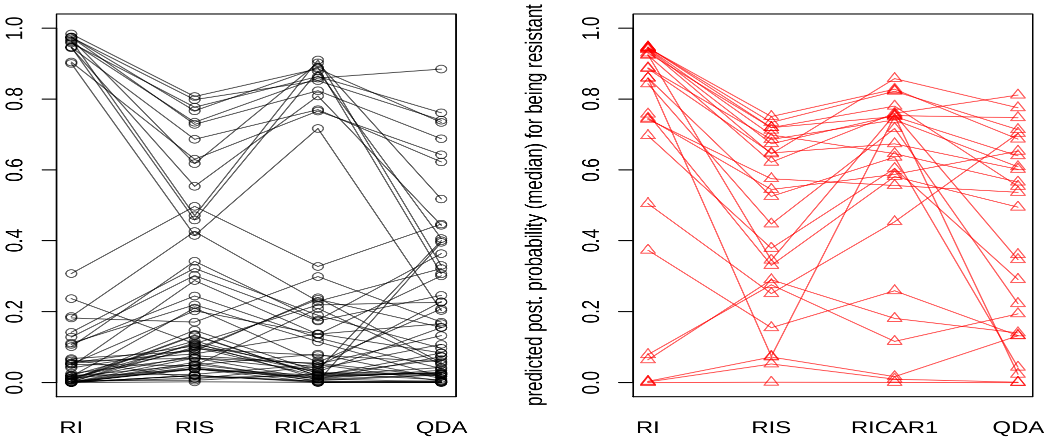


Figure 3.1: Differences between predicted posterior probabilities due to the (M)MS selection in longQDA (non-resistant patients in black on the left, resistant patients in red on the right).

With real data, the "true", underlying model structure is not known. However in simulation studies, this lack can be overcome and the following research questions are addressed:

- Which is the most appropriate approach for choosing the model structure in the estimation step of longQDA?
- How important is the selection of the correct mixed model? How much performance loss occurs by just applying QDA, ignoring the time structure? How large are the effects of an incorrect structure on the results, at the global as well as at the individual level?

For the first issue, two approaches are compared. An heuristic approach would be to fit the longitudinal biomarker data by different mixed models and to select that structure for the two groups which yielded the best classification performance. This approach was chosen by Tomasko et al. (1999) and Wernecke et al. (2004) based on the error rate. Independent of the appropriateness of the approach to use a performance measure as selection criterion, it is recommendable to refrain from the error rate as misclassifications are treated equally for both groups (Pepe, 2003). Another approach would be to select the model with the best fit according to the BIC in the estimation step of longQDA. The use of the BIC to select the best covariance structure of various linear mixed models is common practice and either favoured over AIC (Littell et al., 2000) or leads to the same decisions (Gurka, 2006). Two things are noteworthy when using BIC under REML estimation as we did. First, BIC comparisons are only allowed for models with the same fixed effects (Weiss, 2005, p.18). This is the case for all our models. Second, there is some controversy regarding the correct form of the BIC (Gurka, 2006; Greven, 2007). There are concerns regarding the sample size ($N - u$ in the penalizing term for REML instead of $N$ for ML in the case of $u$ fixed effects) and the number of parameters ($O$ for REML or $u + O$ for ML). Consequently, the computation of the BIC differs from software to software. However, whenever only models with the same fixed effects are compared within the same software, the absolute value of the BIC does not matter. We use the BIC as implemented in the R packages `nlme` and `lme4`:

$$\mathrm{BIC} = -2 \cdot l_{REML} + (u + O) \cdot log(N - u), \tag{3.1}$$

where $l_{REML}$ is the restricted log-likelihood and $O$ is the number of unique parameters in the covariance matrix $\boldsymbol{V}_i$.

The advantage of the BIC approach is the separation of the model selection from the actual goal of the analysis, the classification. Further on, this decision is based on different data: The estimation step is performed on the training – and the classification on the test sets. The importance of choosing the correct mixed model and not applying QDA is a second issue and assessed by the differences between the performance measures at a global level and between the estimated posterior probabilities at an individual level.

## 3.3 Simulation Setup

As the findings might depend on the number of visits and the assumed model structure, data with 3, 6 and 10 visits were simulated according to a RI, RIS or a RICAR1 structure. These "true" covariance structures are denoted by *, i.e. there are RI*, RIS* and RICAR1*. The structures were again assumed to be the same in both patient groups. The biomarker measurements were simulated according to the implied marginal densities of the mixed models with the following parameters:

**RI***

Non-resistant:
$\boldsymbol{\beta}_{[0]} = (4.0, -4.4)^T$, $\boldsymbol{D}_{[0]} = 0.79, \sigma^2_{[0]} = 0.77$

Resistant:
$\boldsymbol{\beta}_{[1]} = (4.0, -1.5)^T$, $\boldsymbol{D}_{[1]} = 0.55, \sigma^2_{[1]} = 0.51$

**RIS***

Non-resistant:
$\boldsymbol{\beta}_{[0]} = (4.0, -4.4)^T$, $\boldsymbol{D}'_{[0]} = \begin{pmatrix} d^2_{\mathrm{I}[0]} & \rho_{\mathrm{IS}[0]} \\ \rho_{\mathrm{IS}[0]} & d'^2_{\mathrm{S}[0]} = d^2_{\mathrm{S}[1]}/d^2_{\mathrm{I}[1]} \end{pmatrix} = \begin{pmatrix} 0.39 & 0.35 \\ 0.35 & 17.1 \end{pmatrix}$,
$\sigma^2_{[0]} = 0.55$

Resistant:
$\boldsymbol{\beta}_{[1]} = (4.0, -1.4)^T$, $\boldsymbol{D}'_{[1]} = \begin{pmatrix} 0.25 & 0.47 \\ 0.47 & 16.3 \end{pmatrix}$, $\sigma^2_{[1]} = 0.39$

(Note the special notation of $\boldsymbol{D}'$. It contains the variance of the intercept $d^2_{\mathrm{I}}$ as usual but the slope's variance standardized by the variance of the intercept in the diagonal and the correlation $\rho_{\mathrm{IS}}$ in the off-diagonal. Thus the assumed correlation between the random intercept and the random slope may directly be read off and compared to the variances on the same scale.)

**RICAR1***

Non-resistant:
$\boldsymbol{\beta}_{[0]} = (4.2, -4.8)^T$, $\boldsymbol{D}_{[0]} = 0.58, \sigma^2_{[0]} = 1.24, \phi_{[0]} = 0.0008$

Resistant:
$\boldsymbol{\beta}_{[1]} = (4.1, -1.6)^T$, $\boldsymbol{D}_{[1]} = 0.47, \sigma^2_{[1]} = 0.69, \phi_{[1]} = 0.0001$

These are the estimates of the HIV application data set when using 50-fold MCCV. The data set comprised all complete cases with 6 visits.

The models under consideration for the simulated data were RI, RIS, RICAR1 and QDA. The models were always fitted with the same structure for both groups. For the mixed models, a linear fixed time effect as given by the true model was part of the model and an unstructured mean profile was fitted for QDA as usual. The parameters of the "true" models are fixed by the simulation design and were not estimated. The classification rule, however, had to be evaluated in the test sets for the "true" as well as for the examined models to assess the biomarker performance. The best performance is expected for the MMS matching the underlying "true" model. Our approach provided the induced deviations when estimating the mean vector and the covariance matrix with the correct resp. the incorrect model structures.

To explore the effect of an incorrect model selection with increasing number of visits, biomarker data measured at 10 visits were generated and the performance was assessed using all visits as well as only the first 3 and the first 6 by deleting the subsequent visits. The individual times of the visits were simulated according to the empirical ones of the HIV data. The empirical study times between visits for patient $i, i = 1, \ldots, N$ were approximated by the following distributions (after mapping them to [0; 1]):

$$\begin{aligned}
\Delta_{t_{i1},t_{i2}} &\sim 0.12 \cdot N(\mu_1 = 0.05, \sigma_1^2 = 0.001) + 0.76 \cdot N(0.5, 0.05) \\
&\quad +0.12 \cdot N(0.95, 0.001) \\
\Delta_{t_{i2},t_{i3}} &\sim N(0.25, 0.006) \\
\Delta_{t_{i3},t_{i4}} &\sim 0.7 \cdot N(0.45, 0.004) + 0.3 \cdot N(0.95, 0.004) \\
\Delta_{t_{i4},t_{i5}} &\sim N(0.45, 0.004) \\
\text{and } \Delta_{t_{ij},t_{i(j+1)}} &\sim N(0.45, 0.0045) \text{ for visit number } j = 5, \ldots, 9.
\end{aligned}$$

The last time difference (between the 5th and the 6th visit) served also for simulating the individual visit times from visit 7 up to visit 10.

The resulting biomarker trajectories are depicted in Figure 3.2. The profiles of the non-resistant group decrease continuously over time whereas the median biomarker level of the resistant group remain quite stable.

Taken together, this resulted in nine simulation scenarios given by the combination of RI*, RIS* or RICAR1* with 3, 6 or 10 visits where the effect of choosing an RI, RIS, RICAR1 or QDA structure was studied. A 50-fold MCCV was applied in the simulation study and patients were randomized per group either to the training or the test set with a ratio of 2:1. The prevalence was fixed to 0.18 for the resistant group as in the application and the group sample size was tripled to get stable results in terms of sample sizes. As a result, our simulation sample consists of 177 non-resistant and 78 resistant patients. The results are based on 25 simulation repetitions and an increase to 50 repetitions yielded just the same results up to small numerical differences.

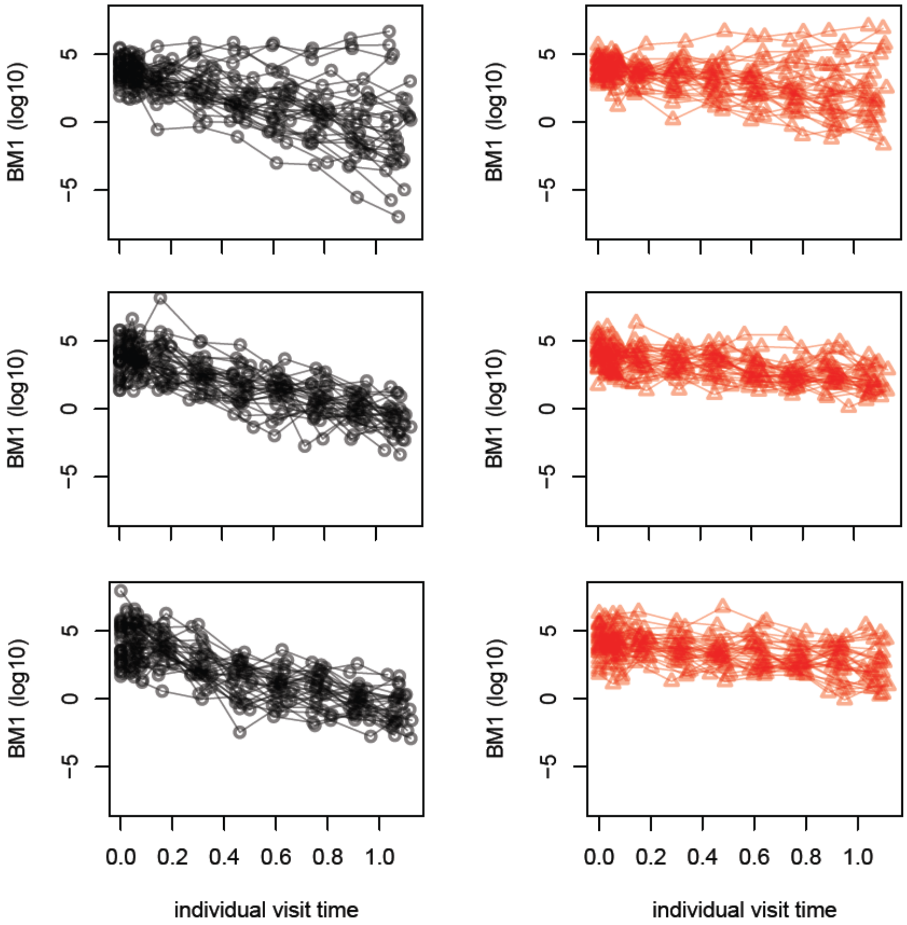


Figure 3.2: Simulated biomarker profiles according to RIS* (top), RI* (middle) and RICAR1* (bottom) with 10 individual visit times. The sample consists of 30 randomly selected non-resistant patients on the left and 30 randomly selected resistant patients on the right.

## 3.4 Simulation Results

Note for the interpretation of our findings in the following that the results of RI*, RIS* and RICAR1* should not be compared: The simulated data led to different performance, e.g. due to the higher variability and its increase over time in any RIS model compared to RI and RICAR1 structures. With 6 visits, for example, RI* yielded a Brier score of 0.866 [0.841; 0.891], RIS* of 0.827 [0.808; 0.845] and RICAR1* of 0.851 [0.827; 0.868] (given as median with 10th and 90th percentile over all MCCV samples). When biomarker profiles were simulated according to RI* or RICAR1*, the performance measures improved remarkably over time - especially for RI* - due to less variation between the profiles over time. This yielded a very good performance for 10 measurements. In general, the effect of assuming a wrong (M)MS is less an issue in the case of well separated groups. Then, it is more unlikely that patients are misclassified just due to biased and therefore possibly more similar model parameter estimates between the groups.

Expected convergence problems occurred for RI* when fitting models with too complex structures like RIS and RICAR1. For RIS, this was only the case for some of the MCCV samples, whereas for RICAR1 no convergence was achieved at all. Hence the considered incorrect model structures are confined to RIS and QDA for RI*. For RIS*, the simulation scenario with 3 visits was excluded from the assessment for model selection as the longitudinal structure of the simulated RIS* models, represented by the random linear slopes, could not be captured based on 3 measurements. This was reflected by the estimated mixed model parameters of RIS, which differed clearly from the true ones of RIS* resp. by the estimated performance measures of RIS, which did not match up with those of RIS*. There were also convergence problems: In about a fourth of the comparisons, the RICAR1 models did not converge for at least one of the groups. For all other simulation scenarios, the MMS matching to the one underlying the simulated biomarker data achieved the same performance as the corresponding MMS*. This confirmed that the estimation of the mixed model parameters went well, without any critical information loss with regard to the classification.

### 3.4.1 Criterion for Model Selection: Highest Performance Measure or BIC?

First, a heuristic approach was applied, selecting the model which yielded the best performance measure. As it was unclear which performance measure is the most suitable for this purpose, we assessed as well whether all performance measures presented in Section 2.2 support the correct model selection. The results for 10 visits are displayed in Table 3.2.

For RI* with 10 visits, the classification performance was very good, resulting in small differences between the (M)MS. For this scenario, all examined performance measures did not uniquely point to the correct model: RI and RIS achieved the same results as RI*, only the measures of QDA correctly indicated a worse performance. Using the first 6 visits, where the classification is worse than for 10 visits, yielded the same indistinguishable results. For three visits, the performance measures of all (M)MS were practically the same and therefore of no use for model selection purposes.

For RIS* with 10 biomarker measurements, the performance was uniquely best for the RIS model according to AUC, BS, $\text{BS}_C$, $\text{BS}_{D_M}$ and the correlation $\rho_{z_{[1]},p_{[1]}}$. The calibration component $\text{Cal}_L$ was not helpful in that the QDA structure gave similar small values as the RIS model. Absolutely inadequate as selection criteria were the performance measures $\text{Diff}_{\bar{p}_{[1]}} = \bar{p}_{[1][z_{[1]}=1]} - \bar{p}_{[1][z_{[1]}=0]}$ and $\text{Q}_{\text{Var}}$: They indicated that RI and RICAR1 are the correct covariance structures instead of RIS. The same applies to RIS* with 6 visits. The

| (M)MS | AUC | BS | $\text{BS}_C^{[-]}$ | $\text{BS}_{D_M}$ | $\text{Diff}_{\bar{p}_{[1]}}$ | $\rho_{z_{[1]},p_{[1]}}$ | $\text{Q}_{\text{Var}}^{[-]}$ | $\text{Cal}_L^{[-]}$ |
|---|---|---|---|---|---|---|---|---|
| RI* | 1.00 [1.00; 1.00] | .99 [.98; 1.00] | .002 [.000; .008] | .15 [ .13; .15] | .96 [.91; .98] | .81 [.79; .83] | 1 [0; 1] | .01 [.01; .02] |
| RI | 1.00 [1.00; 1.00] | .99 [.98; 1.00] | .003 [.000; .009] | .15 [ .13; .15] | .95 [.91; .98] | .81 [.79; .83] | 1 [0; 1] | .01 [.01; .02] |
| RIS | 1.00 [1.00; 1.00] | .99 [.98; 1.00] | .003 [.000; .009] | .15 [ .12; .15] | .95 [.90; .98] | .81 [.79; .83] | 1 [0; 1] | .01 [.01; .02] |
| QDA | 1.00 [ .98; 1.00] | .98 [.96; 1.00] | .005 [.001; .012] | .14 [ .11; .15] | .92 [.85; .97] | .79 [.75; .82] | 1 [1; 1] | .01 [.01; .02] |
| RIS* | .87 [ .82; .91] | .85 [.83; .88] | .017 [.009; .029] | .01 [−.01; .04] | .30 [.24; .36] | .50 [.42; .58] | 12 [8; 19] | .00 [.00; .00] |
| RIS | .87 [ .82; .91] | .85 [.83; .87] | .018 [.006; .032] | .01 [−.01; .04] | .30 [.23; .37] | .49 [.40; .57] | 12 [8; 20] | .00 [.00; .01] |
| RICAR1 | .82 [ .76; .88] | .82 [.78; .86] | .034 [.015; .060] | .00 [−.02; .03] | .43 [.33; .53] | .43 [.33; .52] | 7 [4; 13] | .03 [.01; .05] |
| RI | .81 [ .74; .86] | .78 [.73; .83] | .069 [.043; .106] | −.01 [−.03; .02] | .49 [.38; .60] | .42 [.42; .51] | 6 [3; 10] | .05 [.03; .10] |
| QDA | .79 [ .71; .85] | .81 [.78; .85] | .027 [.015; .046] | −.02 [−.04; .01] | .24 [.15; .34] | .37 [.25; .48] | 17 [9; 41] | .00 [.00; .00] |
| RICAR1* | 1.00 [1.00; 1.00] | .97 [.96; .99] | .007 [.002; .014] | .13 [ .11; .14] | .85 [.79; .91] | .78 [.74; .80] | 1 [1; 1] | .01 [.01; .02] |
| RICAR1 | 1.00 [ .99; 1.00] | .97 [.96; .99] | .007 [.002; .016] | .13 [ .11; .14] | .85 [.79; .91] | .77 [.74; .80] | 1 [1; 1] | .01 [.01; .02] |
| RIS | .99 [ .98; 1.00] | .97 [.95; .99] | .007 [.002; .015] | .12 [ .10; .14] | .84 [.77; .91] | .77 [.73; .80] | 1 [1; 1] | .01 [.01; .02] |
| RI | .99 [ .97; 1.00] | .96 [.94; .98] | .007 [.002; .015] | .11 [ .09; .14] | .88 [.81; .94] | .75 [.70; .80] | 1 [1; 1] | .02 [.01; .03] |
| QDA | .98 [ .95; 1.00] | .95 [.93; .97] | .010 [.004; .020] | .11 [ .08; .13] | .79 [.68; .87] | .74 [.68; .78] | 1 [1; 2] | .01 [.00; .01] |

Table 3.2: Performance measures with different (M)MS of longQDA with 10 visits, given as median with 10th and 90th percentile. Data were simulated according to RI*, RIS* or RICAR1*.

other performance measures such as AUC etc. , which were adequate in the case of 10 visits, exhibited another problem: The measures of all (M)MS were so similar in size that there is no single best one.

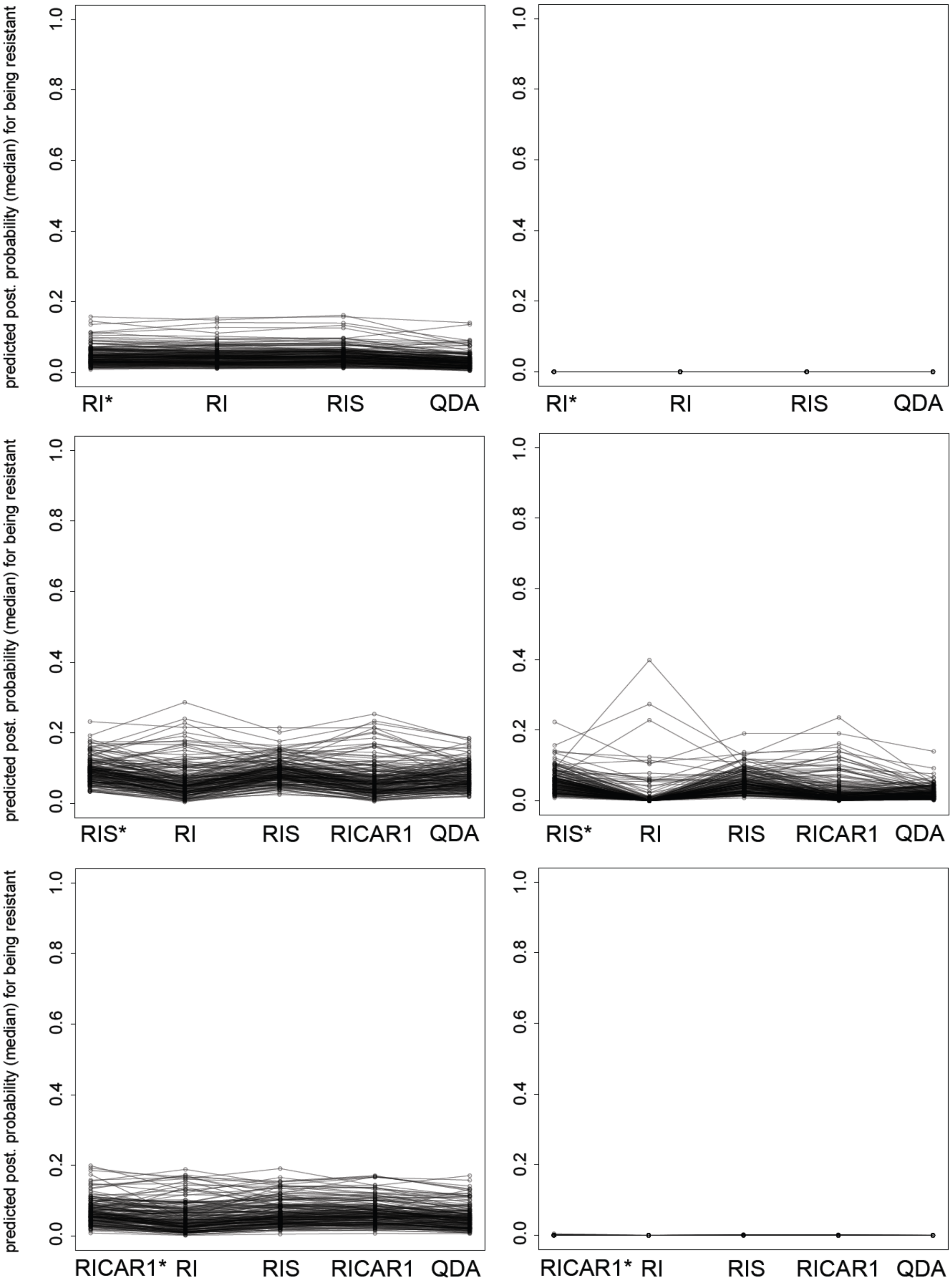
For RICAR1* with 10 biomarker measurements, AUC, BS, $\mathrm{BS}_{D_M}$ and the correlation $\rho_{z_{[1]},p_{[1]}}$ indicated correctly the highest performance for RICAR1. For 6 measurements, this was only the case for AUC and $\rho_{z_{[1]},p_{[1]}}$. Inadequate as selection criteria are $\mathrm{Diff}_{\bar{p}_{[1]}}$ for 6 and 10 visits and $Q_{\mathrm{Var}}$ for 6 visits suggesting RI as best model as well as $\mathrm{Cal}_L$ for 10 visits with QDA as best one. No unique optimal model due to similar performance may be selected based on $\mathrm{BS}_C$, $\mathrm{Q}_{\mathrm{Var}}$ and $\mathrm{Cal}_L$ for 6 and 10 visits, and additionally by $\mathrm{BS}_{D_M}$ for 6 visits. For biomarker profiles of length 3, all performance measures exhibited this problem of too similar sizes.

What about the BIC? There is strong evidence for the approach of selecting the model with the minimal BIC to work. For all assumed MMS* with 10 visits, the model with the same covariance structure as MMS* yielded the minimal BIC in all simulation repetitions, all MCCV samples and both patient groups. For RI*, RIS* and RICAR1* with 6 visits and for RI* with 3 visits, the selection was also successful in nearly all comparisons between the fitted MMS. For RICAR1* with 3 visits, the BIC of RICAR1 was at least in the majority (44%) of all comparisons the minimal one.

### 3.4.2 Global and Individual Effects of an Incorrect (M)MS

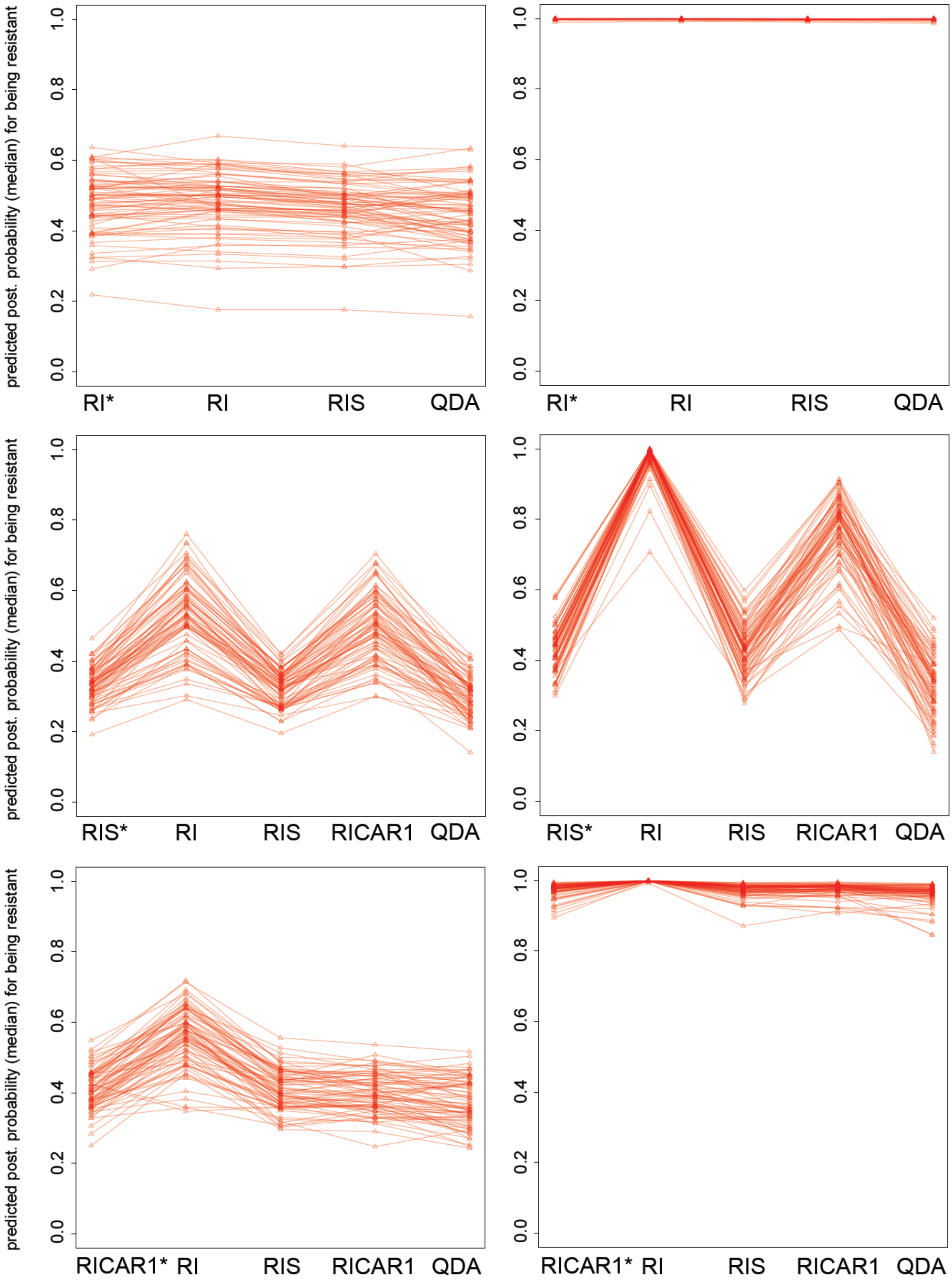
The effects of an incorrect (M)MS are assessed at the global as well as at the individual level. The global level is assessed in terms of the performance measures. All of them are based on the predicted posterior probability for each patient to be resistant to the therapy. The individual level is directly assessed by these posterior probabilities. This is obviously a stricter assessment than at the global level but it gives a more detailed picture about the implications at the patient level.

For biomarker profiles of length 3, no effects of using various model structures were observed for RI*, RIS* or RICAR1*, neither at the global nor at the individual level. Thus the assessment was restricted to 6 and 10 visits. At the global level, considerable differences between the performance measures were only observed for 10 visits under RIS* (Table 3.2, second block). Incorrect model structures are characterized by worse median performance measures but also by high variances over all MCCV samples. For the BS, for example, the maximum deviation was observed for RI where the median score was 0.07 smaller and the confidence interval did not overlap with that of RIS*. The smallest median AUC resulted with QDA, it is 0.79 instead of 0.87. For 10 visits under RI* or RICAR1*, the groups are so well separated that an incorrect estimation of the model parameters was not an issue.



(a) Non-resistant patients

Figure 3.3: Profiles of individual predicted posterior probabilities comparing the (M)MS RI, RIS, RICAR1 and QDA. The true MMS was RI* in the first row, RIS* in the second and RICAR1* in the third row. The profiles were based on data from 6 visits in the left column of each subfigure and on data from 10 visits in the right column.



(b) Resistant patients

For 6 visits under RI*, RIS* or RICAR1*, the differences in the discrimination rule induced by the (M)MS were probably too small to have an effect on the performance measures.

Further on, the effects were assessed at the individual level. In Figure 3.4(a), the median posterior probabilities of all MCCV samples that were predicted for one therapy-resistant patient under each assumed model are represented by circles on one line. The first mark of each line gives the posterior probability estimated by the model with the true covariance structure.

The deviations from this benchmark were much more pronounced for the resistant than for the non-resistant group (cf. Figure 3.4(a) and 3.3(a)). For RI* and 6 visits, QDA tended to predict lower posterior probabilities to be resistant, which are at most 0.1 smaller in size. For 10 visits, the very good classification performance of RI* did not lead to any differences between the (M)MS. Under RIS*, the highest deviations occured especially for RI and and also for RICAR1. They were of unacceptable size for therapy-resistant patients with 6 resp. 10 biomarker measurements: The probabilities of the RI models were about 0.4 resp. 0.5 higher than predicted under RIS, and those of the RICAR1 models deviated about 0.4 for both profile lengths. For both groups, RI and RICAR1 overestimated the probabilities to be part of the same group as according to the Gold Standard. QDA overestimated these probabilities for the non-resistant patients but underestimated them for the resistant ones. But this was different for 6 visits where the incorrect structure QDA resulted hardly in any bias. Spuriously high predicted probabilities to be resistant to the therapy, like e.g. under RI and RICAR1, implied a falsely low uncertainty in classification whereas spuriously low probabilities, like e.g. for QDA with 10 visits, reflected a falsely high uncertainty. Both are undesirable effects. Assuming an RICAR1* structure, the implications for the patients were at most critical when fitting RI models. The posterior probabilities of the resistants were incorrectly higher – similar in size as under RIS*. High deviations for single patients were due to unstable estimation, they disappeared by increasing the number of simulation repetitions.

### 3.4.3 Summary

As the simulations showed, an incorrect model structure had at first effects at the individual level resulting in incorrectly low or high uncertainty in classification. The QDA structure was always one of the incorrect models which was identified in the simulations. There was no effect when the group separation in the scenario was very good. Effects at the global level were only observed for the most complex structure RIS* with longer profiles of length 10. This means that the selection of the correct structure was especially important for models that assumed a time-variant longitudinal data structure and the deviations in the results enlarged with increasing length of the profiles. Therefore, it is important to have an objective model selection criterion. The BIC approach was found to perform much better than the performance measures. The decision for a specific model structure should not be based on the model with the best performance measure as they were too similar in size for most scenarios (especially those with 3 and 6 visits). Additionally, if they differed, only few of them as AUC, BS and $\rho_{z_{[1]},p_{[1]}}$ were able to select the correct model. These findings suggest to use the BIC as model selection criterion as already done in the application in Section 2.3. At the beginning of this section, the influence of the assumed model structure on the estimations were shown for the HIV therapy resistance data. The direction of the bias was similar in the simulation study (cf. Figure 3.1 on the right with the left plot in the middle of Figure 3.4(a) for resistant patients and Figure 3.1 on the left with the left plot in the middle of Figure 3.3(a) for non-resistant patients.).

In the multivariate case when the performance of biomarker combinations are assessed, the model selection process is even more important. The selection of an incorrect model structure does not only result in a wrong assessment of a biomarker combination but also influences the search for the best biomarker combination. Possible multivariate extensions for the longitudinal QDA are the topic of the next chapter.

**Remark**

This chapter is based on Kohlmann, M., Held, L., Grunert, V.P. (2009). Classification of Therapy Resistance Based on Longitudinal Biomarker Profiles. *Biometrical Journal* 51, 610 – 626.

# 4 Multivariate Extensions of Longitudinal QDA for Biomarker Combinations

## 4.1 Introduction

In Chapter 2, statistical methods were proposed to assess the diagnostic value of single longitudinal biomarkers. In this chapter, longQDA is extended to the multivariate case to assess biomarker combinations. This opens up further opportunities: In addition to the potential to improve the classifier by using repeated measurements like in univariate longQDA, the combination of biomarkers may also yield to a better performance.

Multivariate longitudinal data are sometimes referred to as doubly multivariate data (Timm, 2002) as they exhibit two multivariate features, one is the longitudinal nature, the repeated measurements over time, the other the set of variables, the biomarkers. From a diagnostic point of view, the two features complement one another well. The longitudinal measurements contain information about the development of disease processes whereas biomarkers may be qualitatively selected to cover various aspects of the disease. The biomarkers are, for example, characteristic for the inflammation process or for the degradation process triggered by the disease. Combining biomarkers in one diagnostic test is advantageous for the patient as only one blood sample is required to determine the level of several biomarkers. Moreover, with the technical development of multi-test diagnostic platforms about the same amount of blood serum is required for the measurement of one biomarker as for the simultaneous measurement of a biomarker panel.

So statistical models need to be established to allow the assessment of a biomarker panel with longitudinal profiles. For multivariate longQDA, the key point is the estimation of the multivariate group-specific means and covariance matrices. Multivariate linear mixed models are considered for the plug-in estimation and three different classes of model structures are examined. One is characterized by the independence assumption for the biomarkers (Section 4.2.1), the other two are appropriate for correlated biomarkers: multivariate random effects models (Section 4.2.2) and multivariate covariance pattern models (Section 4.2.3). Having established appropriate models, the subsequent proceeding is the same as in the univariate case. For model selection, we rely on the simulation results showing the appropriateness of the BIC and use it as selection criteria for multivariate models, too. In Section 4.3, the established multivariate longQDA is applied to biomarker data which are examined for their ability to predict drug resistance for patients with rheumatoid arthritis (RA). The chapter ends with exemplary simulated data settings where the classification performance of biomarker panels exceeds the performance of their single components (Section 4.4).

## 4.2 Multivariate Linear Mixed Models

We have measurements $\boldsymbol{w}_i = \left(\boldsymbol{w}_i^{[1]T}, \ldots, \boldsymbol{w}_i^{[q]T}\right)^T$ of $\ell = 1, \ldots, q$ biomarkers, collected at $t_{ij}, j = 1, \ldots, p$ time points from $i = 1, \ldots, n_{[k]}$ patients in each patient group $k = 0, 1$. In applications from medical diagnostics, all biomarker levels are usually determined from the same blood sample taken at one time. So for each patient, the time points are identical for all biomarkers. As above, the subscripts $k = 0, 1$ indicating that the estimation is performed group-wise are omitted in the following.

The multivariate mixed model is defined as

$$\boldsymbol{w}_{i(pq\times 1)} = \boldsymbol{X}_{i(pq\times uq)}\boldsymbol{\beta}_{(uq\times 1)} + \boldsymbol{Z}_{i(pq\times sq)}\boldsymbol{b}_{i(sq\times 1)} + \boldsymbol{\epsilon}_{i(pq\times 1)}$$

where

$$\begin{aligned} \boldsymbol{b}_i &\sim N(\boldsymbol{0}_{(sq\times 1)}, \boldsymbol{D}_{(sq\times sq)}) \\ \boldsymbol{\epsilon}_i &\sim N(\boldsymbol{0}_{(pq\times 1)}, \boldsymbol{R}_{i(pq\times pq)}). \end{aligned}$$

Due to the fact that all measurements are determined from one blood sample, the matrix $\boldsymbol{Z}_i$ (and also the time-related columns of $\boldsymbol{X}_i$) consists of $q$ submatrices on the diagonal and each submatrix equals the design matrix of the univariate model (see p. 6 for the definition).

So the biomarker measurements of one patient are distributed as

$$\boldsymbol{w}_i \sim N(\boldsymbol{\mu}_i = \boldsymbol{X}_i\boldsymbol{\beta}, \boldsymbol{V}_i = \boldsymbol{Z}_i^T\boldsymbol{D}\boldsymbol{Z}_i + \boldsymbol{R}_i).$$

The $n_{[K]}$ patient-specific regression models are then combined as described for the univariate case in Eq. (2.2).

The models can also be expressed exclusively in terms of matrices. Then the response matrix has $q$ columns: $\boldsymbol{W}_i = \left(\boldsymbol{w}_i^{[1]T}, \ldots, \boldsymbol{w}_i^{[q]T}\right)$. The matrices of fixed and random effects as well as the residuals' error matrix have $q$ columns as well. The matrices $\boldsymbol{X}_i$, $\boldsymbol{Z}_i$, $\boldsymbol{D}$ and $\boldsymbol{R}_i$ are the same as in the vector notation. This implies, however, a less flexible structure, requiring for all biomarkers the same dimension of fixed and random effects.

The multivariate modeling for biomarker profiles is of higher complexity compared to the univariate one due to the increased number of covariance parameters in $\boldsymbol{V}_i$, in particular due to the correlations. Aside from the autocorrelations between the measurements of one biomarker being referred to as the within correlation of the biomarkers, the between correlation of the biomarkers need to be considered. These correlations between the biomarkers (cross-correlations) are time-dependent, too. We distinguish two forms: one between the biomarkers measured at the same time point and the other between measurements of (distinct) biomarkers measured at different time points. The latter is a particular feature for multivariate longitudinal data, the first is well-known from the analysis of cross-sectional multivariate data.

Suppose $q = 2$ biomarkers were measured at $p = 3$ time points. The corresponding covariance matrix $\boldsymbol{V}_i$ is

$$\boldsymbol{V}_{i(6\times 6)} = \left(\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{E} \end{array}\right) = \left(\begin{array}{ccc|ccc} v_{11}^{[1]2} & v_{21}^{[1]} & v_{31}^{[1]} & v_{11}^{[1,2]} & v_{21}^{[1,2]} & v_{31}^{[1,2]} \\ v_{21}^{[1]} & v_{22}^{[1]2} & v_{32}^{[1]} & v_{12}^{[1,2]} & v_{22}^{[1,2]} & v_{32}^{[1,2]} \\ v_{31}^{[1]} & v_{32}^{[1]} & v_{33}^{[1]2} & v_{13}^{[1,2]} & v_{23}^{[1,2]} & v_{33}^{[1,2]} \\ \hline v_{11}^{[1,2]} & v_{12}^{[1,2]} & v_{13}^{[1,2]} & v_{11}^{[2]2} & v_{21}^{[2]} & v_{31}^{[2]} \\ v_{21}^{[1,2]} & v_{22}^{[1,2]} & v_{23}^{[1,2]} & v_{21}^{[2]} & v_{22}^{[2]2} & v_{32}^{[2]} \\ v_{31}^{[1,2]} & v_{32}^{[1,2]} & v_{33}^{[1,2]} & v_{31}^{[2]} & v_{32}^{[2]} & v_{33}^{[2]2} \end{array}\right), \tag{4.1}$$

where the unique parameters to be estimated are printed in blue. On the block-diagonal are the covariance matrices of each biomarker, namely $\boldsymbol{A}$ and $\boldsymbol{E}$. They contain the variances on the diagonal and the symmetric autocovariances on the off-diagonal. The off-diagonal matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are the covariance matrices containing the within-time cross-correlations on their diagonal and the between-time cross-correlations on their off-diagonal. The diagonals of $\boldsymbol{B}$ and $\boldsymbol{C}$ are equal and the lower triangular matrix of $\boldsymbol{C}$ equals the upper triangular matrix of $\boldsymbol{B}$. Note that the matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are not symmetric, caused by the asymmetric property of the between-time cross-correlations. In other words, biomarker $\boldsymbol{w}_i^{[1]}$ measured at time point $t_{ij}$ is differently related to biomarker $\boldsymbol{w}_i^{[2]}$ measured at time $t_{ij'}$ than biomarker $\boldsymbol{w}_i^{[1]}$ measured at time $t_{ij'}$ is related to biomarker $\boldsymbol{w}_i^{[2]}$ measured at time $t_{ij}$.

Due to the large number of parameters, parsimonious multivariate extensions of mixed models are needed that are effective when the number of observations is not large enough to estimate an unstructured covariance matrix. Furthermore, the possible problem of instability resulting from overparameterising the covariance matrix may be circumvented if a useful representation of the underlying correlation structures is found.

There is the usual distinction between random effects models and covariance pattern models. For the first, the random effects $\boldsymbol{b}_i$ are assumed to capture most of the variation within the patient and between the biomarkers and a simple covariance structure as $\boldsymbol{R}_i = \boldsymbol{\Sigma}_{(q\times q)} \otimes \boldsymbol{I}_{(p\times p)}$ is assumed for the residuals where $\boldsymbol{\Sigma}$ is an unstructured matrix. In a pure covariance pattern model, there are no random effects $\boldsymbol{b}_i$. The dependencies in the data are modeled by a more complex structure for $\boldsymbol{R}_i$ than in the random effects model and thus $\boldsymbol{V}_i = \boldsymbol{R}_i$. Parsimonious multivariate random effects models as well as covariance pattern models are presented in the following.

But first, the differences between those two model classes are depicted with respect to the auto- and cross-correlations. In Figure 4.1, one arrow represents one correlation parameter, assuming that no restrictions are placed on the models.

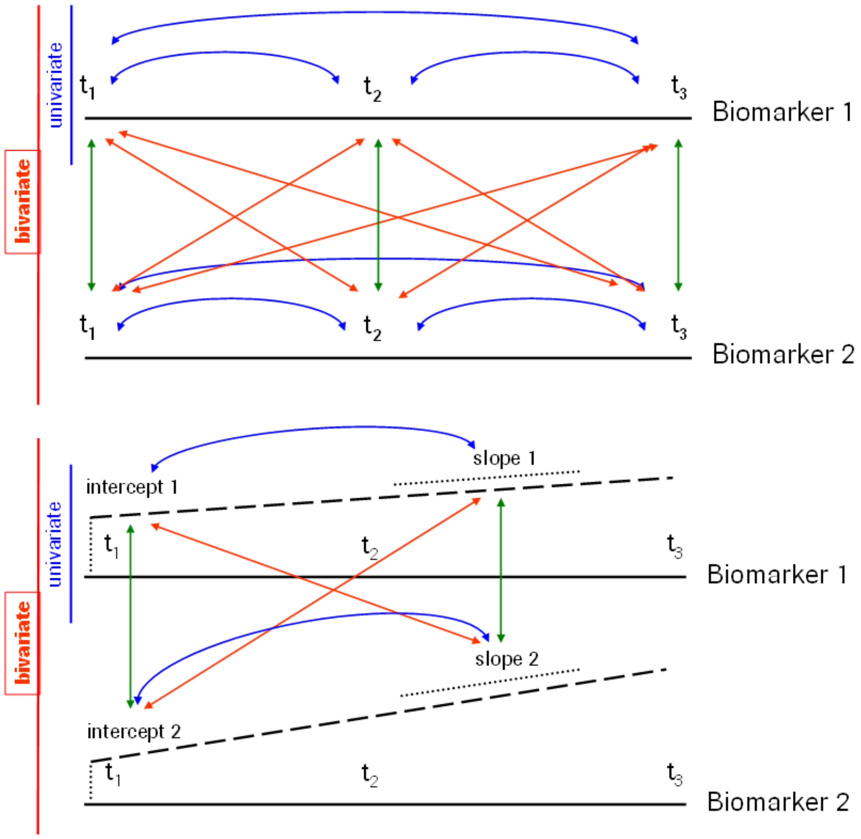


Figure 4.1: Correlation scheme of bivariate mixed models. On the top: covariance pattern model, on the bottom: random effects model. Blue arrows symbolize autocorrelations, green arrows within-time cross-correlations and red arrows between-time cross-correlations.

The implications of the change from the univariate to the multivariate model are clear: The number of correlations increase as the blue ones are extended by the red and green types. The random effects models (below) induce a more parsimonious structure than the covariance pattern models (above). The difference between the number of parameters increases with $q$ and is thus more pronounced in the multivariate case. It will be investigated in Subsection 4.2.3 that strict assumptions are required for covariance pattern models to ensure a reasonable relation between the number of parameters and the sample size and hence to enable the estimability of the models. The following presentation is restricted to the bivariate case considering $q = 2$ biomarkers without loss of generality. Models for $q > 2$ biomarkers require an extended approach and are therefore shortly discussed in Chapter 6.

### 4.2.1 Models with Uncorrelated Random Effects

The independence assumption of the biomarkers leads us back to the univariate case, only the blue arrows of both schemes in Figure 4.1 are modelled. Whether the autocorrelations of the biomarkers for longQDA are modelled only by random effects (RIind or RISind) or additionally by time-dependent residuals (RICAR1ind), the approach is the same: It is sufficient to model each biomarker separately and then the results are combined as follows to assess the combined biomarker performance. Treating the $q = 2$ biomarkers $\boldsymbol{w}^{[1]}$, $\boldsymbol{w}^{[2]}$ as independent, the mean and the covariance matrices are estimated separately as the joint density can be factorized as

$$P(k = K|\, \boldsymbol{w}^{[1]}, \boldsymbol{w}^{[2]}) \propto \pi_{[K]} \cdot f(\boldsymbol{w}^{[1]}|\, k = K) \cdot f(\boldsymbol{w}^{[2]}|\, k = K)$$

for group $K$.

It follows with Equation (2.3) that the corresponding posterior probability $p_{i[K]}^{[1,2]}$ based on 2 biomarkers is

$$\begin{aligned} p_{i[K]}^{[1,2]} &= \frac{\exp\left[-\frac{1}{2}\sum_{\ell=1}^{2}\left((\boldsymbol{w}_i^{[\ell]}-\boldsymbol{\mu}_{[K]}^{[\ell]})^T\ (\boldsymbol{V}_{[K]}^{[\ell]})^{-1}\ (\boldsymbol{w}_i^{[\ell]}-\boldsymbol{\mu}_{[K]}^{[\ell]})\right)\right]\cdot\left[\prod_{\ell=1}^{2}|\boldsymbol{V}_{[K]}^{[\ell]}|\right]^{-\frac{1}{2}}\cdot\pi_{[K]}}{\sum_{k=0}^{1}\exp\left[-\frac{1}{2}\sum_{\ell=1}^{2}\left((\boldsymbol{w}_i^{[\ell]}-\boldsymbol{\mu}_{[k]}^{[\ell]})^T\ (\boldsymbol{V}_{[k]}^{[\ell]})^{-1}\ (\boldsymbol{w}_i^{[\ell]}-\boldsymbol{\mu}_{[k]}^{[\ell]})\right)\right]\cdot\left[\prod_{\ell=1}^{2}|\boldsymbol{V}_{[k]}^{[\ell]}|\right]^{-\frac{1}{2}}\cdot\pi_{[k]}} \\ &= \frac{1}{\pi_{[K]}}\prod_{\ell=1}^{2} p_{i[K]}^{[\ell]}. \end{aligned} \tag{4.2}$$

That is, the results of the univariate mixed models as given by Equation (4.2) are used in that the posterior probabilities of the univariate RI, RIS, RICAR1 model are combined to yield the posterior probabilities of the biomarker pair.

The implied marginal covariance matrix $\boldsymbol{V}_i$ of Equation (4.1) has only non-zero entries in the matrices on the block diagonal, all entries in the off-diagonal are assumed to be zero, including the cross-correlations:

$$\boldsymbol{V}_{i(6\times6)} = \left(\begin{array}{ccc|ccc} v_{11}^{[1]2} & v_{21}^{[1]} & v_{31}^{[1]} & 0 & 0 & 0 \\ v_{21}^{[1]} & v_{22}^{[1]2} & v_{32}^{[1]} & 0 & 0 & 0 \\ v_{31}^{[1]} & v_{32}^{[1]} & v_{33}^{[1]2} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & v_{11}^{[2]2} & v_{21}^{[2]} & v_{31}^{[2]} \\ 0 & 0 & 0 & v_{21}^{[2]} & v_{22}^{[2]2} & v_{32}^{[2]} \\ 0 & 0 & 0 & v_{31}^{[2]} & v_{32}^{[2]} & v_{33}^{[2]2} \end{array}\right).$$

The selection of the assumed univariate mixed model structure is guided by the model with the smallest BIC as examined in Chapter 3. There is a high flexibility with regard to the model structure as it can even differ across biomarkers. This is especially favourable for $q > 2$.

### 4.2.2 Models with Correlated Random Effects

The models presented in this section are direct extensions of the univariate mixed models with an RI or RIS structure (Chapter 2, p. 6) and denoted as RIcorr resp. RIScorr. They are increasingly common in medical applications (Mickey et al., 1994; Shah et al., 1997; Chakraborty et al., 2003; Beckett et al., 2004). The random effects are now assumed to be correlated, resulting in the following covariance matrices $D$. For the RIcorr model, the covariance matrix of the random intercepts is

$$\boldsymbol{D} = \begin{pmatrix} d_{\mathrm{I}}^{[1]2} & d_{\mathrm{II}}^{[1,2]} \\ d_{\mathrm{II}}^{[1,2]} & d_{\mathrm{I}}^{[2]2} \end{pmatrix}.$$

For the RIScorr model, the $(4 \times 4)$-covariance matrix $D$ of the random effects is

$$\boldsymbol{D} = \begin{pmatrix} d_{\mathrm{I}}^{[1]2} & d_{\mathrm{IS}}^{[1]} & d_{\mathrm{II}}^{[1,2]} & d_{\mathrm{IS}}^{[2,1]} \\ d_{\mathrm{IS}}^{[1]} & d_{\mathrm{S}}^{[1]2} & d_{\mathrm{IS}}^{[1,2]} & d_{\mathrm{SS}}^{[1,2]} \\ d_{\mathrm{II}}^{[1,2]} & d_{\mathrm{IS}}^{[1,2]} & d_{\mathrm{I}}^{[2]2} & d_{\mathrm{IS}}^{[2]} \\ d_{\mathrm{IS}}^{[2,1]} & d_{\mathrm{SS}}^{[1,2]} & d_{\mathrm{IS}}^{[2]} & d_{\mathrm{S}}^{[2]2} \end{pmatrix}.$$

The covariances $d_{\mathrm{II}}^{[1,2]}$ and $d_{\mathrm{SS}}^{[1,2]}$ of the random effects allow for the possibility that the average levels of the intercepts resp. the slopes are correlated. Models with constraints between the biomarkers, i.e. where the random intercepts are correlated and the random slopes are not or vice versa, have not been considered.

For the residuals, a simple covariance structure as $\boldsymbol{R}_i = \boldsymbol{\Sigma}_{(q\times q)} \otimes \boldsymbol{I}_{(p\times p)}$ is assumed where $\boldsymbol{\Sigma}$ is an unstructured matrix. The covariance parameter $\sigma^{[1,2]}$ contains the correlation between the residuals of two biomarkers measured at the same point in time. For the validity of this model, both biomarker measurements need to be recorded simultaneously. We assume – also in the application later on – that conditional on the random effects the error components are uncorrelated, thus $\sigma^{[1,2]} = 0$ and $\boldsymbol{R}_i = \mathrm{diag}(\sigma^{[1]2}, \sigma^{[2]2}) \otimes \boldsymbol{I}$. This is a reasonable assumption for $q$ biomarkers which come from different test assays and was also suggested in the literature (Shah et al., 1997).

Assuming $p = 3$ measurements at $(t_{i1}, t_{i2}, t_{i3})$, the overall covariance matrix $\boldsymbol{V}_i$ of the RIScorr model is

$$\begin{aligned}
\boldsymbol{V}_i &= \boldsymbol{Z}_i \boldsymbol{D} \boldsymbol{Z}_i^T + \boldsymbol{R}_i \\
&= \begin{pmatrix} 1 & t_{i1} & 0 & 0 \\ 1 & t_{i2} & 0 & 0 \\ 1 & t_{i3} & 0 & 0 \\ 0 & 0 & 1 & t_{i1} \\ 0 & 0 & 1 & t_{i2} \\ 0 & 0 & 1 & t_{i3} \end{pmatrix} \cdot D \cdot \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ t_{i1} & t_{i2} & t_{i3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & t_{i1} & t_{i2} & t_{i3} \end{pmatrix} \\
&\quad + \begin{pmatrix} \sigma^{[1]2} & 0 & 0 & \sigma^{[1,2]} & 0 & 0 \\ 0 & \sigma^{[1]2} & 0 & 0 & \sigma^{[1,2]} & 0 \\ 0 & 0 & \sigma^{[1]2} & 0 & 0 & \sigma^{[1,2]} \\ \sigma^{[1,2]} & 0 & 0 & \sigma^{[2]2} & 0 & 0 \\ 0 & \sigma^{[1,2]} & 0 & 0 & \sigma^{[2]2} & 0 \\ 0 & 0 & \sigma^{[1,2]} & 0 & 0 & \sigma^{[2]2} \end{pmatrix} .
\end{aligned}$$

The single entries in $\boldsymbol{V}_i$ have a complex structure, e.g. the $(1,1)$-entry is $d_{\mathrm{I}}^{[1]2} + d_{\mathrm{I}}^{[2]2} + 2t_{i1}\left(d_{\mathrm{IS}}^{[1]} + d_{\mathrm{IS}}^{[2]} + d_{\mathrm{II}}^{[1,2]} + d_{\mathrm{IS}}^{[1,2]} + d_{\mathrm{IS}}^{[2,1]}\right) + t_{i1}^2\left(d_{\mathrm{S}}^{[1]2} + 2d_{\mathrm{SS}}^{[1,2]} + d_{\mathrm{S}}^{[2]2}\right) + \sigma^{[1]2}$.

A special case of the multivariate random effects models is the latent variable approach. If the random effects are perfectly correlated, that is if $\boldsymbol{b}_i^{[1]} = g\,\boldsymbol{b}_i^{[2]}$, where $g$ is a constant, then this model is equivalent to assuming a common latent variable with a multivariate normal distribution (see e.g. Gueorguieva (2001)).

Statistical software for fitting linear mixed models (R packages `nlme,lmer`, SAS `proc mixed`) are not designed for multivariate mixed models but can be used with a customized set-up of the dataset (see Subsection 5.3.1 for details). Mickey et al. (1994) and Shah et al. (1997) use the EM algorithm for estimation, treating the random effects as missing data, Schafer and Yucel (2002) follow the same approach but extend the algorithm to cope with "real" missing data. For a discussion of the RIScorr model including interpretation guidance and pitfalls confer Fieuws and Verbeke (2004).

### 4.2.3 Covariance Pattern Models with Kronecker Product Structure

Covariance pattern models are mixed models with no random effects but a residual covariance matrix with a complex time-dependent structure. As described above, multivariate covariance patterns need parsimonious structures to avoid that maximally $\frac{pq(pq+1)}{2}$ distinct parameters of the covariance matrix need to be estimated. As far as we know, the only multivariate extension that has been proposed (among others by Galecki, 1994), is the product covariance model. Its name is due to the specific assumption that the covariance matrix $\boldsymbol{V}_i$ is factorised as a Kronecker product and it is referred to in the following as the model with the Kronecker product structure (KPS). The matrix is separated into the matrix $\boldsymbol{\Delta}$ describing the variance of and the covariance of the biomarkers independently from time and into $\boldsymbol{\Psi}$ containing the within-covariance which is assumed to be the same for all biomarkers.

For $q = 2$ and $p = 3$, without any constraints on $\boldsymbol{\Psi}$, the covariance matrix is

$$\boldsymbol{V}_i = \boldsymbol{\Delta} \otimes \boldsymbol{\Psi} = \begin{pmatrix} \delta^{[1]2} & \delta^{[1,2]} \\ \delta^{[1,2]} & \delta^{[2]2} \end{pmatrix} \otimes \begin{pmatrix} \psi_{11}^2 & \psi_{12} & \psi_{13} \\ \psi_{21} & \psi_{22}^2 & \psi_{23} \\ \psi_{31} & \psi_{32} & \psi_{33}^2 \end{pmatrix}$$

$$= \left(\begin{array}{ccc|ccc} \delta^{[1]2}\psi_{11}^2 & \delta^{[1]2}\psi_{12} & \delta^{[1]2}\psi_{13} & \delta^{[1,2]}\psi_{11}^2 & \delta^{[1,2]}\psi_{12} & \delta^{[1,2]}\psi_{13} \\ \delta^{[1]2}\psi_{21} & \delta^{[1]2}\psi_{22}^2 & \delta^{[1]2}\psi_{23} & \delta^{[1,2]}\psi_{21} & \delta^{[1,2]}\psi_{22}^2 & \delta^{[1,2]}\psi_{23} \\ \delta^{[1]2}\psi_{31} & \delta^{[1]2}\psi_{32} & \delta^{[1]2}\psi_{33}^2 & \delta^{[1,2]}\psi_{31} & \delta^{[1,2]}\psi_{32} & \delta^{[1,2]}\psi_{33}^2 \\ \hline \delta^{[1,2]}\psi_{11}^2 & \delta^{[1,2]}\psi_{12} & \delta^{[1,2]}\psi_{13} & \delta^{[2]}\psi_{11}^2 & \delta^{[2]2}\psi_{12} & \delta^{[2]2}\psi_{13} \\ \delta^{[1,2]}\psi_{21} & \delta^{[1,2]}\psi_{22}^2 & \delta^{[1,2]}\psi_{23} & \delta^{[2]2}\psi_{21} & \delta^{[2]2}\psi_{22}^2 & \delta^{[2]2}\psi_{23} \\ \delta^{[1,2]}\psi_{31} & \delta^{[1,2]}\psi_{32} & \delta^{[1,2]}\psi_{33}^2 & \delta^{[2]2}\psi_{31} & \delta^{[2]2}\psi_{32} & \delta^{[2]2}\psi_{33}^2 \end{array}\right)$$

In the top schema of Figure 4.1, p. 36, the blue arrows (representing the autocorrelations) are captured by the covariance pattern of $\boldsymbol{\Psi}$ in this model. The variances of the biomarkers are the entry-wise product of the diagonal entries of $\boldsymbol{\Delta}$ and $\boldsymbol{\Psi}$. The green arrows are the product of the between-biomarker covariance and the within-biomarker variances. The red arrows representing the cross-correlations are the autocorrelations scaled (through multiplication) by the between-biomarker correlation.

As both matrices, $\boldsymbol{\Delta}$ and $\boldsymbol{\Psi}$, are symmetric, the covariance matrix of the second biomarker, denoted as submatrix $\boldsymbol{E}$ in Equation (4.1), is proportional to the covariance matrix of the first biomarker. The product-covariance model sets

$$\boldsymbol{E} = \upsilon_1 \boldsymbol{A} = \frac{\delta^{[2]2}}{\delta^{[1]2}} \boldsymbol{A}$$

with the scale variance parameter $\upsilon_1$. For the cross-covariance matrix $\boldsymbol{B} = \boldsymbol{C}$, there is also a proportional relationship in that

$$\boldsymbol{C} = \upsilon_2 \boldsymbol{\Psi} = \delta^{[1,2]} \boldsymbol{\Psi}$$

with the scale cross-covariance parameter $\upsilon_2$. Weiss (2005) notes that in the product covariance model, the within-time cross-correlations always have higher absolute values than the between-time cross-correlations which is a reasonable assumption. This becomes obvious when comparing the within-time cross-correlation

$$\mathrm{corr}(w_{ij}^{[\ell]}, w_{ij}^{[\ell']}) = \frac{\delta^{[1,2]} \psi_{11}^2}{\delta^{[1]} \delta^{[2]} \psi_{11}^2} = \frac{\delta^{[1,2]}}{\delta^{[1]} \delta^{[2]}}$$

with the between-time cross-correlation

$$\mathrm{corr}(w_{ij}^{[\ell]}, w_{ij'}^{[\ell']}) = \frac{\delta^{[1,2]} \psi_{12}}{\delta^{[1]} \psi_{11} \delta^{[2]} \psi_{22}} = \frac{\psi_{12}}{\psi_{11} \psi_{22}} \cdot \mathrm{corr}(w_{ij}^{[\ell]}, w_{ij}^{[\ell']})$$

and realizing that the first term of the last equation is a correlation and therefore less than one in absolute value. A further implied assumption is the symmetry of the cross-correlations.

For univariate covariance pattern models, common covariance structures include compound symmetry (CS), autoregressive (AR), banded, Toeplitz, unstructured (UN) and others (Verbeke and Molenberghs, 2000). In the bivariate, an unstructured covariance pattern is assumed for $\boldsymbol{\Delta}$ and for $\boldsymbol{\Psi}$ either compound symmetry (CS), an continuous autoregressive process of order 1 (CAR1) or an unstructured (UN) pattern. These structures are denoted as UN $\otimes$ CS, UN $\otimes$ CAR1 and UN $\otimes$ UN.

Caution is advised when implementing the Kronecker product structure as some indetermination may result (Galecki, 1994): The non-identifiability arises from the fact that if $\boldsymbol{\Delta} \otimes \boldsymbol{\Psi}$ is the overall covariance matrix, then there exists a continuum of other pairs of covariance matrices, e.g. $c \cdot \boldsymbol{\Delta}$ and $\boldsymbol{\Psi}/c$ $(c > 0)$, which results in the same Kronecker product. The covariance matrix $\boldsymbol{\Psi}$ is then rescaled to yield $\psi_{11} = 1$, assuring the identifiability of the matrices $\boldsymbol{\Delta}$ and $\boldsymbol{\Psi}$. The remaining entries of the overall covariance matrix may then be interpreted as ratios on the scale of $\psi_{11}$. Further on, the covariance matrix $\boldsymbol{\Psi}$ equals the correlation matrix for homogeneous variance structures like CS and CAR1.

Together with this constraint, the covariance matrix $\boldsymbol{\Psi}$ is as follows for the three considered model structures:

- CS:
  The constraint implies that $\sigma^2 = 1 - \psi^2$ and therefore
  $\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I} + \psi^2 \boldsymbol{J} = (1 - \psi^2)\boldsymbol{I} + \psi^2 \boldsymbol{J}$

where $\boldsymbol{I}$ is the $p$-dimensional identity matrix and $\boldsymbol{J}$ a $p$-dimensional matrix of 1's.

For $p = 3$ for example, it is $\begin{pmatrix} 1 & \psi^2 & \psi^2 \\ \psi^2 & 1 & \psi^2 \\ \psi^2 & \psi^2 & 1 \end{pmatrix}$.

- CAR1:
  The constraint implies that $\sigma^2 = 1$ and therefore
  $\boldsymbol{\Psi} = \left[\sigma^2 \cdot \psi^{|t_{ij} - t_{i(j+h)}|}\right] = \left[\psi^{|t_{ij} - t_{i(j+h)}|}\right]$ with $\psi \in [0; 1]$.
  The time points $j$ and $j + h$ are $h$ lags apart for the $h^{th}$ superdiagonal and $h = 0$ for the diagonal of the matrix. Thus, the matrix will be different for each individual $i$ if the individual times between biomarker measurements vary from patient to patient.

- UN:
  The constraint implies that $\sigma_1^2 = 1$. This is the only condition which is placed on the covariance matrix $\boldsymbol{\Psi}$ in the unstructured case.
  For $p = 3$ for example, $\boldsymbol{\Psi} = \begin{pmatrix} 1 & \psi_{12}\sigma_2 & \psi_{13}\sigma_3 \\ \psi_{12}\sigma_2 & \sigma_2^2 & \psi_{23}\sigma_2\sigma_3 \\ \psi_{13}\sigma_3 & \psi_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{pmatrix}$.

The number of parameters in the covariance matrix to be estimated are $\frac{q(q+1)}{2} + 1$ for UN $\otimes$ CS and UN $\otimes$ CAR1 and $\frac{q(q+1)}{2} + \frac{p(p+1)}{2} - 1$ for UN $\otimes$ UN.

For the proposed model structures, the time component, the distinctive feature of longitudinal data compared to cross-sectional ones, is only partially taken into account: for CS and UN by the order of the biomarker measurements, for CAR1 by the time distance between the measurements. The first models assume equidistant visits and modifying the time scale has no effect whatsoever on the estimated model parameters. This is a general limitation of covariance pattern models.

In the univariate case, the marginal representation of a mixed model with a random intercept equals that of a covariance pattern model with a compound symmetry structure (Verbeke and Molenberghs, 2000). Due to the strong assumptions underlying a covariance pattern model with a KPS, the equivalence between the multivariate RI model (with uncorrelated residual variances as above) and the UN $\otimes$ CS structure is only valid under the following three assumptions:

$$\begin{aligned} \sigma^2 &= 2 \\ \delta^{[1]2} &= \frac{\sigma^{[1]2}}{2} \\ \delta^{[2]2} &= \frac{\sigma^{[2]2}}{2} \end{aligned}$$

### Estimation

The computational benefits of the Kronecker product structure become evident when considering the model estimation. But at first, a general outline of the estimation is given. The full log-likelihood function $l^{ML}$ is based on the marginal model $\boldsymbol{w} \sim N(\boldsymbol{X\beta}, \boldsymbol{V}(\boldsymbol{\theta}))$ where $\boldsymbol{\theta} = (\theta_o)$, $o = 1, \ldots, O$ is a vector containing the distinct covariance parameters to be estimated. Maximising the likelihood (omitting constant terms)

$$l^{ML}(\boldsymbol{w}|\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{V}(\boldsymbol{\theta})| - \frac{1}{2}(\boldsymbol{w} - \boldsymbol{X\beta})^T \boldsymbol{V}(\boldsymbol{\theta})^{-1}(\boldsymbol{w} - \boldsymbol{X\beta}) \tag{4.3}$$

with respect to $\boldsymbol{\beta}$ for fixed $\boldsymbol{\theta}$ yields

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) = (\boldsymbol{X}^T\boldsymbol{V}(\boldsymbol{\theta})^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{V}(\boldsymbol{\theta})^{-1}\boldsymbol{w}. \tag{4.4}$$

As it is a well-known result (Verbeke and Molenberghs, 2000) that a full maximum likelihood approach provides biased estimates for the covariance parameters $\boldsymbol{\theta}$, a restricted maximum likelihood (REML) approach (Patterson and Thompson, 1971) is preferred. REML corrects for the loss of degrees of freedom when estimating $\boldsymbol{\beta}$ and therefore produces less biased variance parameters than ML. There are several derivations available for mixed models (Patterson and Thompson, 1971; Harville, 1977; Verbyla, 1990). The conditional derivation given by Verbyla (1990) is presented in the following.

Let the matrix $\boldsymbol{M} = [\boldsymbol{M}_1, \boldsymbol{M}_2]^T$ be a non-singular matrix where $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are $pq \times uq$ and $pq \times (pq - uq)$ matrices respectively. The submatrices are chosen to satisfy the following conditions

$$\boldsymbol{M}_1^T\boldsymbol{X} = \boldsymbol{I}_{(uq \times uq)} \qquad \text{and} \qquad \boldsymbol{M}_2^T\boldsymbol{X} = \boldsymbol{0}.$$

Transforming $\boldsymbol{w}$ by these matrices, the new partitioned response vector is

$$\begin{pmatrix} \boldsymbol{M}_1^T\boldsymbol{w} \\ \boldsymbol{M}_2^T\boldsymbol{w} \end{pmatrix} = \begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{pmatrix} \sim N\left[\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{0} \end{pmatrix}, \sigma^2 \begin{pmatrix} \boldsymbol{M}_1^T\boldsymbol{V}\boldsymbol{M}_1 & \boldsymbol{M}_1^T\boldsymbol{V}\boldsymbol{M}_2 \\ \boldsymbol{M}_2^T\boldsymbol{V}\boldsymbol{M}_1 & \boldsymbol{M}_2^T\boldsymbol{V}\boldsymbol{M}_2 \end{pmatrix}\right]$$

and consequently, the distribution of $\boldsymbol{w}_2$ is independent on $\boldsymbol{\beta}$. The joint likelihood of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ can be factorized into the conditional and the marginal likelihood as

$$L^{REML}(\boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{w}_1, \boldsymbol{w}_2) = L^{REML}(\boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{w}_1|\boldsymbol{w}_2) \cdot L^{REML}(\boldsymbol{\theta}; \boldsymbol{w}_2). \tag{4.5}$$

The log-likelihood of the conditional distribution of $\boldsymbol{w}_1|\boldsymbol{w}_2$ omitting constant terms is

$$l^{REML}(\boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{w}_1|\boldsymbol{w}_2) = \frac{1}{2}\Big(\log|\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}| - \boldsymbol{\epsilon}_1^T\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\boldsymbol{\epsilon}_1\Big)$$

where $\boldsymbol{\epsilon}_1 = \boldsymbol{w}_1 - \boldsymbol{\beta} - \boldsymbol{w}_2^*$ with $\boldsymbol{w}_2^* = \boldsymbol{M_1^T V M_2 (M_2^T V M_2)^{-1} w}_2$. This conditional likelihood is independent from the choice of $\boldsymbol{M}$, it only depends on $\boldsymbol{X}$. The score vector for $\boldsymbol{\beta}$ is

$$\boldsymbol{s}^{REML}(\boldsymbol{\beta}) = \boldsymbol{X^T V^{-1} X \epsilon_1},$$

yielding the same estimate for $\hat{\boldsymbol{\beta}}$ as under ML (Eq. (4.4)). The log-likelihood of the marginal distribution of $\boldsymbol{w}_2$ is the likelihood of observing the sample residuals (not the sample data) and, ignoring constants, can be rewritten as

$$\begin{aligned} l^{REML}(\boldsymbol{\theta}; \boldsymbol{w}_2) &= -\frac{1}{2}\Big(\log|\boldsymbol{V}| + \log|\boldsymbol{X^T V^{-1} X}| + \hat{\boldsymbol{\epsilon}}^{\boldsymbol{T}} \boldsymbol{V}^{-1} \hat{\boldsymbol{\epsilon}}\Big) \\ &= -\frac{1}{2}\Big(\log|\boldsymbol{V}| + \log|\boldsymbol{X^T V^{-1} X}| + \boldsymbol{w^T P w}\Big) \end{aligned} \tag{4.6}$$

where $\hat{\boldsymbol{\epsilon}} = \boldsymbol{w - X\hat{\beta}}$ and $\boldsymbol{P = V^{-1} - V^{-1} X (X^T V^{-1} X)^{-1} X^T V^{-1}}$.
The score vector $\boldsymbol{s}^{REML}(\boldsymbol{\theta}) = (s^{REML}(\theta_1), \ldots, s^{REML}(\theta_O))^T$ contains the entries

$$s^{REML}(\theta_o) = -\frac{1}{2}\text{tr}\Big(\boldsymbol{P}\frac{\partial \boldsymbol{V_i}}{\partial \theta_o}\Big) + \frac{1}{2}\boldsymbol{w^T P}\frac{\partial \boldsymbol{V_i}}{\partial \theta_o}\boldsymbol{Pw}. \tag{4.7}$$

for $o = 1, \ldots, O$. A detailed derivation of the REML results is found in Taylor (2005).

In the multivariate setting, the estimation can be a computionally intensive estimation as $\boldsymbol{V}_i$ of size $pq \times pq$ needs to be differentiated and inverted (cf. Equations (4.4)–(4.7)). Due to the assumed Kronecker product structure, however, it is not and involves only the less dimensional differentiation and inversion of $\boldsymbol{\Delta}_{q\times q}$ and $\boldsymbol{\Psi}_{p\times p}$. Here, $\boldsymbol{\theta} = (\boldsymbol{\delta}, \boldsymbol{\psi})$ and contains all distinct covariance parameters of $\boldsymbol{\Delta}$ resp. $\boldsymbol{\Psi}$. We have

$$\frac{\partial \boldsymbol{V_i}}{\partial \theta_o} = \begin{cases} \frac{\partial \boldsymbol{\Delta}}{\partial \theta_o} \otimes \boldsymbol{\Psi} & \text{if } \theta_o \in \boldsymbol{\delta} \\ \boldsymbol{\Delta} \otimes \frac{\partial \boldsymbol{\Psi}}{\partial \theta_o} & \text{if } \theta_o \in \boldsymbol{\psi} \end{cases}$$

and

$$\boldsymbol{V}_i^{-1} = (\boldsymbol{\Delta} \otimes \boldsymbol{\Psi})^{-1} = \boldsymbol{\Delta}^{-1} \otimes \boldsymbol{\Psi}^{-1}.$$

When assuming a covariance structure such as CS or CAR1 for $\boldsymbol{\Psi}$, the estimation formulae are even more simplified. The resulting partial derivatives and the inverse for those two structures are

- CS:

$$\frac{\partial \boldsymbol{\Psi}}{\partial \psi} = -2\psi\,\boldsymbol{I} + 2\psi\,\boldsymbol{J}$$

$$\boldsymbol{\Psi}^{-1} = \frac{1}{(1-\psi^2)}\boldsymbol{I} - \frac{\psi^2}{(1-\psi^2)(1+(p-1)\psi^2)}\boldsymbol{J}$$

- CAR1:

$$
\begin{aligned}
\frac{\partial \boldsymbol{\Psi}}{\partial \psi} &= \Big[ |t_{ij} - t_{i(j+h)}| \cdot \psi^{(|t_{ij} - t_{i(j+h)}|-1)} \quad \Big] \\
\boldsymbol{\Psi}^{-1} &= \frac{1}{1 - \psi^{2|t_{ij} - t_{i(j'+1)}|}} \cdot \mathrm{diag}(1, 1, \dots, 1, 0) + \frac{1}{1 - \psi^{2|t_{ij} - t_{i(j'-1)}|}} \cdot \mathrm{diag}(0, 1, \dots, 1, 1) \\
&\quad - \frac{\psi^{|t_{ij} - t_{ij'}|}}{1 - \psi^{2|t_{ij} - t_{ij'}|}} \boldsymbol{J}^{*}
\end{aligned}
$$

  where $\boldsymbol{J}^{*}$ is a tridiagonal matrix with zeros on the diagonal and ones on the first super- and the first subdiagonal.

As can be seen in Equation (4.5), the estimation of $\boldsymbol{\theta}$ which is based on the second factor does not depend on $\boldsymbol{\beta}$. Therefore, the selected approach for the REML estimation was to first get an estimate for $\boldsymbol{\theta}$ by numeric constrained optimization according to Equations (4.6) and (4.7) and second, to compute $\hat{\boldsymbol{\beta}}$ according to Equation (4.4). The numeric constrained optimization was done by the `R` function `optim` based on the L-BFGS-B algorithm (see Kohlmann (2005) and the references therein). Besides the likelihood function and the gradient function, start values and constraints for the parameters were given. The partial derivatives under UN $\otimes$ UN were derived by the `R` package `Ryacas` (Goedman et al., 2008).

Other alternatives for the numerical computation would have been a Fisher-Scoring algorithm as presented by O'Brien and Fitzmaurice (2005).The L-BFGS-B algorithm was opted for as no information matrix needs to be computed. For the simpler case where the mean is assumed to be unstructured and estimated by the average value over $\boldsymbol{w}$, the estimation of $\boldsymbol{\theta}$ does not longer depend on $\beta$. Dutilleul (1999) proposes a flip-flop algorithm, estimating $\boldsymbol{\theta}$ by iterating between the estimation of $\boldsymbol{\Delta}$ for fixed $\boldsymbol{\Psi}$ and vice versa. Further, a likelihood ratio test was derived to test whether the covariance matrix is separable with a KPS (Lu and Zimmerman, 2005; Mitchell et al., 2006; Roy and Khattree, 2003). The flip-flop algorithm is faster than a Fisher-Scoring algorithm (for comparisons see Lu and Zimmerman, 2005) and even faster, non-iterative methods have recently been proposed by Werner et al. (2008).

## 4.3 Application of Multivariate longQDA: RA Therapy Resistance Data

The application data set for multivariate longQDA is again about therapy resistance, but this time with respect to a drug for patients suffering from rheumatoid arthritis. Seven biomarkers (coded as BM1 up to BM7) were repeatedly measured throughout the study, first at the beginning of the study, then 1, 4 and 6 months thereafter. The drug was administered at the beginning of the study (after the blood sample was taken) and two weeks later. A patient was defined as non-resistant according to an improvement after 6 months based on the ACR50 criteria (Felson et al., 1993) involving criteria such as the status of the joints, disease activity and pain intensity. This definition is the Gold Standard, the reference classification. The study population under treatment comprises 55 non-resistant and 110 resistant patients with complete data of all 4 visits.

The biomarker profiles were transformed to better meet the normality assumptions of longQDA, biomarker 1 to 6 by $\ln(x + 10)$, biomarker 7 by the variance-stabilizing arcsine transformation suitable for proportions. The individual profiles as well as their group-wise summary are displayed in Figure 4.2. Biomarker 1 to 4 exhibit a downwards pattern over time for both groups whereas for biomarker 5 to 7, there are only few dynamics over time. The heterogeneity is especially peculiar for biomarker 2. At visit 1, the measurements had not been affected by the drug yet. Given that there is no selection bias present regarding the individual medication history, the classification at this point in time assesses what could be termed 'predisposition' for being resistant. Except for biomarker 5, the non-resistant patients had higher initial biomarker levels than the resistant ones and the group difference in relation to the variance was highest for biomarker 4. During the treatment, all biomarker profiles except that of biomarker 1 decreased more for non-resistant patients over time.

The autocorrelations (squares on the diagonal of Figure 4.3(a) and 4.3(b)) were high, similar in both groups and higher than the cross-correlations (squares on the off-diagonal). For none of the biomarkers, a pattern that cross-correlations at the same time points differed structurely from those at different time points was observed. The cross-correlations indicated two conceptual groupings of biomarkers: biomarker 6 and 7 with relations to 1, and biomarker 2 to 5. Between biomarkers of those groups, the cross-correlations were negligible in size. There was a slight tendency for resistant patients to have higher correlated biomarkers. This was in accordance with the smaller decrease of the profiles over time.

The model selection for the estimation of the group-specific mean and covariance was based on the minimal BIC as exposed in Chapter 3. For the univariate assessment, RI, RIS and RICAR1 structures were fitted and for the bivariate, the structures presented in Section 4.2.1-4.2.3: RIind, RISind, RICAR1ind, RIcorr, RIScorr, UN$\otimes$CS, UN$\otimes$CAR1 and UN$\otimes$UN. A fixed intercept and a fixed linear slope (with study time measured in months) were always part of the mixed model. In each group, the model structure that was selected in the majority of the 50 MCCV loops was determined. If the selected structure differed between non-resistant and resistant patients, the results of the structure with the highest overlap between the groups were reported. This is analogous to the approach applied previously.

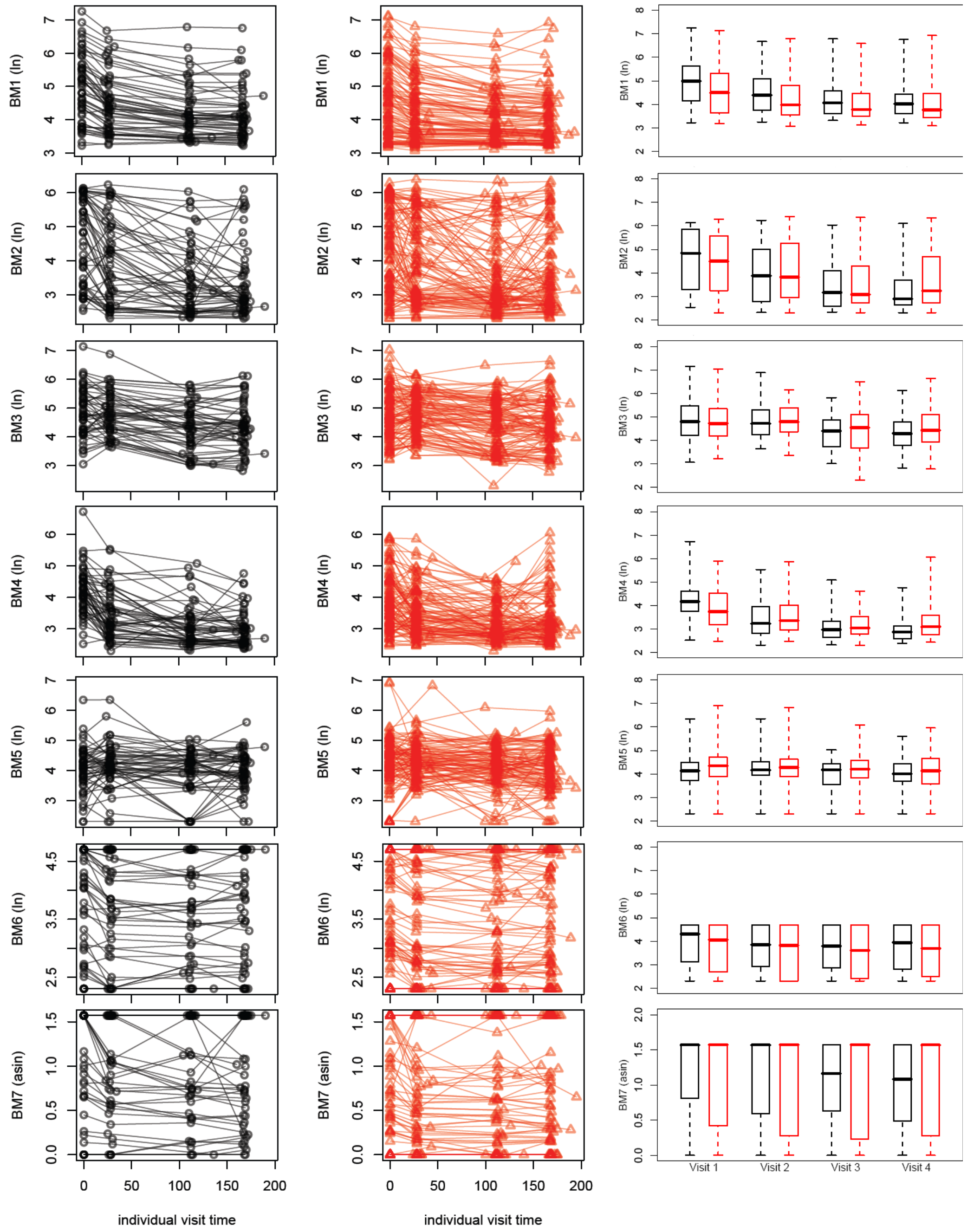
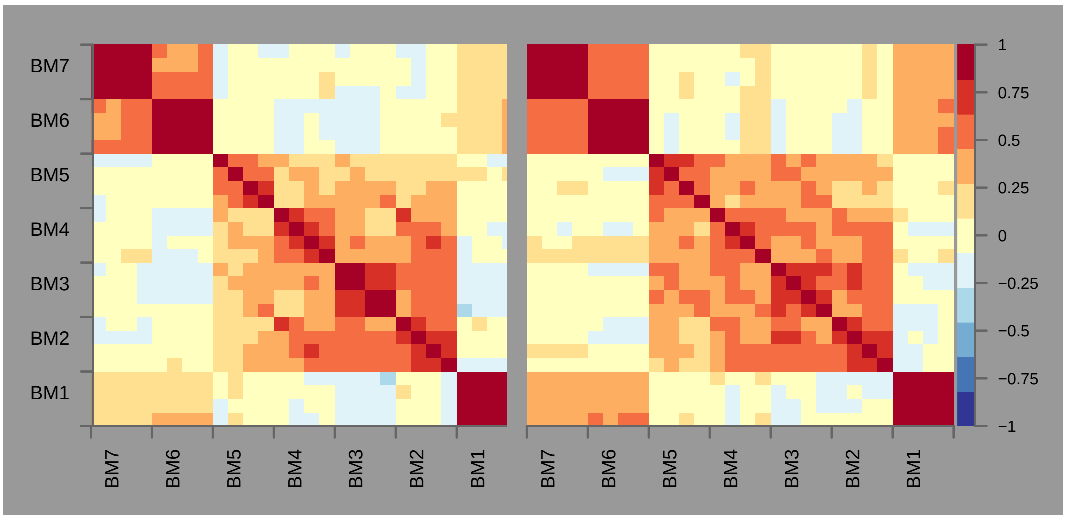


Figure 4.2: Individual profiles of biomarker indicative for RA therapy resistance (in black non-resistant, in red resistant patients)



(a) non-resistant patients (b) resistant patients

Figure 4.3: Auto- and cross-correlations of transformed biomarkers

For 3 as well as for 4 visits, RIS was selected for BM1 (3 visits: in 88% of the MCCV loops for non-resistant, in 100% for resistant patients; 4 visits: 94%, 100%), RI for the BM2, 4-7 (3 visits: 92% and more, 65% and more; 4 visits: 65% and more, 65% and more). There were two discrepancies with differing model structures: The RIS structure was selected in 86% and RI only in 14% for BM4 in the resistant group with 3 visits and also in 52% RIS versus 34% RI for BM2 in the non-resistant group with 4 visits. For BM3, RI was selected in most of the MCCV loops with 3 visits (98%, 50%), but RICAR1 with 4 visits (46%, 62%).

In the bivariate longQDA, RISind was chosen for all pairs with BM1, RIind for pairs with BM5, BM6 or BM7 (except RIcorr for BM3 with BM5 and BM5 with BM6 for 4 visits) and RIcorr for the pairs BM2 with BM3 or BM4, BM3 with BM4 and BM6 with BM7 for 3 and 4 visits. For 3 visits, the model selection was not always consistent for both groups in that RIind was rather appropriate for the non-resistant patients. This matches with the simulation results of Section 3.4 that a correct selection of the model structure is more difficult for shorter biomarker profiles. The selection of the MMS for the other biomarker combinations was unambiguous with more than 50% frequency in both groups. The distinction between a MMS with or without cross-correlations matches with the empirical cross-correlations illustrated in Figure 4.3. The RIcorr structure was selected for models with empirical cross-correlations of more than 0.25. Unsurprisingly, RIScorr did not converge in all of the MCCV samples for the biomarker pairs with low empirical cross-correlations. Sometimes there were convergence problems with UN⊗UN which is inflexible due to the combination of the maximum degrees of freedom for the estimation of the autocorrelations and the minimal proportionality structure between the biomarkers. None of the Kronecker product structures was selected as the most appropriate one in this application.

As an example, the differences of the fixed mean profiles and the covariance parameters between the empirical and those of the different MMS are illustrated in Figure 4.4 resp. Figures 4.5 and 4.6 for the biomarker pair BM3 and BM4.

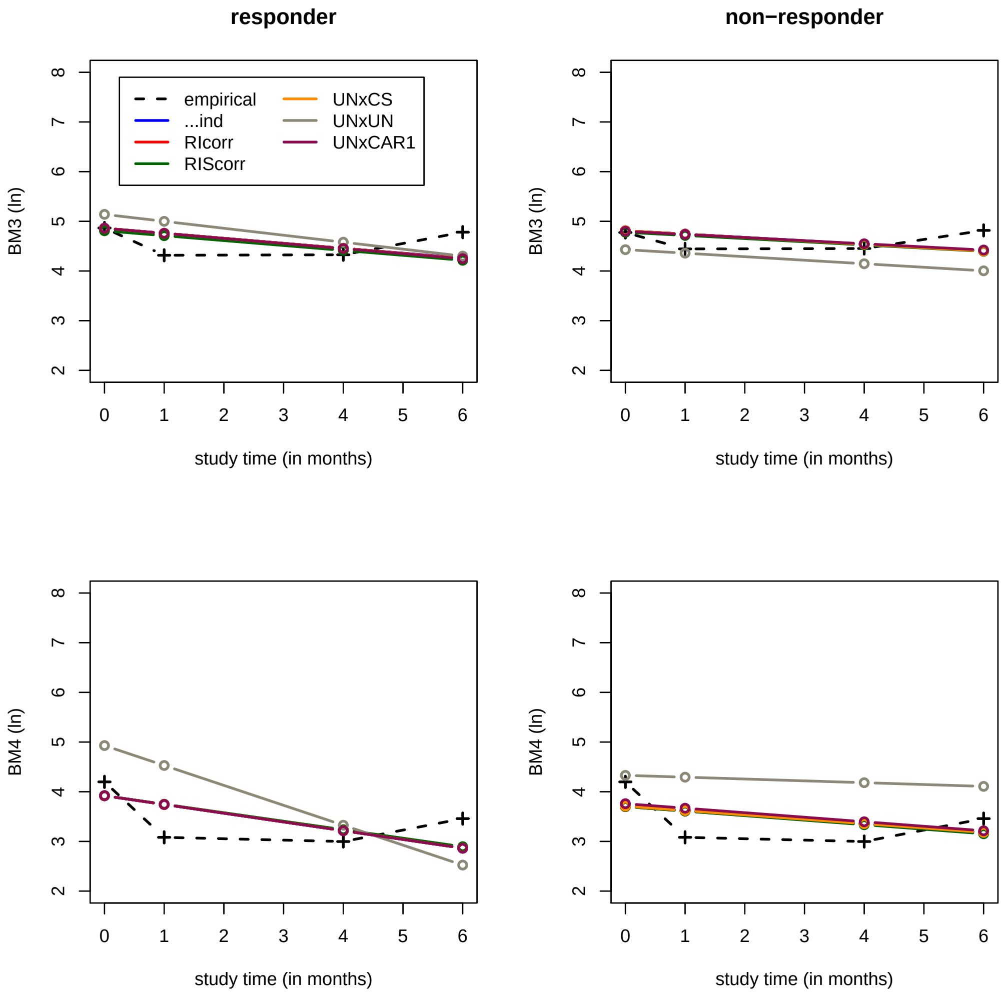


Figure 4.4: Empirical and estimated fixed mean profiles of BM3 and BM4

The estimated linear mean profile was the same for all model structures except that of UN⊗UN and matched well with the empirical equivalent in both groups. The mismatch of the empirical profiles to the estimated profiles for UN⊗UN was a first evidence for the inappropriateness of this covariance structure. It was more pronounced for BM4 than for

BM3 where it even led to a falsely high difference between the profiles of the two patient groups.

The empirical autocorrelation parameters of the non-resistant patient group were high for BM3 and higher between visit 1 and 2 and visit 3 and 4, and decreased slightly for BM4 with an increasing time difference between the visits (Figure 4.5). The cross-correlations were low and smallest between BM3 at visit 1 and 2 and BM4 at visit 3 and 4.

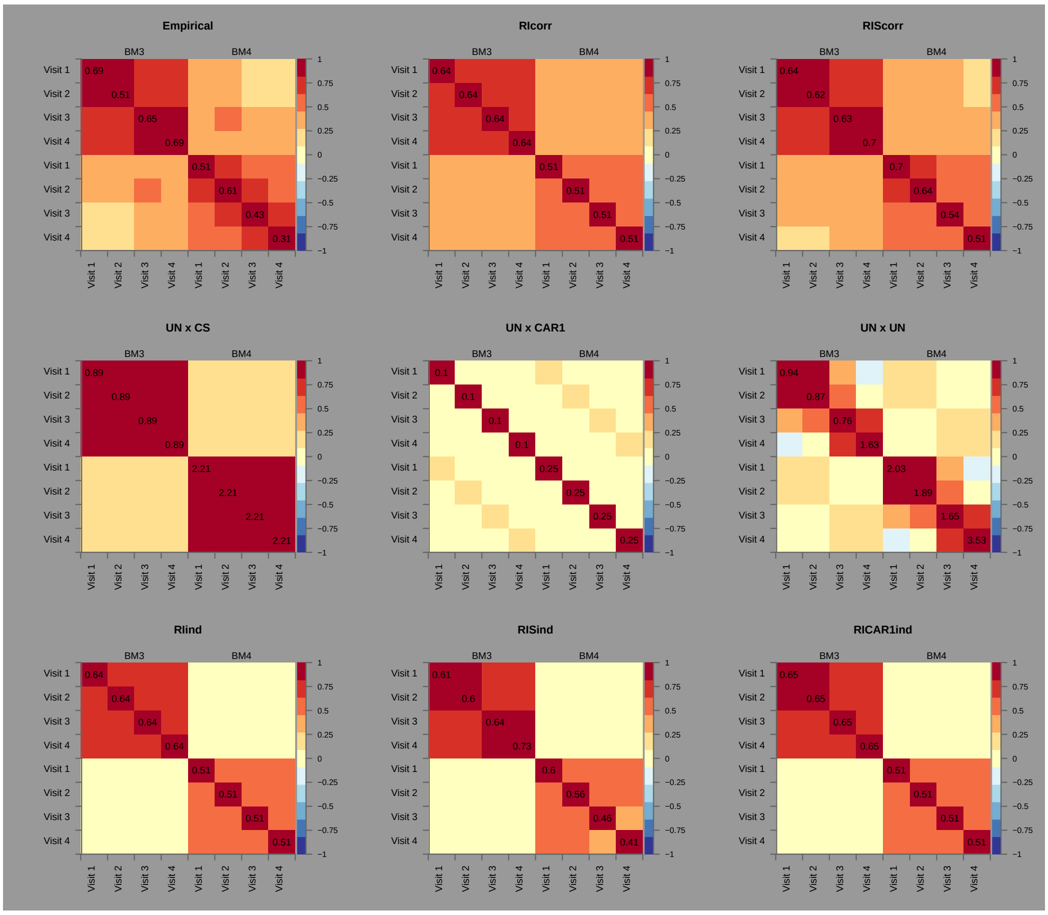


Figure 4.5: Empirical and estimated auto- and cross-correlations of BM3 and BM4 with variance parameters printed on the diagonal, non-resistant patients

The biomarker measurements of the resistant group (Figure 4.6) exhibited high time-independent autocorrelations for BM3 and BM4, the cross-correlations at the same visit were of similar size as the autocorrelations but smaller for different visits.

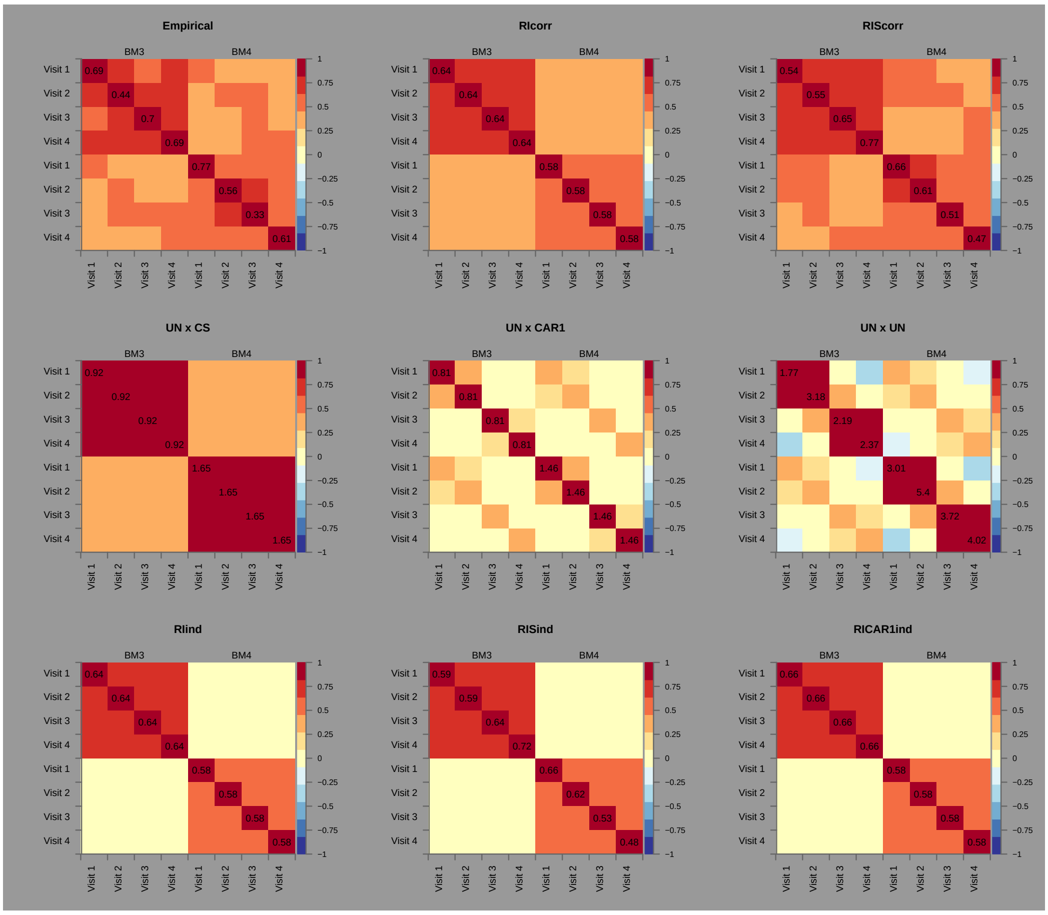


Figure 4.6: Empirical and estimated auto- and cross-correlations of BM3 and BM4 with variance parameters printed on the diagonal, resistant patients

The illustrations of the Kronecker product structures in the second row show the special proportionality assumption of BM3 to BM4 resp. to the covariance between them: The patterns of the estimates are the same. This simplifying assumption yielded mainly to underestimation within both groups. Especially for UN⊗CS, the pattern of the empirical correlations were lost. All KPS structure estimates show lower differences between the group parameters than there were in the data. The RIcorr structure is more parsimonious than RIScorr but captured the most important patterns in the correlations. However, RIScorr outperformed the other MMS regarding the replication of the empirical correlation pattern. The autocorrelation parameters of RIind were similar to the one of RIcorr. But this was not the case for RISind and RIScorr, the estimation of cross-correlations in the latter resulted also in a more precise estimation of the autocorrelations. The estimated RICAR1ind parameters were very similar to those of RIind but the empirical ones did also not exhibit a AR(1) pattern.

This ranking of the structures based on the correlations applies to the variance parameters as well. The models without cross-correlations did not have so highly overestimated variances as the Kronecker product structures. Moreover, unrealistic model structure simplifications carried the danger to decrease or increase the empirically observed group differences and thus the classification performance. The BIC selected RIind in 82%, RIcorr in 10% and RICAR1ind in 8% of the 50 MCCV samples for the non-resistant group and RIcorr in 100% for the resistant group. Compared to the visual ranking, the BIC takes the fit and the number of parameters of the models into account, penalising models with too many parameters.

In the following, the univariate longitudinal classification performance of the biomarkers for the first 3 and 4 visits is presented in comparison to the cross-sectional one of QDA at visit 1. At the first visit, biomarker 1 and 4 had the best cross-sectional performance with an median AUC of 0.62, a BS of 0.785 and a correlation with the Gold Standard of 0.2 (see Table 4.1 for the corresponding confidence intervals and Figure 4.7, first row of the cubes). (The prior probabilities were estimated by the group proportions in all analyses, i.e. $\pi = (1/3, 2/3)$. This yielded $[-0.77, 0.77]$ as limits of the biserial correlation. The reference limit of the Brier Score is 0.778.)

| Perf. Measure | Biomarker | Visit 1 (QDA) | 4 visits (longQDA) |
|---|---|---|---|
| AUC | BM 1 | 0.63 [0.53;0.67] | 0.67 [0.58;0.76] |
| | BM 4 | 0.62 [0.54;0.71] | |
| | BM 4,5 | 0.65 [0.57;0.76] | |
| BS | BM 1 | 0.785 [0.766;0.789] | 0.792 [0.755;0.812] |
| | BM 4 | 0.786 [0.769;0.794] | |
| | BM 4,5 | 0.786 [0.756;0.806] | |
| $\rho_{z_{[1]},p_{[1]}}$ | BM 1 | 0.20 [0.03;0.26] | 0.27 [0.11;0.42] |
| | BM 4 | 0.19 [0.07;0.29] | |
| | BM 4,5 | 0.22 [0.09;0.39] | |

Table 4.1: Performance measures of biomarkers indicative for RA therapy response. They are given as median with 10th and 90th percentile of 50-fold MCCV results.

The group-wise distributions of the estimated posterior probabilities were hardly separated (Figure 4.8(a) and (b)). When using 3 or 4 visits with longQDA, BM2, BM3, BM4 and BM6 gained in performance, BM1, BM5 and BM7 lost. Only BM4 achieved a clearly higher performance than the cross-sectional reference (median AUC: 0.67, BS 0.792, correlation 0.27).

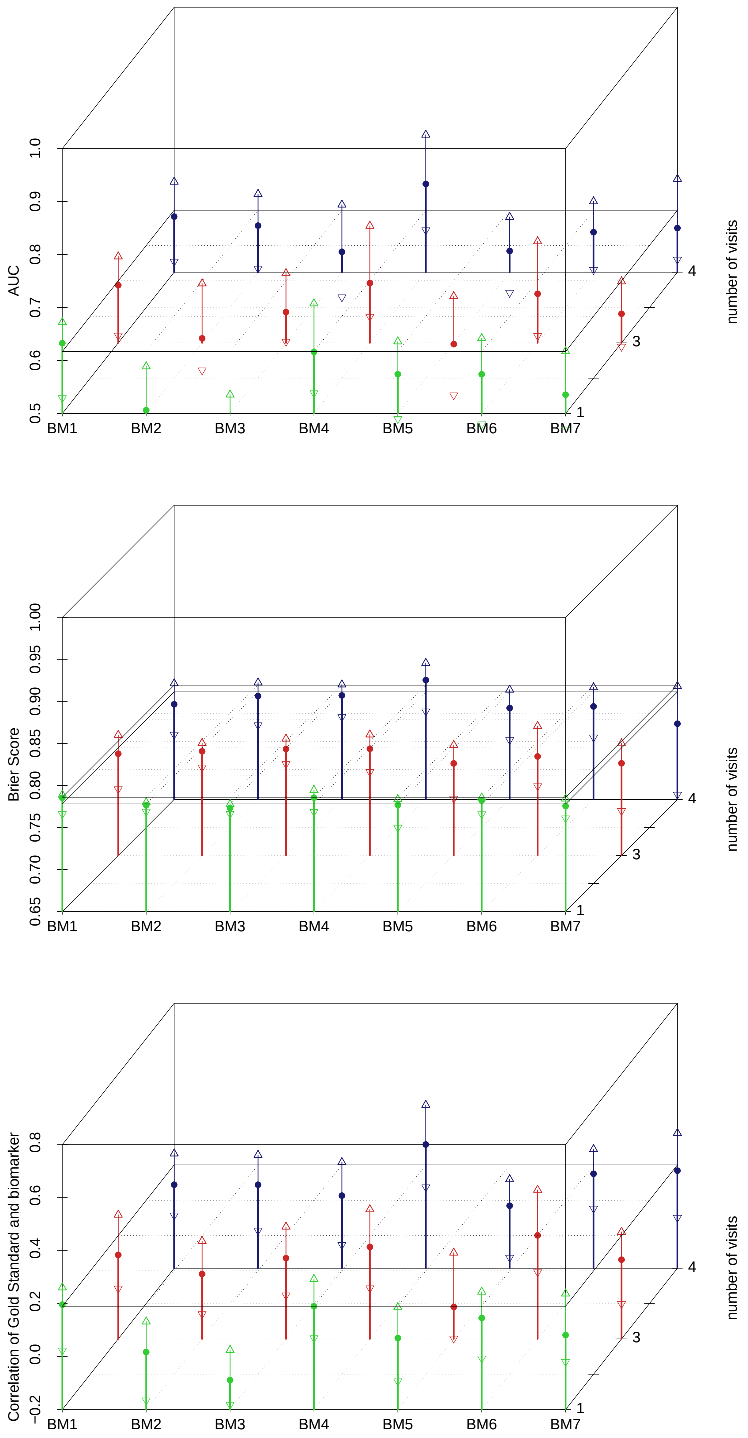


Figure 4.7: Univariate performance of biomarkers, displaying the dependence on the number of visits included in the assessment. The reference grid marks the best cross-sectional performance of BM4, the cube of the BS has a second one at the reference limit of 0.778.

Also at the individual posterior probability level (Figure 4.8(c)), an improvement was observed. But the biomarker was still of minor quality for predicting therapy resistance. The group-wise estimated mixed model parameters for the non-resistants were $\beta_{[0]} = (3.92, -0.176)$, $d^2_{[0]\mathrm{I}} = 0.25$, $\sigma^2_{[0]} = 0.26$ and $\beta_{[1]} = (3.71, -0.088)$, $d^2_{[1]\mathrm{I}} = 0.30$, $\sigma^2_{[1]} = 0.28$ for the resistants. In both groups, the total residual variance was twice as high as the variance of the random intercept, yielding an intraclass correlation of about one third.

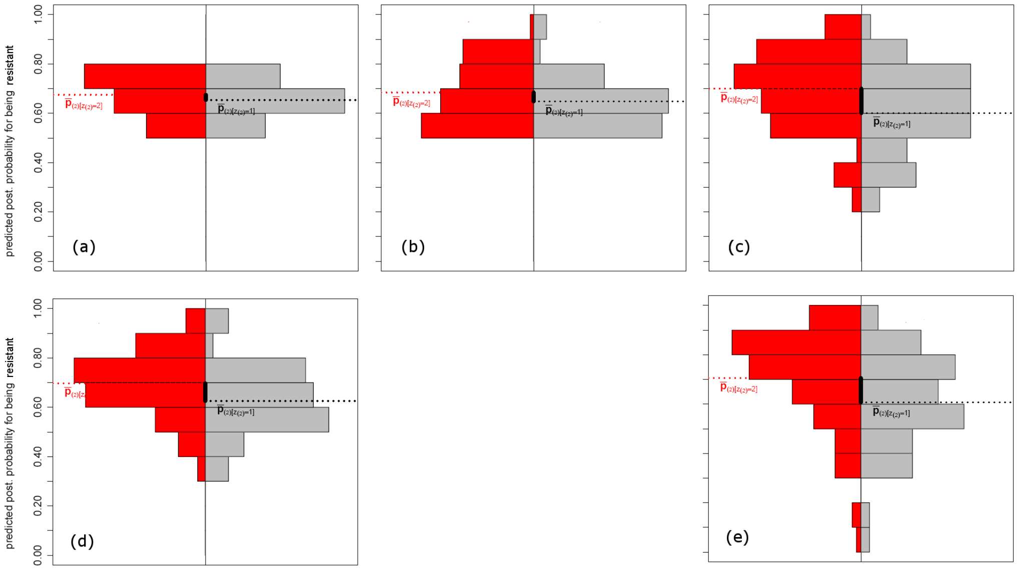


Figure 4.8: Histograms of posterior probabilities for resistant (red) and non-resistant patients (gray). (a) BM1, QDA with data of visit 1; (b) BM4, QDA with data of visit 1; (c) BM 4, longQDA with data of 4 visits; (d) BM4 and BM5, QDA with data of visit 1; (e) BM4 and BM5, longQDA with data of 4 visits.

In the following, the performance of biomarker pairs is considered. The best ones were achieved by combinations with the best univariate biomarker BM4 (Figure 4.9). At the first visit, the pair of BM4 and BM5 achieved a slightly better performance with QDA with a median AUC of 0.65, a BS of 0.786 and a correlation of 0.22 than BM4 as a single marker (Table 4.1 and Figure 4.8(d) vs. (b)). But it was still worse than the univariate longitudinal performance of BM4. This remained also the best performance when combining biomarkers over time with longQDA (Figure 4.9 and Figure 4.8(e) vs. (c)).

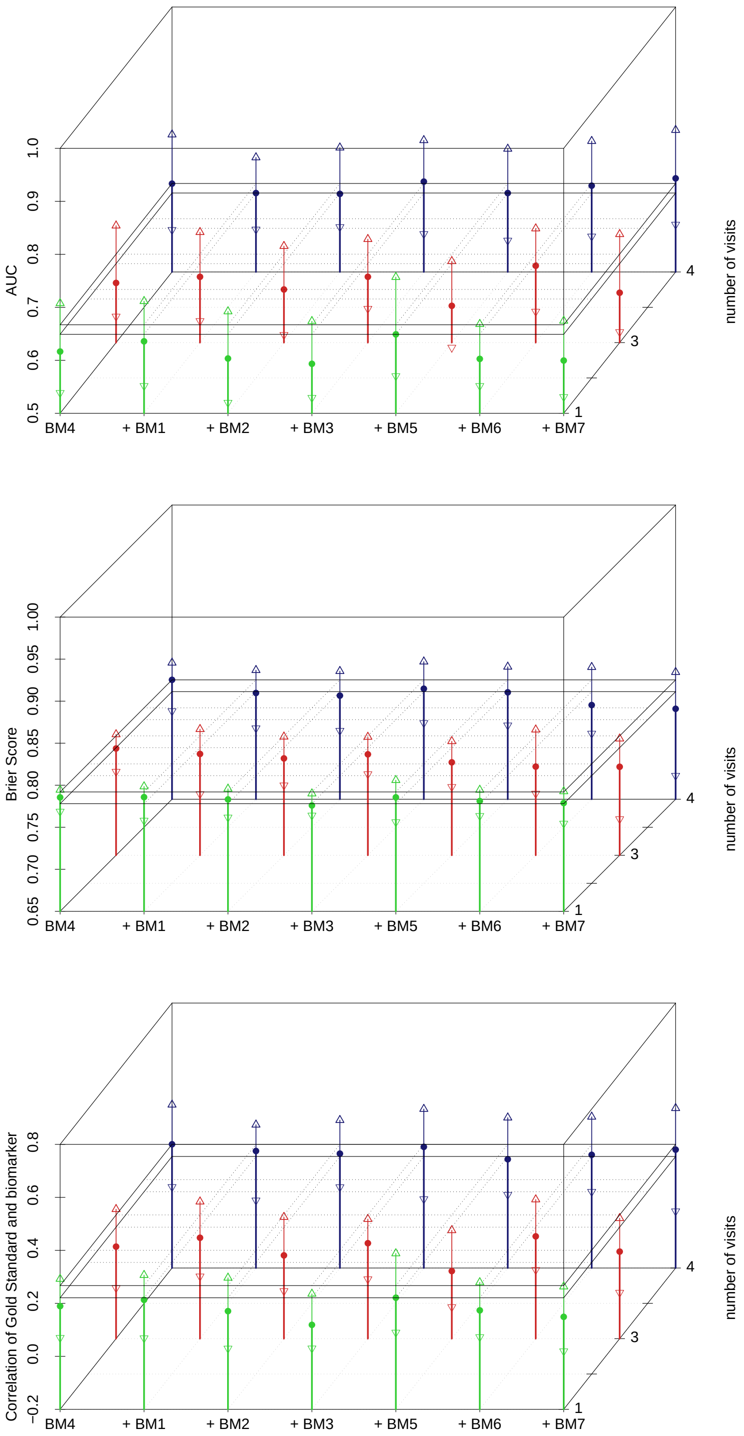


Figure 4.9: Bivariate performance of biomarkers, displaying the dependence on the number of visits included in the assessment. The lower reference grid marks the best cross-sectional performance of the pair BM4,BM5, the higher one the best longitudinal one of BM4.

## 4.4 Improvement by Multivariate Modelling: Some Examples

For the application data, there was no additional gain in performance by combining biomarkers. Therefore, exemplary scenarios where such an improvement has been achieved are presented in the following. For this purpose, bivariate biomarker data were simulated according to a random coefficients model with RIcorr structure (denoted as RIcorr*). This is the simplest bivariate model in that only one additional parameter needs to be estimated comparing the bivariate model with the two univariate models. It is the correlation $\rho_{\mathtt{II}}^{[1,2]}$ between the random intercepts of each biomarker. The univariate, i.e. the biomarker-specific parameters were assumed to be the same. Thus, both biomarkers yield the same univariate classification performance and a potential gain is only triggered by the bivariate modelling. As above, the residuals between the variables were assumed to be independent.

There were 4 simulation scenarios and their parameters are given in Table 4.2. In scenario 1 to 3, the parameters differed only with regard to $\rho_{[k]\mathtt{II}}^{[1,2]}$. The difference of the group-specific correlations of the random intercepts was smallest for scenario 1 with 0.2 ($\rho_{[0]\mathtt{II}}^{[1,2]} = 0.8, \rho_{[1]\mathtt{II}}^{[1,2]} = 0.6$). The difference was increased to 0.6 for scenario 2 ($\rho_{[1]\mathtt{II}}^{[1,2]} = 0.2$) and further to 1.6 for scenario 3 ($\rho_{[1]\mathtt{II}}^{[1,2]} = -0.8$). The fourth scenario exhibits a data setting where the correlations differed as slightly as in scenario 1 but the fixed slopes as well as the residual variances differed in addition by 0.1 each ($\beta_{[0]1} = -0.05, \beta_{[1]1=0.05}$ and $\sigma_{[0]}^2 = 0.3, \sigma_{[1]}^2 = 0.2$).

| Parameter | Scenario | Group 0 | Group 1 |
|---|---|---|---|
| $\beta_0$ | 1,2,3,4 | 3.7 | 3.7 |
| $\beta_1$ | 1,2,3 | 0.05 | 0.05 |
| | 4 | -0.05 | 0.05 |
| $d_{\mathtt{I}}$ | 1,2,3,4 | 0.15 | 0.15 |
| $\rho_{\mathtt{II}}^{[1,2]}$ | 1,4 | 0.8 | 0.6 |
| | 2 | 0.8 | 0.2 |
| | 3 | 0.8 | -0.8 |
| $\sigma^2$ | 1,2,3 | 0.2 | 0.2 |
| | 4 | 0.3 | 0.2 |

Table 4.2: Group-specific parameter settings of simulated RIcorr* models

Biomarker data with 4 visits measured at baseline, 1, 4, and 6 months thereafter were simulated. The population comprised twice the sample sizes of the application data, 110 non-resistant and 220 resistant patients. We applied 50-fold MCCV and did 25 simulation repetitions. For each simulation scenario, two classification performances were evaluated to illustrate the potential gain: the univariate performance by a random coefficient model including the measurements of one biomarker and the corresponding bivariate performance including both biomarkers.

The histograms of the estimated posterior probabilities for group 1 are shown in Figure 4.10 and the resulting performance measures AUC, BS and $\rho_{z_{[1]},p_{[1]}}$ are listed in Table 4.3 for each simulation scenario. The limits of the latter two are the same as for the application data. Note that the univariate results of scenario 2 and 3 were the same as for scenario 1 and were therefore omitted.

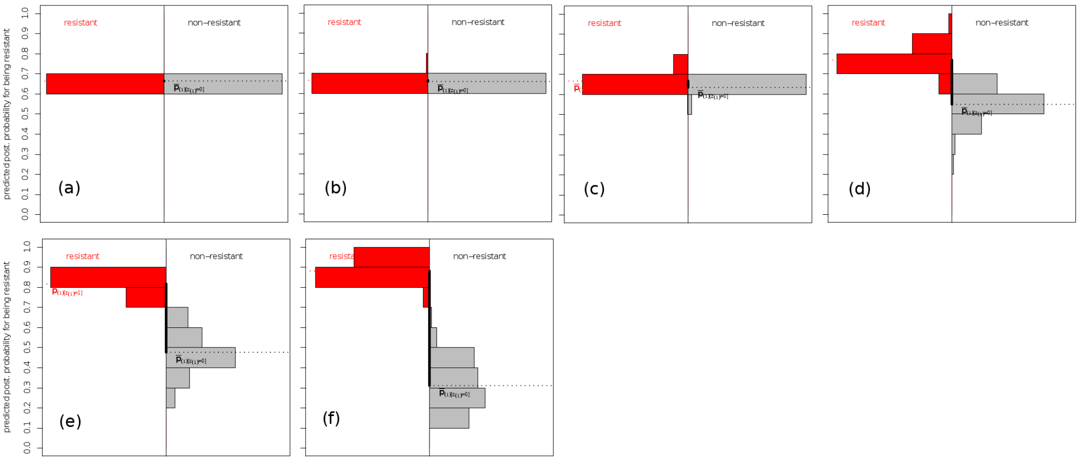


Figure 4.10: Histograms of posterior probabilities for resistant (red) and non-resistant patients (gray), simulated bivariate biomarker data. (a) Sc. 1, RI; (b) Sc. 1, RIcorr; (c) Sc. 2, RIcorr (d) Sc. 3, RIcorr; (e) Sc. 4, RI; (f) Sc. 4, RIcorr.

| Scenario | MMS | AUC | BS | $\rho_{z_{[1]},p_{[1]}}$ |
|---|---|---|---|---|
| 1 | RI | .50 [.42; .57] | .776 [.767; .782] | .00 [−.13; .13] |
| 1 | RIcorr | .52 [.44; .60] | .773 [.760; .785] | .03 [−.11; .17] |
| 2 | RIcorr | .59 [.51; .68] | .782 [.767; .798] | .16 [ .03; .30] |
| 3 | RIcorr | .79 [.73; .84] | .835 [.812; .853] | .50 [ .41; .58] |
| 4 | RI | .81 [.75; .87] | .844 [.818; .867] | .55 [ .44; .64] |
| 4 | RIcorr | .87 [.82; .92] | .873 [.848; .897] | .66 [ .56; .73] |

Table 4.3: Selected performance measures demonstrating the benefit of multivariate modelling. They are given as median with 10th and 90th percentile.

Comparing the univariate performance of scenario 1 with the corresponding bivariate performance ((a) vs. (b)), a performance of very minor quality was achieved in both cases. A difference between the correlation parameters of 0.2 was not sufficient for an improvement and a difference of 0.6 as in scenario 2 was only slightly better ((a) vs. (c)). Scenario 3 exhibited an extreme improvement by bivariate modelling ((a) vs. (d)). A difference of 1.6 yielded an AUC of 0.79 [0.73; 0.84] vs. 0.5 [0.42; 0.57], a BS of 0.835 [0.812; 0.853] vs. 0.776 [0.767; 0.782] and also a clearly higher biserial correlation of 0.50 [0.41; 0.58] vs. 0.00 [−0.13; 0.13].

But a very high difference is not the only possible scenario to yield an improvement by bivariate modeling. Another possible parameter setting is illustrated by scenario 4. The difference between the correlation parameters was with 0.2 as small as in scenario 1 but two further parameters differed slightly between the groups. This yielded quite a good univariate classification performance but it was also further improved by combining the biomarkers ((e) vs. (f)).

# 5 Software Implementation: The R package longQDA

For a wide-spread application of statistical methods, an implementation in a state-of-the-art software environment like R (Ihaka and Gentleman, 1996; R Development Core Team, 2008) is indispensable nowadays. The package longQDA provides the necessary general framework for executing quadratic discriminant analysis with longitudinal data. All univariate and multivariate models presented in Chapter 2 to 4 have been implemented. It comprises about 3500 effective lines of code (determined by LineStats, Fridman (2005)) and is available on request from the Biostatistics Department of Roche Diagnostics GmbH, Penzberg.

In the next section, an application of longQDA is shown for the HIV therapy resistance data set. The results of these analyses have already been presented in Chapter 2. The package longQDA contains these data as an exemplary data set with a univariate biomarker. The implementation of the software follows the object-orientated concept of S4 classes (Chambers, 1998, 2006), which is provided by the R package methods. This conceptual approach determined the software design and is presented in the subsequent section. One of the main advantages of object-oriented programming (OOP) is easy extensibility and this is demonstrated for two features of the software, the multivariate version of longQDA and the generation and analysis of simulated data with the option to use parallel computing. Further possible extensions may be, for example, mixed models with a mixture density for the random effects. Appendix B complements this section by help files of the most important functions in longQDA.

## 5.1 Application: Analyzing Univariate Biomarker Data with longQDA

The package longQDA provides functionality for the entire data analysis process, from the descriptive and explorative analysis of longitudinal data up to the comparison of results, e.g. when contrasting longQDA with different model assumptions or longQDA to QDA.

In the following, the main functionality of `longQDA` is demonstrated with the data set called AIDS2, which contains one biomarker that is indicative for a patient's response to HIV therapy. The following exemplary features of the analysis steps are presented: describing and exploring longitudinal data, defining subanalyses, performing those subanalyses by QDA or univariate longQDA, comparing analysis results and, last but not least, documenting the results.

The documentation is facilitated by a sequential report generation during the entire data analysis process. In every analysis step, the output may be redirected into a report and a preview with the results of these analyses may be displayed. At any time during the analysis process, further additional plots or comments may be added to the report. Having finished the analysis, the user generates a report file (in tex or pdf format) which documents the analyses and contains all the output: plots, tables and comments.

### 5.1.1 Report Setup

Prior to the actual data analysis, it is recommended to set up the report. The constructor function `Report` instantiates a report. In the subsequent stages of the analysis, the report object is filled with the names of the output files and the generated output files are saved to the location specified in the argument `folder`.

```
> rep1 <- Report(folder = "C:/AIDS2/")
```

An instance of class `Report` is a list with four entries, one for the exploratory analysis, one for the analysis setup, one for the longQDA analyses and one for the comparison of results (for more details, refer to B):

```
> str(rep1)
```

```
Formal class 'Report' [package "longQDA"] with 2 slots
  ..@ .Data :List of 4
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data : list()
  .. .. .. ..@ myname: chr "Exploratory Analysis"
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data : list()
  .. .. .. ..@ myname: chr "Analysis Setup"
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data : list()
  .. .. .. ..@ myname: chr "Analysis"
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data : list()
  .. .. .. ..@ myname: chr "Comparisons"
  ..@ folder: chr "C:/AIDS2/"
```

### 5.1.2 Data Import and Creation of the LongData Object

After the report object has been created, the data set 'AIDS2' is imported. As it is contained in the package, it is easily loaded by

```
> data(AIDS2)
```

Otherwise, data can be imported to R in the usual way.

A short description of the data set is obtained by calling ?AIDS2 to display the corresponding help entry. Here is a short overview of the data set:

```
> head(AIDS2)

   PATID TIMEDAY VISIT  CENSOR   BM1     SDURN  GROUP    GROUPVAR
1      1       0     1       1  20161 0.00000000     1 non-resistant
2      1      14     2       1   9699 0.03835616     1 non-resistant
3      1      29     3       1    442 0.07945205     1 non-resistant
4      1      42     4       1    310 0.11506849     1 non-resistant
5      2       0     1       1 172748 0.00000000     1 non-resistant
6      2       8     2       1  40921 0.02191781     1 non-resistant

> str(AIDS2)

'data.frame':   1506 obs. of  8 variables:
 $ PATID   : int  1 1 1 1 2 2 2 2 2 2 ...
 $ TIMEDAY : int  0 14 29 42 0 8 37 64 99 162 ...
 $ VISIT   : int  1 2 3 4 1 2 3 4 5 6 ...
 $ CENSOR  : int  1 1 1 1 1 1 1 1 1 3 ...
 $ BM1     : int  20161 9699 442 310 172748 40921 290 188 49 8 ...
 $ SDURN   : num  0.0000 0.0384 0.0795 0.1151 0.0000 ...
 $ GROUP   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ GROUPVAR: Factor w/ 2 levels "non-resistant",..: 1 1 1 1 1 1 1 1 1 1 ...
```

The variables of the data set are PATID, the patient identification number, TIMEDAY and SDURN, the study time in days resp. years, VISIT, the number of the visit ranging from 1 to 6, CENSOR, information about the censoring, BM1, the measurements of the biomarker HIV RNA, and GROUP and GROUPVAR, the information about the group membership according to the Gold Standard formatted as an integer resp. a factor variable.

Then the raw data set needs to be formatted according to the implemented standardized longitudinal data structure. For this purpose, the user specifies the name(s) of the biomarker(s) and of other important variables like the study time. The scheduled visit times and the coding of the groups need to be provided, too. The constructor function LongData formats the raw data set to comply with the package-specific longitudinal data structure and stores important metainformation about the data which is repeatedly needed in the subsequent analysis steps.

```
> d <- LongData(rawdata=AIDS2, name=dsname, markerlabels="BM1",
+ scheduledtimes=c(0, 2, 4, 8, 16, 24)/52,
+   groups=c("non-resistant" = 1, resistant = 2),
+ timename="SDURN")
```

The implemented standardized structure for longitudinal data is as follows:

```
> str(d)

Formal class 'LongData' [package "longQDA"] with 17 slots
  ..@ id2rows          : int [1:356, 1:6] 1 5 11 13 16 22 28 31 33 38 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:356] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:6] "1" "2" "3" "4" ...
  ..@ visitindex       : int 3
  ..@ timeindex        : int 6
  ..@ markerlabels     : Named chr "BM1"
  .. ..- attr(*, "names")= chr "BM1"
  ..@ markerindices    : int 5
  ..@ scheduledtimes   : num [1:6] 0.0000 0.0385 0.0769 0.1538 0.3077 ...
  ..@ rawdata          :'data.frame':   1506 obs. of  8 variables:
  .. ..$ PATID   : int [1:1506] 1 1 1 1 2 2 2 2 2 2 ...
  .. ..$ TIMEDAY : int [1:1506] 0 14 29 42 0 8 37 64 99 162 ...
  .. ..$ VISIT   : int [1:1506] 1 2 3 4 1 2 3 4 5 6 ...
  .. ..$ CENSOR  : int [1:1506] 1 1 1 1 1 1 1 1 1 3 ...
  .. ..$ BM1     : int [1:1506] 20161 9699 442 310 172748 40921 290 188 49 8 ...
  .. ..$ SDURN   : num [1:1506] 0.0000 0.0384 0.0795 0.1151 0.0000 ...
  .. ..$ GROUP   : int [1:1506] 1 1 1 1 1 1 1 1 1 1 ...
  .. ..$ GROUPVAR: Factor w/ 2 levels "non-resistant",..: 1 1 1 1 1 1 1 1 1 ...
  ..@ name             : chr "AIDS2"
  ..@ groupindex       : int 7
  ..@ grouplabels      : chr [1:2] "non-resistant" "resistant"
  ..@ groupcodes       : num [1:2] 1 2
  ..@ groupcolors      : chr [1:2] "black" "red"
  ..@ groupsymbols     : num [1:2] 1 2
  ..@ grouppriors      : num [1:2] 0.820 0.180
  ..@ fixedgrouppriors : logi FALSE
  ..@ idindex          : int 1
  ..@ markertransformfn: chr(0)
```

Besides the data set with coded biomarker names (BM1, BM2,...) in the slot rawdata, the original biomarker names are stored in the slot markerlabels with the coded names as names of the vector. (For this data set, the biomarker was already coded so there is no difference between the coded and the original biomarker name.) The slot markertransformfn is still empty, but later on, the function names used for the biomarker transformation are stored there, it is e.g "log10". The slot scheduledtimes contains the scheduled visit times from which the individual visit times usually differ. The metainformation of the data set comprises in addition the column numbers of the data set containing important variables such as the patient identification number (idindex), the group (groupindex), the study time and the visits (timeindex and visitindex) and that of the biomarker (markerindices). This allows a fast access to repeatedly requested information by various methods. The same applies to the slot id2rows which contains the row numbers of the data set for each patient at each visit. The other slots contain information about the groups defined by the Gold Standard. These are the prior probability for each group (grouppriors), the labels (grouplabels), the symbols to mark the group membership in plots (groupsymbols) and the numerical codes used in the dataset (groupcodes). The user can choose either to use estimated group priors according to the group proportions in the training sets or user-specified, fixed group priors in the analyses. The default is the first and the choice of the user is saved in the slot fixedgrouppriors. If fixed group priors should be used, they can be provided in the argument priors of LongData.

### 5.1.3 Exploratory and Descriptive Data Analysis

The first step of the analysis is to explore and describe the longitudinal data. By calling plotLongMarkers, plots of the biomarker profiles for the non-resistant and the resistant patients are created (Figure 5.1).

```
> par(mfrow = c(1, 2)) # determines arrangement of plots
> plotLongMarkers(d)
```

Some patients have incomplete biomarker profiles due to missing visits or missing biomarker measurements. These are the shorter or intermittent lines in the figure.

Next, completeCases is used to exclude all patients with less than 6 visits or missing biomarker measurements. The biomarker is then transformed by the log10 function to ease the comparison of the profiles between the two groups.

```
> d <- completeCases(d)
```

```
Records of  271 patients deleted due to incompleteness of the first
6 visits. Records of  0 patients deleted due to incomplete biomarker
measurements.
```

```
> d <- transform(d, list(log10 = log10))
```

This yields a LongData object containing the reduced data set and the transformed biomarker.

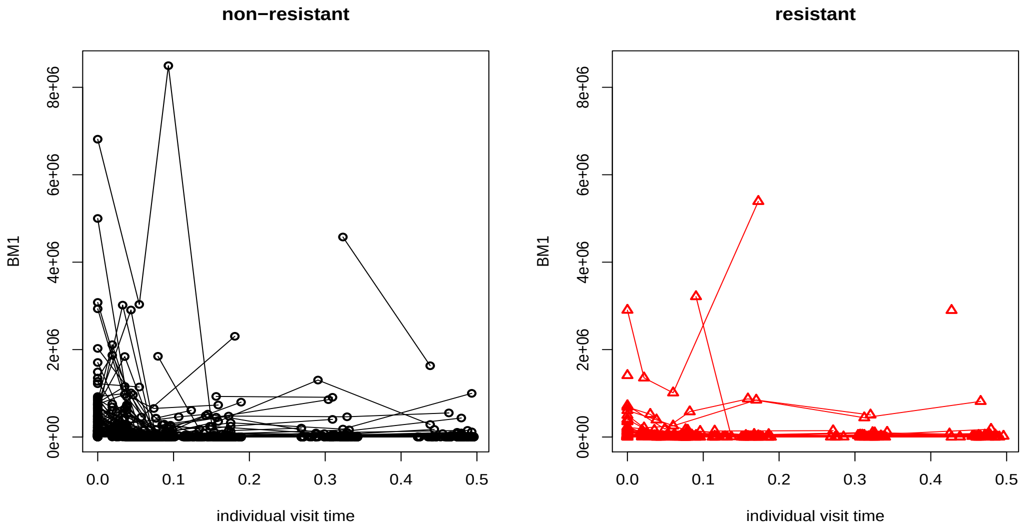


Figure 5.1: Empirical biomarker profiles in each group, produced by plotLongMarkers.

Next, the median profile and its variance are explored by boxplotLongMarkers, the variation of the individual visit times around the scheduled ones are illustrated by scatterVisitTimes and the autocorrelations within each group are estimated by autoCorr. The user has two options here: Either the functions are separately executed (with the advantage to control all parameters individually) or standardized output is created by calling createReportFiles. In the latter case, the four functions are executed with predefined parameter settings, the output files are saved and the location of the output files are automatically added in the Report object, updating the object rep1.

This is achieved by

```
> rep1 <- createReportFiles(d, report = rep1)
```

The Report object's current structure is the following:

```
> str(rep1)
```

```
Formal class 'Report' [package "longQDA"] with 2 slots
  ..@ .Data :List of 4
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data :List of 1
  .. .. .. .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. .. .. .. ..@ .Data :List of 6
  .. .. .. .. .. .. .. ..$ : chr "C:/AIDS2/AIDS2(completecases_6v)plotLongMarkers.pdf"
  .. .. .. .. .. .. .. ..$ : chr "C:/AIDS2/AIDS2(completecases_6v)boxplotLongMarkers.pdf"
  .. .. .. .. .. .. .. ..$ : chr "C:/AIDS2/AIDS2(completecases_6v)scatterVisitTimes.pdf"
  .. .. .. .. .. .. .. ..$ : chr "C:/AIDS2/AIDS2(completecases_6v)statsVisitTimes.tex"
  .. .. .. .. .. .. .. ..$ : chr "C:/AIDS2/AIDS2(completecases_6v)autoCorr.tex"
  .. .. .. .. .. .. .. ..$ : chr "C:/AIDS2/AIDS2(completecases_6v)autoCorrPlots.pdf"
  .. .. .. .. .. .. ..@ myname: chr "AIDS2(complete cases,6v)"
  .. .. .. ..@ myname: chr "Exploratory Analysis"
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data : list()
  .. .. .. ..@ myname: chr "Analysis Setup"
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data : list()
  .. .. .. ..@ myname: chr "Analysis"
  .. ..$ :Formal class 'NamedList' [package "longQDA"] with 2 slots
  .. .. .. ..@ .Data : list()
  .. .. .. ..@ myname: chr "Comparisons"
  ..@ folder: chr "C:/AIDS2/"
```

In most cases, it is the best to combine both options. Suppose additional plots of the biomarker profiles are needed, in this case with transparent lines according to an alpha transparency level of 0.3. The first option may be used for this purpose by calling the function manually. This yields Figure 5.2.

```
> par(mfrow = c(1, 2))
> plotLongMarkers(d, linealpha = 0.3)
```

At every time during the analysis, plots which are displayed in the current graphics device may be included in the report. For this plot, it is achieved by

```
rep1 <- save2Report(rep1, section="expl",
                        subsection="AIDS2 (complete cases)",
                        name="PlotTrajTransparency",
                        text="Here are the biomarker trajectories
                        with a transparency value of $0.3$.")
```

The plot as well as the comment given by the argument `text` are added to the subsection 'AIDS2 (complete cases)'[1] in the report section for the exploratory analysis for which the results are saved in the `rep1` list entry `expl`.

[1] The subsection to which the output is added need to exist already when using `save2Report`.

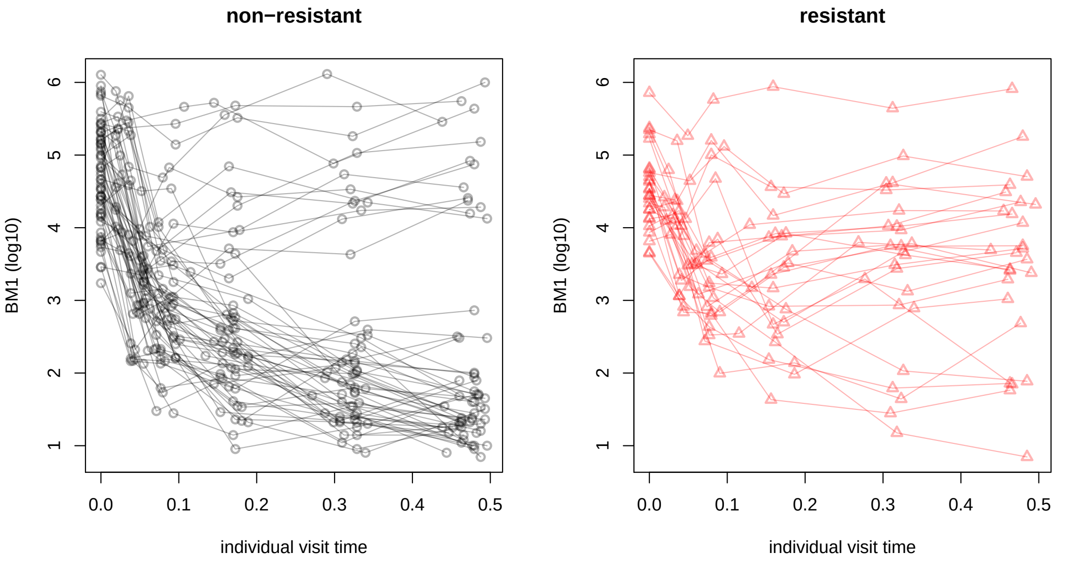


Figure 5.2: Empirical transformed biomarker profiles in each group with transparent lines, produced by plotLongMarkers.

This optional inclusion of any plot in the report may also be used for further explorative analyses when plots have been produced by R functions implemented in other packages or written by the user.

### 5.1.4 Analysis Setup

We move on to set up the discriminant analyses using the constructor function AnalysisSetup. We specify to use 10-fold MCCV with training sets containing twice as many patients as the test sets. The sub-analyses to be performed are specified in the argument paths as lists with a specific structure.

```
> ana.setup <- AnalysisSetup(d, nMCsamples=10, ratiotrain=2/3,
+                      paths=list(list(isLongitudinal=FALSE,
+                                      isMarkerComb=FALSE,
+                                      whichVisits=1,
+                                      isSingleVisit=TRUE),
+                                 list(isLongitudinal=TRUE,
+                                      isMarkerComb=FALSE,
+                                      whichVisits=(1:ncol(d@id2rows)),
+                                      modelparams=list(covstr="RIS",
+                                                       timestr="quadratic"))))
```

In the example above, the biomarker performance of the first as well as that of all 6 visits are of interest. So two analysis paths were defined: Path 1 contains the settings for evaluating the performance at baseline (1st visit) by QDA, path 2 describes the settings for longQDA with linear mixed models including random intercepts and random linear slopes (RIS), and up to quadratic fixed time effects.

One analysis path represents one sub-analysis and all analysis paths together form an analysis tree. E.g. for a data set with 2 biomarkers (denoted as BM1 and BM2) which were measured at 3 visits, a tree of maximal size is given in Figure 5.3. The paths are described by the boolean variable `isLongitudinal` to indicate whether the longitudinal structure of the data should be accounted for (i.e. choosing QDA or longQDA), by `isMarkerComb` to decide for a univariate or multivariate biomarker assessment and by `isSingleVisit` to indicate whether QDA should be executed with single cross-sectional or longitudinal data handling them as multivariate cross-sectional data. The other parameters of the list entries in `paths` determine the visits to be included in the sub-analysis and, if applicable, the covariance structure and the order of the fixed effect of the time variable for the mixed models in univariate longQDA.

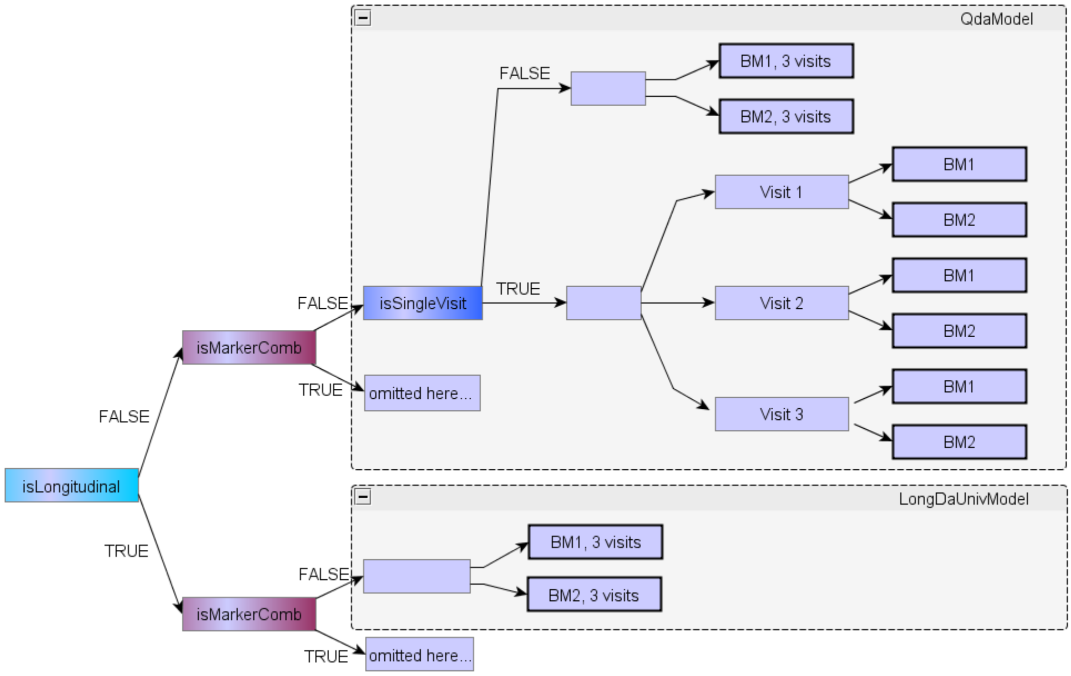


Figure 5.3: Analysis tree illustrating the setup of univariate analysis paths. Multivariate paths where `isMarkerComb` = `TRUE` are omitted. Created with yEd Graph Editor (yWorks, 2008).

The MCCV samples are then generated by calling `AnalysisSetup` and the analysis tree is fully described. These analysis settings are saved to the report by

```
> rep1 <- createReportFiles(ana.setup, report = rep1)
```

To facilitate reproducible research, the analysis setup should be saved as a `R` object, e.g. to the report folder, by

```
> saveRobject(ana.setup, reportfolder=rep1@folder,
              name="AnaSetup_longQDA6visitslin")
```

This enables the user to reload these analysis settings by calling `loadRobject` and to reproduce all analysis results at a later point in time. This functionality is also helpful when the user needs to add further analysis paths. This way, a full comparability of the results is ensured as the analysis is executed with exactly the same data set, same MCCV samples and same performance measures parameters but different (long)QDA settings. The old settings are reloaded by `loadRobject` and only the paths are updated when calling `updateAnalysisSetup` to yield the new analysis setup. All (long)QDA settings that are specified in the argument `paths` of the constructor function `AnalysisSetup` may be changed in the same-named argument of `updateAnalysisSetup`.

### 5.1.5 Analysis of all Analysis Paths

The method analyze is called to start the analysis which comprises the estimation of the mean vector and the covariance matrix for each group (by mixed models for longitudinal data, by the empirical analogues for QDA), the prediction of the posterior probabilities based on the quadratic discriminant rule and the evaluation of the performance measures for each analysis path.

```
> full.tree <- analyze(ana.setup)

  Path 1
     fit of QDA .........

  Path 2
     fit of RIS .........
```

The structure of the tree, containing all analysis results, has 4 hierarchical levels. The first one comprises the analysis paths, the second the visits, the third the biomarker (combinations) and the forth level a list with entries for each MCCV sample. The structure of the second path of `full.tree` is shown below:

```
 > full.tree[[2]]

 $`Visit(s) 1;2;3;4;5;6`
 $`Visit(s) 1;2;3;4;5;6`$`Markers BM1`
 $`Visit(s) 1;2;3;4;5;6`$`Markers BM1`$`CV 1`
 [...]
```

It is organized so that the results of each MCCV sample are contained in the leaf nodes of the tree. One leaf node comprises a list with estimations of each part of `analyze` described above and are saved in the list entries `model`, `predict` and `performance`:

```
> str(full.tree[[2]][[1]][[1]][[1]])

List of 3
 $ model      :Formal class 'RisModel' [package "longQDA"] with 5 slots
  .. ..@ mean       :List of 2
  .. .. ..$ group1: Named num [1:3]   4.48 -13.08  18.92
  .. .. .. ..- attr(*, "names")= chr [1:3] "(Intercept)" "SDURN" "I(SDURN^2)"
  .. .. ..$ group2: Named num [1:3]  4.40 -8.46 14.01
  .. .. .. ..- attr(*, "names")= chr [1:3] "(Intercept)" "SDURN" "I(SDURN^2)"
  .. ..@ cov        :List of 2
  .. .. ..$ group1:List of 2
  .. .. .. ..$ D: Named num [1:3] 0.558 7.378 0.447
  .. .. .. .. ..- attr(*, "names")= chr [1:3] "DInt" "DSlope" "D_IS"
  .. .. .. ..$ R: Named num 0.321
  .. .. .. .. ..- attr(*, "names")= chr "sigQuad"
  .. .. ..$ group2:List of 2
  .. .. .. ..$ D: Named num [1:3] 0.343 6.279 0.477
  .. .. .. .. ..- attr(*, "names")= chr [1:3] "DInt" "DSlope" "D_IS"
  .. .. .. ..$ R: Named num 0.288
  .. .. .. .. ..- attr(*, "names")= chr "sigQuad"
  .. ..@ BIC        :List of 2
  .. .. ..$ group1: Named num 602
  .. .. .. ..- attr(*, "names")= chr "CV1"
  .. .. ..$ group2: Named num 262
  .. .. .. ..- attr(*, "names")= chr "CV1"
  .. ..@ modelparams:List of 2
  .. .. ..$ covstr : chr "RIS"
  .. .. ..$ timestr: chr "quadratic"
  .. ..@ reshapefn  : chr "minimize"
 $ predict    :List of 3
  ..$ class      : Factor w/ 2 levels "1","2": 1 1 2 1 2 2 1 2 1 1 ...
  ..$ posterior  : num [1:27, 1:2] 0.864 0.962 0.357 0.818 0.377 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:27] "5" "88" "102" "104" ...
  .. .. ..$ : chr [1:2] "1" "2"
  ..$ posteriorMV: num [1:27, 1:2] 0.004356 0.000370 0.000214 0.003271 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:27] "5" "88" "102" "104" ...
  .. .. ..$ : chr [1:2] "1" "2"
 $ performance:Formal class 'PerformanceMeasures' [package "longQDA"]
               with 13 slots
  .. ..@ auc              : num 0.865
  .. ..@ roc              :Formal class 'performance' [package "ROCR"]
                            with 6 slots
  .. .. .. ..@ x.name      : chr "False positive rate"
  .. .. .. ..@ y.name      : chr "True positive rate"
  .. .. .. ..@ alpha.name  : chr "Cutoff"
  .. .. .. ..@ x.values    :List of 1
  .. .. .. .. ..$ : num [1:101] 0 0 0 0 0 0 0 0 0 0 ...
  .. .. .. ..@ y.values    :List of 1
  .. .. .. .. ..$ : num [1:101] 0 0 0 0 0 0 0 0 0 0 ...
  .. .. .. ..@ alpha.values:List of 1
  .. .. .. .. ..$ : num [1:101] 1.00 0.99 0.98 0.97 0.96 ...
  .. ..@ bs               : num 0.87
  .. ..@ bsC              : num 0.0237
  .. ..@ bsD              : num 0.100
  .. ..@ bsDM             : num 0.112
  .. ..@ diffpig          : num 0.332
  .. ..@ pig2quer         : num 0.559
  .. ..@ corGSBM          : num 0.618
  .. ..@ deltaSigmaPostProb: num 0.196
  .. ..@ sigmaMinPostProb  : num 0.0234
  .. ..@ ratioVarPostProb  : num 8.4
  .. ..@ calLarge          : num 0.000395
```

Especially in the case of long computation times, it is recommended to save this object as the analysis settings before by `saveRobject`. This enables the user to analyze the results further, also in another R session later on.

```
> saveRobject(full.tree, reportfolder=rep1@folder,
+             name="fullTree_longQDA6visitslin")
```

The user is probably not interested in the results of a single MCCV loop but in the summarized results of all MCCV samples. Therefore the method `mccvSummary` returns a result tree where the leaf nodes contain summaries in form of matrices, lists etc. E.g. for the second path, the result looks as follows:

```
> sum.tree <- mccvSummary(full.tree, ana.setup)
> str(sum.tree[[2]], 6)

List of 1
 $ Visit(s) 1;2;3;4;5;6:List of 1
  ..$ Markers BM1:Formal class 'MccvSummary' [package "longQDA"] with 5 slots
  .. .. ..@ modelEstimates:List of 2
  .. .. .. ..$ group1:List of 3
  .. .. .. .. ..$ estimates  : num [1:10, 1:7] 4.48 4.56 4.52 4.42 4.52 ...
  .. .. .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ BIC        : Named num [1:10] 602 597 590 606 605 ...
  .. .. .. .. .. ..- attr(*, "names")= chr [1:10] "CV1" "CV2" "CV3" "CV4" ...
  .. .. .. .. ..$ modelparams:List of 2
  .. .. .. ..$ group2:List of 3
  .. .. .. .. ..$ estimates  : num [1:10, 1:7] 4.40 4.29 4.39 4.38 4.32 ...
  .. .. .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ BIC        : Named num [1:10] 262 238 250 261 251 ...
  .. .. .. .. .. ..- attr(*, "names")= chr [1:10] "CV1" "CV2" "CV3" "CV4" ...
  .. .. .. .. ..$ modelparams:List of 2
  .. .. ..@ roccv       :Formal class 'performance' [package "ROCR"] with 6 slots
  .. .. .. .. ..@ x.name      : chr "False positive rate"
  .. .. .. .. ..@ y.name      : chr "True positive rate"
  .. .. .. .. ..@ alpha.name  : chr "Cutoff"
  .. .. .. .. ..@ x.values    :List of 10
  .. .. .. .. ..@ y.values    :List of 10
  .. .. .. .. ..@ alpha.values:List of 10
  .. .. ..@ pmmatrix    : num [1:10, 1:12] 0.865 0.569 0.612 0.763 0.789 ...
  .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ : chr [1:10] "CV 1" "CV 2" "CV 3" "CV 4" ...
  .. .. .. .. ..$ : chr [1:12] "auc" "bs" "bsC" "bsD" ...
  .. .. ..@ posterior2  :'data.frame':    85 obs. of  11 variables:
  .. .. .. ..$ group: int [1:85] 1 1 1 1 1 1 1 2 1 1 ...
  .. .. .. ..$ 1    : num [1:85]    NA 0.136    NA    NA    NA ...
  .. .. .. ..$ 2    : num [1:85]     NA 0.0444     NA     NA 0.3387 ...
  .. .. .. ..$ 3    : num [1:85] NA NA NA NA NA ...
  .. .. .. ..$ 4    : num [1:85]    NA 0.104    NA    NA 0.296 ...
  .. .. .. ..$ 5    : num [1:85]    NA 0.119 0.212    NA 0.354 ...
  .. .. .. ..$ 6    : num [1:85] NA NA NA NA NA ...
  .. .. .. ..$ 7    : num [1:85]    NA    NA    NA    NA 0.206 ...
  .. .. .. ..$ 8    : num [1:85]    NA    NA 0.174    NA    NA ...
  .. .. .. ..$ 9    : num [1:85]     NA     NA     NA 0.0798     NA ...
  .. .. .. ..$ 10   : num [1:85] NA NA NA NA NA NA NA NA NA NA ...
  .. .. ..@ analysisPath: chr "Path 2_Visit(s) 1;2;3;4;5;6_Markers BM1"
```

Based on this summary, default output files for each path are generated by calling `createReportFiles`. This output includes the group-specific estimated model parameters, a table with quantiles of the performance measures, ROC curves, histograms of the performance measures of all MCCV samples, a back-to-back histogram of the predicted posterior probabilities and a calibration curve for every analysis path.

```
> rep1 <- createReportFiles(sum.tree, ana.setup, report = rep1,
+     ROCaveragefn = "mean")
```

By default, all output of this section is summarized in a temporary pdf file. This preview, named 'myReport.pdf', can be found in the report folder. As above, it is also possible to individualize the output by calling the internal output methods separately. Vide the documentation (`?createReportFiles`) for all methods which are provided for this purpose.

Comments can be added to the report to a section resp. subsection. Further on, comments can be saved together with a single plot as shown above. Below, the first option is demonstrated by adding a comment to the longQDA subanalysis (path 2):

```
> rep1 <- save2Report(rep1, section = "analysis", subsection = 2,
+     text = "longQDA includes 6 biomarker measurements per patient.")
```

### 5.1.6 Comparison of Analysis Paths

At the end of the exemplary data analysis, the results of all subanalyses are compared. The subanalyses are selected by their leaf numbers which are returned by

```
> showLeafs(sum.tree)

Path 1
     Visit(s) 1
         Markers BM1  =>  1
Path 2
     Visit(s) 1;2;3;4;5;6
         Markers BM1  =>  2
```

As there are only 2 subanalyses, both of them are selected. A name for the comparison as well as names for the analyses described by the analysis paths need to be provided.

```
> leafs6 <- selectResultLeafs(sum.tree, leafnumbers=(1:2),
+                          compname="Baseline vs. all 6 visits",
+                          pathnames=c("QDA, Baseline", "RIS, 6 vis."))
```

In addition, output files are generated by `createReportFiles` giving 2 plots with 2 ROC curves for our example. One ROC curve is coloured according to the changing threshold of the posterior probability, the other according to the path-specific colour (alterable by the argument `ROCcolors`) and contains the legend for the ROC curves as well. In addition, boxplots of the performance measures are plotted. A table with the frequencies of the covariance structures (RI, RIS and RICAR1) which achieved the minimal BIC makes no sense in this example, so `freqMinBIC=FALSE` is set.

```
> rep1 <- createReportFiles(leafs6, ana.setup, rep1,
+                            ROCcolors=c("firebrick3","midnightblue"),
+                            legendtext=c("RIS, BM1", "QDA, BM1"),
+                            boxcolors=c("firebrick3", "midnightblue"),
+                            freqMinBIC=FALSE)
```

### 5.1.7 Report Generation

Finally, a report file is created (standard name is 'myreport.tex') where all output plots and tables stored in the report object rep1 are included. The report is structured in sections and subsections as defined by the list structures in rep1. A name for the data set on which the analysis was based and details about the author of the report are provided. Then createReport generates a report file in tex format which is stored in the subfolder 'Report'. The full path of this file is returned by this method.

```
> createReport(rep1, dataname=ana.setup@data@name,
+   author="Mareike Kohlmann \\\\ DXRQB2 \\\\ Roche Diagnostics GmbH")

[1] "C:/AIDS2/Report/myReport.tex"
```

If necessary, the report can then be edited manually by adding comments or plots, rearranging or deleting output files etc. A tex compiler such as pdflatex is necessary to compile the tex file to pdf. The resulting report file for this exemplary analysis is included as Appendix A.

## 5.2 Software Design

The software for performing longQDA was implemented to provide a software solution for the statistical evaluation of longitudinal biomarkers in clinical studies at Roche Diagnostics. The conceptual requirements are therefore a data-independent implementation with a user-friendly handling, easy extensibility and a good run-time performance. A further requirement is to support a comfortable reporting of the analysis results. With this implementation, the user may apply univariate longQDA, multivariate longQDA and multivariate QDA to real as well as to simulated data. All classification analyses are based on the resampling method Monte Carlo cross validation to ensure a reliable estimation for small sample sizes as well.

To meet all these needs, an object-oriented design was chosen which is facilitated by the S4 system in R. This system provides functionality that allows OOP with classes and methods within the functional R language. The idea of OOP is a close mapping from the reality to the programming in a modular fashion. A class, the main component of this concept, defines a new data type comprising a specific set of attributes called slots. E.g. for a class Data, slots may be the name of the data set, the column number of the patient identification number etc. Objects are generated as instances of a class. Computations on these objects consist of invoking methods on them. A method is specifically defined for one class and therefore objects are part of the method's arguments and are frequently returned as well. Exemplary methods are `show` or `plot` which determine the class-individual printing resp. plotting. Methods need to be declared as generic functions in R. Via this mechanism, S4 is compatible with the functional requirements of R.

Further, classes may be related to each other by class hierarchies, i.e. classes inherit class properties of the base class and are extended by specific properties. In `longQDA`, for example, the classes `LongData` for longitudinal data and `XsecData` for cross-sectional data are derived classes of the base class `Data`. Both are designed for objects which include a named data set and, for example, a column for the patient identification number, but `LongData` has its own attribute `id2rows` which is a matrix matching the row numbers to each patient's observations. Objects of class `XsecData` do not need this attribute as there is always only one row per patient in cross-sectional data.

Classes can be abstract, i.e. the class cannot be instantiated. The opposite is a concrete class. Abstract classes are designed as base class and the specific functionality is then implemented in the derived classes. The class hierarchy supports the extensibility as other data structures can be easily incorporated. A user-friendly handling, an easy maintenance and a fast debugging are further advantages of OOP due to the sparse representation of complex relations. For example, methods are invoked on objects corresponding to the classes of the method's arguments. Thus, if-statements in methods are avoided to direct the proper, class-specific execution.

The user-friendly handling is furthermore enhanced by an intuitive guidance following the typical process of a statistical analysis: first the explorative and descriptive analysis of the longitudinal data, followed by the evaluation and comparison of the biomarker performance under various model assumptions by longQDA. Up to a basic level, these tasks can be performed with the knowledge of only few central methods which are `LongData`, `AnalysisSetup`, `analyze`, `MccvSummary`, `selectResultLeafs`, `Report` and `createReportFiles`. The corresponding documentation is given in Appendix B, the relations between the functions and a more detailed description follow in the next subsection which is 5.2.1.

Throughout the entire analysis, a template for the report can be created continuously. At some points, namely for the descriptive and exploratory analysis and for the comparison of the models, the user can select either the predefined default analysis or the customized analysis or can even combine both options. The first is accomplished by `createReportFiles`, the latter by using directly the methods included in `createReportFiles`, these are ,for example, `plotLongMarkers`, `boxplotLongMarkers`, `scatterVisitTimes`, `autoCorr` and optionally self-written functions or methods to explore the longitudinal data. A combination of both options is used in Section 5.1 where `plotLongMarkers` has been used with another transparency value in addition to the default output. The report can be complemented by comments or any plot during the analysis. The output format is a tex file which can be compiled to pdf, allowing an easy editing of the template by adding text or rearranging parts after the completion of the data analysis.

### 5.2.1 Software Architecture Overview

In the following, the presentation of the software architecture of longQDA is restricted to the components that are needed for the analysis of real, univariate biomarker data sets. The other components are described in Section 5.3 where the extensibility for the evaluation of multivariate biomarker panels as well as for the analysis of simulated data are demonstrated.

Since 1997, unified modelling language (UML) has been the standard approach for a structured object-oriented software development (Born et al., 2004). One of the key diagrams of UML is the class diagram depicting the class structure with its relationships. At the moment, R does not support the creation of class diagrams from the code (backward engineering), but there are plans to incorporate such a functionality in the future, called Ruml (R Foundation for Statistical Computing, 2008). First steps towards backward engineering are done by the package classGraph (Maechler and Gentleman, 2008). A class diagram includes usually the classes with their attributes and methods as well as the relationships between the classes. For the sake of a condensed presentation, the class-specific attributes and methods are omitted in the class diagram for longQDA (Figure 5.4). They are described exemplarily in the next subsection. Two sorts of class relationships are visualized: Every arrow joins the derived class with its base class (e.g. LongData and XsecData inherit from Data), a line labelled with its description symbolises an undirected relationship (the method toXsec converts an instance of class LongData to an instance of class XsecData with a cross-sectional data structure, for example). Note that only the most important relations are included. Abstract classes are printed in italics.

The software architecture reflects the steps of a typical statistical analysis and is hence quite general and could be easily adopted for other software implementations: There are classes for data objects (on the top right of Figure 5.4), for the analysis setup (in the middle on the top), classes involved in the discriminant analyses (down right), classes containing the raw or summarized results of the analyses (on the left) and the Report class defining a standardized output (down left). We continue by presenting the class diagram, following that order as far as possible, and start with the initial step, the data import. In the upper right corner, all classes for structural data mapping are shown. They have already been partly described above to explain possible class relationships. The user operates only on objects of class LongData which are created from the user-provided raw data sets and which are used to set up the analysis by AnalysisSetup. Objects of class

XsecData are used only internally for tasks that require a cross-section data structure, e.g. for performing QDA or creating plots (in autoCorr, for example).

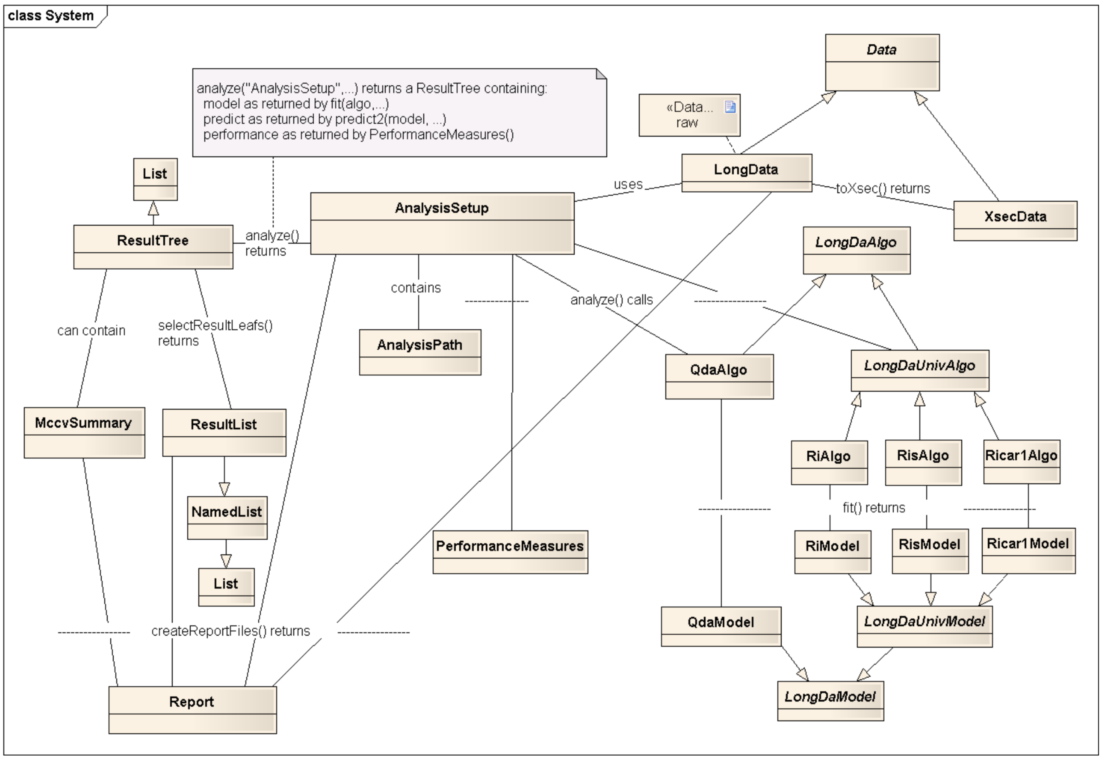


Figure 5.4: UML class diagram of the R package longQDA, omitting attributes and methods. Created with Enterprise Architect (SparksSystems, 2008)

Objects of class AnalysisSetup specify all subanalyses that should be performed, including information about the MCCV design and global precision parameters. Based on the paths attribute of AnalysisSetup describing the subanalyses, AnalysisPath objects are internally set up. These objects determine the instantiation of objects of class QdaAlgo (for QDA) or LongDaUnivAlgo (for the univariate longQDA) during the execution of the method analyze. Besides the determination of the required data structure for the algorithms, the statistical analyses are performed as follows when calling analyze.

The evaluation of the biomarker performance is split up into three steps: the estimation of the group-specific parameters for the quadratic discriminant rule with the training data sets, the prediction of the posterior probabilities by (long)QDA with the test sets and at last, the evaluation of the performance measures. The first step is accomplished by the method fit, defined for the classes determining the algorithms for the discriminant analysis. In the case of longQDA, for example, objects of class RiAlgo, RisAlgo or Ricar1Algo define the estimation of the means and the covariance matrices by mixed models. For QDA, the functionality of the R package MASS (Venables et al., 2008) is used. For the univariate longQDA with an RI or RIS structure, the estimation is done by lme4 (Bates et al., 2008) whereas nlme (Pinheiro et al., 2008) is used for the RICAR1 structure[2]. For the second step, the evaluation of the discriminant rule, predict2 is called. The method is defined for objects of class LongDaUnivModel and returns a list containing the estimated posterior probabilities amongst others. The classes RiModel, RisModel and Ricar1Model are derived classes of LongDaUnivModel which is in turn, as QdaModel, a derived class of LongDaModel. The same structure underlies the classes describing the corresponding algorithms. The third step involves the calculation of the performance measures by the constructor PerformanceMeasures. It uses the returned list of predict2 as input and returns an object of class PerformanceMeasures. The returned values of those three methods are stored in an instance of class ResultTree which is created within analyze.

These results are not yet summarized, they contain the results for each MCCV sample. The method mccvSummary accomplishes this task by modifying the object of class ResultTree to consist of objects of class MccvSummary. To constrain the results for comparison, the method selectResultLeafs returns a smaller version of class ResultList.

It is recommended to create an instance of class Report at the beginning of a data analysis session. At the end of the analysis, the main parts of the report comprise the output which was generated by the method createReportFiles (for objects of class LongData, AnalysisSetup, MccvSummary or ResultList) or by the method save2Report for more individualized output.

### 5.2.2 Description of Central Classes and Methods

The architecture of longQDA at the class level comprises the classes with its attributes and methods. As already explained, they are usually part of the UML class diagram. In R however, this issue is not realised graphically but at least textually by the documentation of the classes in the help files. For S4 classes, the attributes of a class are stored in so called slots of an object. The slots as well as the methods are documented for each class.

As an example, the help file for the class MccvSummary is included here. Except the method show, all methods are documented separately. The method createReportFiles is given and serves furthermore as an example for the sparse but flexible OOP approach. It is not only defined for objects of class MccvSummary but also for objects of class LongData, AnalysisSetup and ResultList. For further class and method descriptions, the reader is referred to the help files in Appendix B.

[2] According to the author of lme4, this package is superior to nlme with regards to its execution time and stability. But as models with an RICAR1 structure are only available in nlme, both packages had to be used in longQDA.

## MccvSummary-class *Class 'MccvSummary' with Summarized Analysis Results of all MCCV samples*

### Description

All important analysis results are summarized over all MCCV loops allowing to produce graphical as well as numerical summaries of the results by the provided methods.

### Objects from the Class

Objects are created by calls of the form `MccvSummary(nMCsamples, pmnames, cvnames, roctemp, id2group)`, but these objects are only created internally. Therefore the arguments are not documented here.

### Slots

`modelEstimates`: Object of class `'list'`, the length equals to the number of groups. Each list entry contains

- `estimates`: Object of class `'matrix'` containing the model parameters estimated by MCCV. Each row correspondes to one MCCV sample (with names `"CV 1"`, `"CV 2"`, ...), each column to one of the model parameters. The model parameters have names and their number differs, depending on the estimated model.
- `BIC`: Object of class `'numeric'` containing the BICs of all MCCV samples (with names `"CV 1"`, `"CV 2"`, ...).
- `modelparams`: Object of class `'list'` containing the following mixed model parameters
  - `covstr`: Object of class `'character'` giving the covariance structure. Can be either `"RI"` (random intercept), `"RIS"` (random intercept and linear slope) or `"RICAR1"` (random intercept and continuous AR(1) residual structure) for univariate longQDA.
  - `timestr`: Object of class `'character'` giving the maximum order for the fixed effect(s) of the time variable. Terms of lower order are automatically included. Can either be `"linear"` or `"quadratic"`.

`roccv`: Object of class `'performance-class'` in package `ROCR`. The slots `x.values`, `y.values` and `alpha.values` are lists, each of length `nMCsamples`; v. the documentation of `ROCR` for details.

`pmmatrix`: Object of class `'matrix'` containing all MCCV estimates of twelve performance measures. Each row corresponds to one MCCV loop (with names `"CV 1"`, `"CV 2"`, ...), each column to one of the performance measures (with names `"auc"`, `"bs"`, ...).

posterior2: Object of class 'data.frame'. Each row corresponds to one patient. The first column contains the numeric group according to the Gold Standard, the second the predicted posterior probabilities for the latter specified group when creating the LongData object of the first MCCV loop, the third the predicted posterior probability of the second MCCV loop and so on. The names of the rows equal to the patient identification numbers, the first column has the name group, the others the number of the MCCV loop.

analysisPath: Object of class 'character' containing the name of the analysis path. It is a concatenation of the path number, the involved visits and the biomarker(s) and will be used for file names of plots and tables, for example.

### Methods

createReportFiles: signature(object = 'MccvSummary'): createReportFiles

plotROC: signature(summary = 'MccvSummary':): plotROC

pmHist: signature(summary = 'MccvSummary'): pmHist

pmQuantiles: signature(summary = 'MccvSummary'): pmQuantiles

post2CalCurve: signature(summary = 'MccvSummary'): post2CalCurve

post2Hist: signature(summary = 'MccvSummary'): post2Hist

show: signature(object = 'MccvSummary'): defined as str(object, 3).

summaryModelEstimates: signature(summary = 'MccvSummary'):
summaryModelEstimates

### Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

### Examples

```
showClass("MccvSummary")

data(AIDS2)
d <- LongData(AIDS2, "AIDS2", c("BM1"), c(0,2,4,8,16,24)/52,
             c('non-resistant'=1, 'resistant'=2), timename="SDURN")
d2 <- completeCases(d, visits=6)
asetup <- AnalysisSetup(d2, nMCsamples=5, ratiotrain=2/3,
                        paths=list(
                               list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                                    whichVisits=(1:ncol(d2@id2rows)),
                                    modelparams=list(covstr="RI",
                                                     timestr="quadratic"))))
tree <- analyze(asetup)
summary.tree <- mccvSummary(tree, asetup)
```

```
node.leaf <- summary.tree[[1]][[1]][[1]]
class(node.leaf)
str(node.leaf,4)
```

---

`createReportFiles-method` *Creates Pre-Configured Output Files for the Report*

---

## Description

At any stage of the analysis, pre-configured output files can be created and saved for a later inclusion into the report. This method calls various other methods for creating graphical and/or numerical output and serves as default output. If modified or additional output is desired, the user has the possibility to call those inner methods separately and to add their graphical output to the report by `save2Report`. There is also an option enabling the user to preview the corresponding part of the report with the generated output files.

The following output generating methods are called within `createReportFiles()`:

**For objects of class `'LongData'`:** `plotLongMarkers`,`boxplotLongMarkers`, `scatterVisitTimes`, `autoCorr`.

**For objects of class `'AnalysisSetup'`:** `str`.

**For objects of class `'ResultTree'`:** `createReportFiles` for each leaf of class `'MccvSummary'`.

**For objects of class `'MccvSummary'`:** `summaryModelEstimates`, `pmQuantiles`, `plotROC`, `pmHist`, `post2Hist`, `post2CalCurve`.

**For objects of class `'ResultList'`:** `plotROC`, `compPerfMeasures`, `freqMinBIC`, `compPostProbs`, `compModelEstimates`.

The execution of some of the methods can be controlled by the user, e.g. by setting `plotpmHist=FALSE` for objects of class `'MccvSummary'` or `'ResultTree'`. Some methods are automatically only executed for simulated data, e.g. `compPostProbs` for objects of class `'ResultList'`.

## Usage

**For objects of class `'LongData'`:**

```
createReportFiles(object, analysis=object, report,
includeVisitTimes=TRUE, compile=TRUE)
```

**For objects of class `'AnalysisSetup'`:**

```
createReportFiles(object, analysis=object, report, compile=TRUE)
```

**For objects of class** ’MccvSummary’:

```
createReportFiles(object, analysis, report, MEaveragefn="mean",
MErounddigits=3, ROCaveragefn="mean", plotpmHist=TRUE,
pmrange=NULL,
PMrounddigits=c(2,3,3,2,2,2,2,2,2,3,0,3,0),
pmnames=c("$\mbox{AUC}$","$\mbox{BS}$", "$\mbox{BS}_C$",
"$\mbox{BS}_D$", "$\mbox{BS}_{D_M}$",
"$\bar{pp}_{(2)[z_{(2)}=1]}-\bar{pp}_{(2)[z_{(2)}=0]}$",
"$\bar{pp}_{(2)[z_{(2)}=1]}$", "$\rho_{{z_{(2)}},pp_{(2)}}$",
"$\Delta\sigma^2_{pp_{(2)}}$","$\sigma^2_{pp_{(2)},min}$",
"$\Delta\sigma^2_{pp_{(2)}}/\sigma^2_{pp_{(2)},min}$",
"$\mbox{Cal}_L$","$\mbox{Sensitivity}$"), onlyROCaverage=FALSE)
```

**For objects of class** ’ResultTree’:

```
createReportFiles(object, analysis, report, MEaveragefn="mean",
MErounddigits=3, ROCaveragefn="mean", plotpmHist=TRUE,
pmrange=NULL,
PMrounddigits=c(2,3,3,2,2,2,2,2,2,3,0,3,0),
pmnames=c("$\mbox{AUC}$","$\mbox{BS}$", "$\mbox{BS}_C$",
"$\mbox{BS}_D$", "$\mbox{BS}_{D_M}$",
"$\bar{pp}_{(2)[z_{(2)}=1]}-\bar{pp}_{(2)[z_{(2)}=0]}$",
"$\bar{pp}_{(2)[z_{(2)}=1]}$", "$\rho_{{z_{(2)}},pp_{(2)}}$",
"$\Delta\sigma^2_{pp_{(2)}}$","$\sigma^2_{pp_{(2)},min}$",
"$\Delta\sigma^2_{pp_{(2)}}/\sigma^2_{pp_{(2)},min}$",
"$\mbox{Cal}_L$","$\mbox{Sensitivity}$"), onlyROCaverage=FALSE,
compile=TRUE)
```

**For objects of class** ’ResultList’:

```
createReportFiles(object, analysis, report, ROCaveragefn="mean",
ROCcolors, legendtext, boxcolors, PMlabels=NULL, xTickLabels=NULL,
freqMinBIC=TRUE, existStarModel=FALSE, PPaveragefn="mean",
PPlinealpha=1, PPyliml=c(0,1), PPylimb=c(-60,100), MEbiaslimits=list(),
PProunddigits=0, compile=TRUE)
```

**For objects of class** ’SimulationSetup’:

```
createReportFiles(object, analysis=object, report, compile=TRUE)
```

**For objects of class** ’SimulationResult’:

```
createReportFiles(object, analysis=object@analysisSetups[[1]],
report, ROCaveragefn="mean", plotpmHist=FALSE, compile=FALSE,...)
```

## Arguments

`object = ’LongData’`: Contains the data set and corresponding metainformation.

`object = ’AnalysisSetup’`: Contains metainformation about the analysis.

`object = ’MccvSummary’`: Contains the summary of results over all MCCV samples.

`object = ’ResultTree’`: Contains all analysis results based on real data.

`object = ’ResultList’`: Contains user-selected analysis results.

`object = 'SimulationSetup'`: Contains metainformation about the simulation.

`object = 'SimulationResult'`: Contains all analysis results based on simulated data.

`analysis`: Default: `object` for `'LongData'` (actually not necessary - just included for compatibility) and `'AnalysisSetup'`. Object of class `'AnalysisSetup'` containing metainformation about the analysis.

`report`: Object of class `'Report'` to which the output files are added.

`compile`: Default: `TRUE`; exeption: `FALSE` for objects of class `SimulationResult`. If `TRUE`, the corresponding part of the report is generated as a preview file in the folder "Report" with name "myreport.pdf". This option does not exist for objects of class `'MccvSummary'`.

**Additionally for objects of class `'LongData'`:**

`includeVisitTimes`: Default: `TRUE`. If `FALSE`, `scatterVisitTimes` is not executed.

**Additionally for objects of class `'MccvSummary'` or `'ResultTree'`:**

`MEaveragefn`: Default: `"mean"`. Character vector giving the function for summarizing the model estimates of all MCCV samples.

`MErounddigits`: Default: `3`. Numerical number of decimal places to which the model estimates are rounded.

`ROCaveragefn`: Default: `"mean"`. Character vector giving the function for summarizing the ROC curves of all MCCV samples.

`plotpmHist`: Default: `TRUE`. If `FALSE`, `pmHist` is not executed.

`pmrange`: Default: `NULL`. Matrix containing the ranges for the histograms of the performance measures. Each column has two rows and contains the range of one measure. If it is not provided by the user (default), the ranges of the performance measures present in the data are taken.

`PMrounddigits`: Default: `c(2,3,3,2,2,2,2,2,2,3,0,3,0)`. Numeric vector of decimal places to which the performance measures are rounded.

`pmnames`: Default:
```
c("$\mbox{AUC}$","$\mbox{BS}$", "$\mbox{BS}_C$",
"$\mbox{BS}_D$", "$\mbox{BS}_{D_M}$",
"$\bar{pp}_{(2)[z_{(2)}=1]}-\bar{pp}_{(2)[z_{(2)}=0]}$",
"$\bar{pp}_{(2)[z_{(2)}=1]}$", "$\rho_{{z_{(2)}},pp_{(2)}}$",
"$\Delta\sigma^2_{pp_{(2)}}$","$\sigma^2_{pp_{(2)},min}$",
"$\Delta\sigma^2_{pp_{(2)}}/\sigma^2_{pp_{(2)},min}$",
"$\mbox{Cal}_L$","$\mbox{Sensitivity}$").
```
Names of the performance measures, used for labelling the columns of the output table and need to be formatted as LaTeX code.

`onlyROCaverage`: Default: `FALSE`. If `TRUE`, only the ROC curve based on the summary of all MCCV samples is plotted. Otherwise, an additional plot with an ROC curve for every MCCV sample is generated.

**Additionally for objects of class `'ResultList'`:**

ROCaveragefn: Default: "mean". Character vector giving the function for summarizing the ROC curves of all MCCV samples.

ROCcolors: v. colors in plotROC.

legendtext: v. legendtext in plotROC.

boxcolors: v. boxcolors in compPerfMeasures.

PMlabels: Default: NULL. Character vector containing the labels of the performance measures, used in compPerfMeasures.

freqMinBIC: Default: TRUE. Should freqMinBIC be executed or not?

existStarModel: Default: FALSE. If TRUE, the data analysis is based on simulated data, compPostProbs and compModelEstimates are executed to assess the bias.

PPaveragefn: Default: "mean". v. averagefn in compPostProbs.

PPlinealpha: Default: 1. v. linealpha in compPostProbs.

PPyliml: Default: c(0,1). v. yliml in compPostProbs.

PPylimb: Default: c(-60,100). v. ylimb in compPostProbs.

MEbiaslimits: Default: list(). v. boxlimits in compModelEstimates.

PProunddigits: Default: 0. v. rounddigits in compPostProbs.

## Value

Object of class 'Report'. The list entries contain the names and the location of the output files.

## Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

## Examples

```
### Example for object of class "LongData"
myreport <- Report(folder=getwd())

data(AIDS2)
d <- LongData(AIDS2, "AIDS2", c("BM1"), c(0,2,4,8,16,24)/52,
             c('non-resistant'=1, 'resistant'=2), timename="SDURN")

# compile=TRUE does not work in Rcheck
myreport <- createReportFiles(object=d, report=myreport, compile=FALSE)
```

## 5.3 Examples for Extending `longQDA`

Extensibility is an important quality aspect in software development. The package `longQDA` already contains two extensions to the basis functionality for analyzing univariate biomarker data. One is the incorporation of multivariate mixed models to evaluate biomarker panels, the other is the generation and analysis of simulated univariate biomarker profiles. Both serve as examples to demonstrate how fast `longQDA` is extended by few additional classes and how easy the user can cope with the minor visible changes implied.

### 5.3.1 Multivariate longQDA

So far, an implementation has been described for analyzing univariate models. To demonstrate the extensibility, we illustrate in the following how multivariate models are included. The presentation is restricted to multivariate mixed models with correlated random effects as defined in Subsection 4.2.2 as the models with uncorrelated random effects (presented in Section 4.2.1) as well as the Kronecker Product models (Subsection 4.2.3) are analogously implemented in `longQDA`. In general, to extend the software by a new model, a class for the model containing the group-specific estimates and a class for the algorithm containing the necessary information for the estimation of the models need to be defined.

For the present example, this is accomplished by the classes `LongDaMvModel` and `RcMvAlgo`. These classes as well as all other necessary changes, which will be described in the following, were added to the UML class diagram of Figure 5.4 and marked in blue (see Figure 5.5). The class `RcMvAlgo` has the newly defined base class `LongDaMvAlgo` to cover also the other algorithms for the multivariate models. This is in turn a derived class of the already existing `LongDaAlgo`.

For the multivariate mixed models with correlated effects, models with and without random slopes are distinguished, so there are the following classes derived from `LongDaMvModel`: `RiCorrModel` with correlated random intercepts, `RisCorrModel` with correlated random intercepts and slopes[3]. The class `LongDaMvModel` is derived from `LongDaModel` like `QdaModel` and `LongDaUnivModel`.

The estimation of the models with the correlated random effects is carried out by the function `lme` of the `R` package `nlme` with its flexible options to specify the random effects. The function requires a data structure implemented in an additional class, called `LongMvData`, which is derived from `Data`. Objects of this class are internally created by

[3] There are actually additional derived classes which are `RIuModel` and `RIuSuModel`. As they are implemented for models with uncorrelated random effects (Section 4.2.1), further details are omitted here.

the method toLongMv which is part of the class LongData. The data structure comprises a blown-up data set in that there is

- one biomarker vector (called response) where the observations of the single biomarker variables are stacked blockwise
- one dummy variable for each marker indicating which observation of response is from which marker (called marker1, marker2, ...) to model the random intercepts
- one time variable for each marker (called time1, time2, ...) containing the study time if the observation of response is from that marker and otherwise zero to model the random slopes.

The multivariate outcome variables are structured in the same way as if only a univariate variable was analyzed and a set of dummy codes are created to "flag" the outcomes accordingly. This recast has been proposed several times, for R users (Lockwood et al., 2003; Doran and Lockwood, 2006) as well as for SAS users (Thiébaut et al., 2002; Hamlett et al., 2003). For a mixed model with correlated random intercepts and slopes with two biomarkers (specified as covstr="RISFull") in longQDA), the R syntax is

```
lme(response~-1 + marker1 + marker2 + time1 + time2,
    random=~-1 + marker1 + marker2 + time1 + time2|patid,
    data=mydataset, method="REML")
```

and yields an object of class RisCorrModel.

Objects of class RiCorrModel can be obtained by modifying the random statements as follows

```
random=~-1+marker1+marker2|patid
```

Biomarker-specific residual errors are specified by setting

```
weights=varIdent(form=~1|marker2)
```

in the lme statement.

By extending the method analyze to select the new RcMvAlgo if requested by the user, all necessary changes are done. At the user-level, the multivariate mixed model is chosen by setting isMarkerComb=TRUE, giving the number of biomarkers to be combined in markerCombLength and the correlation structure in the list of the model parameters in covstr. The available correlation structures are "RIc" and "RISFull" [4].

[4] There are additional correlation structures implemented in longQDA as a RisCorrModel, these are "RIcSu", "RIuSc" and "RIcSc". It is not recommended to fit models with these correlation structures as the random intercepts and slopes within the biomarker are uncorrelated (cf. Chapter 2, p. 7 for a discussion of this issue). The necessary extensions to fit all possible correlated intercepts and slopes combinations are not available in R due to the restriction that correlation structures for each biomarker can only be combined by pdBlocked in nlme which is used internally in the package longQDA.

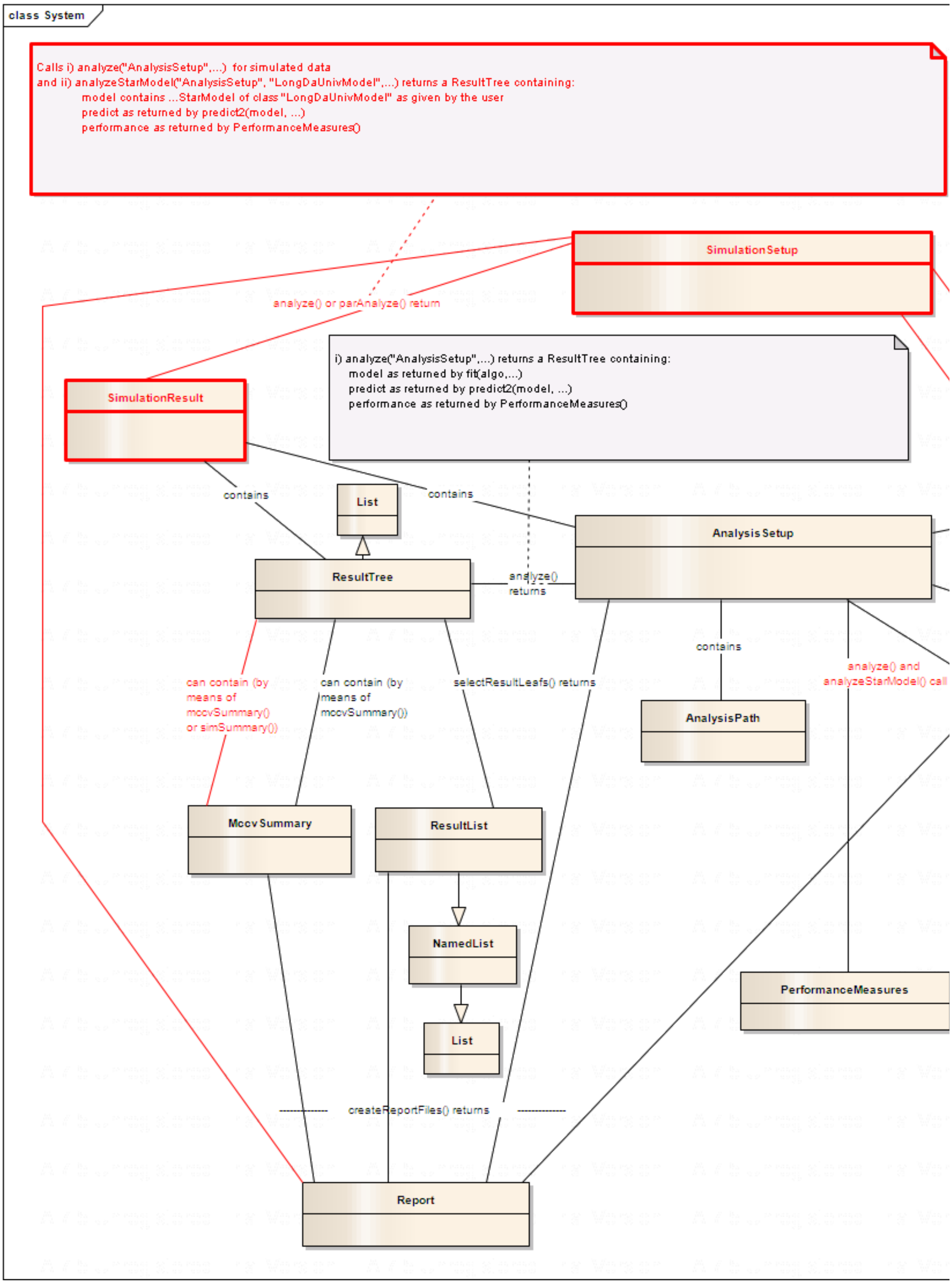
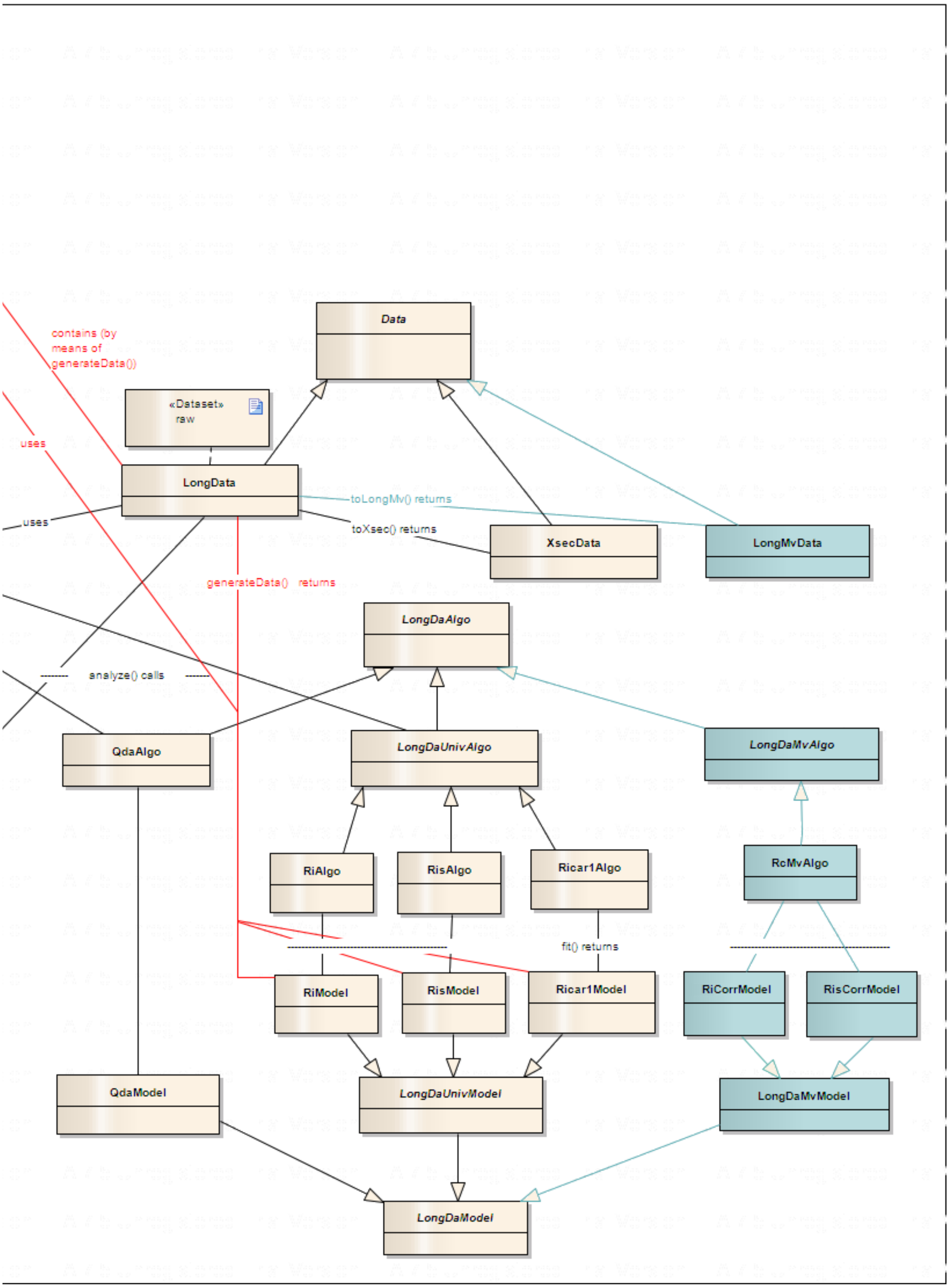

class System
Calls i) analyze("AnalysisSetup",...) for simulated data
and ii) analyzeStarModel("AnalysisSetup", "LongDaUnivModel",...) returns a ResultTree containing:
model contains ...StarModel of class "LongDaUnivModel" as given by the user
predict as returned by predict2(model, ...)
performance as returned by PerformanceMeasures()
SimulationSetup
analyze() or parAnalyze() return
i) analyze("AnalysisSetup",...) returns a ResultTree containing:
model as returned by fit(algo,...)
predict as returned by predict2(model, ...)
performance as returned by PerformanceMeasures()
SimulationResult
contains
List
contains
Analysis Setup
ResultTree
analyze() returns
contains
analyze() and analyzeStarModel() call
AnalysisPath
can contain (by means of mccvSummary() or simSummary())
can contain (by means of mccvSummary())
selectResultLeafs() returns
Mccv Summary
ResultList
NamedList
PerformanceMeasures
List
createReportFiles() returns
Report

Figure 5.5: Extended UML class diagram of the R package longQDA, omitting attributes and methods. The necessary extensions for the multivariate longQDA are marked in blue, those for the simulated data in red.


contains (by means of generateData())
Data
«Dataset» raw
uses
LongData
uses
toLongMv() returns
toXsec() returns
XsecData
LongMvData
generateData() returns
LongDaAlgo
analyze() calls
QdaAlgo
LongDaUnivAlgo
LongDaMvAlgo
RiAlgo
RisAlgo
Ricar1Algo
RcMvAlgo
fit() returns
RiModel
RisModel
Ricar1Model
RiCorrModel
RisCorrModel
QdaModel
LongDaUnivModel
LongDaMvModel
LongDaModel

To update a previously defined analysis setup (called oldSetup) by a bivariate mixed model with correlated random intercepts, the user calls

```
ana.setup1 <- updateAnalysisSetup(oldSetup,
                 paths=list(list(isLongitudinal=TRUE, isMarkerComb=TRUE,
                                 markerCombLength=2,
                                 whichVisits=(1:ncol(data@id2rows)),
                                 modelparams=list(covstr="RIc",
                                                  timestr="quadratic"))))
```

### 5.3.2 Generation and Analysis of Simulated Data

The second example, which shows how easy the functionality of longQDA is extended, comprises all necessary changes to perform the simulation study of Chapter 3. This applies to the following two tasks which need to be performed: the generation of the data and their analysis. As both of them are repeatedly executed, the software design is extended to be also compatible with parallel computing. All additionally defined classes and the new associations are marked in red in Figure 5.5.

The generated data are based on given model parameters for a specified covariance structure as well as on distributions for the individual visit times and they need to comply with the structure of a real data set. For this purpose, the class SimulationSetup has been designed. Within the constructor function SimulationSetup, data sets of class LongData are simulated by the method generateData which takes as input:

- a model of class RiModel, RisModel or Ricar1Model, set up by the user by means of the constructor methods RiStarModel, RisStarModel or Ricar1StarModel
- the distribution functions of the individual visit times (specified in simRandomVisits in the example below)
- and further parameters as the number of visits (nvisits) or the number of patients in each group (ngroups).

The number of simulated data sets corresponds to the number of specified simulation repetitions (simreps). In addition, the paths of the analysis tree to be included in the simulation are specified in the argument paths of SimulationSetup, analogously to the AnalysisSetup for real data.

Here is one example for an exemplary setup of a simulation scenario:

```
# definition of the RIS* model
risStar <- RisStarModel(group1=list(beta=c(4.0,-4.5),
                           DCovstar=matrix(c(0.40, 0.39, 0.39, 16.1),
                                           ncol=2, byrow=FALSE),
                          R=c(sigQuad=0.56)),
                        group2=list(beta=c(4.0,-1.4),
                          DCovstar=matrix(c(0.23, 0.46, 0.46, 15.8),
                                          ncol=2, byrow=FALSE),
                          R=c(sigQuad=0.39)), timestr="linear")

# functions needed for the generation of individual visit times
rmixnorm <- function (nsample, object){
  mu <- object$mu
  sd <- sqrt(object$sig2)
  nj <- rmultinom(n = 1, size = nsample, prob = object$w)
  return(sample(unlist(sapply(seq(along = nj), function(j) rnorm(nj[j],
                             mean = mu[j], sd = sd[j])))))
}
rmixnorm.v1 <- function(n){
  return(rmixnorm(nsample=n, object=list(mu=c(0.05,0.5,0.95),
                                        sig2=c(0.001,0.05,0.001),
                                        w=c(0.12,0.76,0.12))))
}

# [some function definitions are omitted here...]

rnorm.v5 <- function(n){
  return(rnorm(n, 0.45, sqrt(0.0045)))
}

# generation of individual visit times
simRandomVisits <- list(list(fn=rmixnorm.v1,range=c(0.02,0.05)),
                    list(fn=rnorm.v2, range=c(0.02, 0.08)),
                    list(fn=rmixnorm.v3, range=c(0.04, 0.12)),
                    list(fn=rnorm.v4, range=c(0.10, 0.22)),
                    list(fn=rnorm.v5, range=c(0.12, 0.19)),
                    list(fn=rnorm.v5, range=c(0.12, 0.19)),
                    list(fn=rnorm.v5, range=c(0.12, 0.19)),
                    list(fn=rnorm.v5, range=c(0.12, 0.19)),
                    list(fn=rnorm.v5, range=c(0.12, 0.19)))

# definition of analysis paths
testpaths <- list(list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                       whichVisits=(1:6),
                       modelparams=list(covstr="RI", timestr="linear")),
                  list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                       whichVisits=(1:6),
                       modelparams=list(covstr="RIS", timestr="linear")),
                  list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                       whichVisits=(1:6),
                       modelparams=list(covstr="RICAR1", timestr="linear")),
                  list(isLongitudinal=FALSE, isMarkerComb=FALSE,
                       whichVisits=(1:6),
                       isSingleVisit=FALSE))
```

```
# definition of all simulation settings
sim.setup <- SimulationSetup(starModel=risStar, simreps=25, paths=testpaths,
                             nvisits=10, ngroups=2*c(95, 37),
                             grouplabels=c("non-resistant", "resistant"),
                             priors=c(292, 64)/356,
                             randomVisits=simRandomVisits,
                             fixVisits=NULL, nMCsamples=50, ratiotrain=2/3,
                             stepcut=0.01, bmtransform="log10")
```

The analysis of simulated data according to the specified paths is carried out as for real data by means of `fit`, `predict2` and `PerformanceMeasures` within a MCCV design by the already defined method `analyze` (cf. p. 76). However, for the `...StarModel`, the first step (the estimation of the group-specific group parameters) is skipped as the model was fully specified by the user and does not need to be fitted. The other two steps are executed to estimate the biomarker performance, which is achieved by the `...StarModel`, as this serves as direct benchmark for the fitted models. This is accomplished by the newly created method `analyzeStarModel`.

The user does not need to care about this distinction as just the method `analyze`, which has been adapted for objects of class `SimulationSetup`, is called to perform the analysis. The returned object `full.simtree` is of class `SimulationResult` and contains a list of objects of class `AnalysisSetup` and a list of objects of `ResultTree`, each of length `simreps`. Then the results are summarized over the MCCV samples by the method `mccvSummary` and afterwards over all simulation repetitions by `simSummary`:

```
full.simtree <- analyze(sim.setup)

mccv.sumsimtree <- mccvSummary(full.simtree, sim.setup)
sim.sumsimtree <- simSummary(mccv.sumsimtree)
```

The results of the subanalyses can be compared in the same way as for the analysis of real data by `selectResultLeafs` and `createReportFiles`. For this purpose, the output function `createReportFiles` has been adapted to the simulation setting, thus featuring options to assess the introduced bias when assuming an incorrect model structure.

Finally, the computation times are an issue. Suppose we go for 28 simulation repetitions, 50-fold MCCV, 4 model structures to be compared, biomarker profiles of length 6 and have 2 patient groups in a simulation. This results in $25 \cdot 50 \cdot 5 \cdot 2$=12500 models for one `...StarModel` and all computations took 56 minutes on a PC with an Intel Xeon X5450 3 GHz processor and 3 GB RAM. As the data analyses of simulation repetitions are independent, the computing performance may be enhanced by deploying multiple processors in parallel. In this example, one mother process spawns 7 parallel child processes which finished in 11 minutes. This implementation is based on the R package Rmpi which is one of the highly recommended packages by Schmidberger et al. (2009). The necessary configuration steps are described in the vignette "Parallel Computing in longQDA by Rmpi" of longQDA. The user-handling is easy. The required setup for the parallel computing is performed by the method parInit, followed by invoking the equivalent method of analyze for parallel computing which is parAnalyze. The parallel session is terminated by using parFinalize. Here is the syntax for an example with seven child processes:

```
parInit(nslaves=7, name="logOfSlave")
full.simtree <- parAnalyze(sim.setup)
parFinalize()
```

### Remark

Some ideas for the software design originated from a cooperation between Roche Diagnostics and Manuel J. A. Eugster and Friedrich Leisch of the Ludwig-Maximilians-University, Munich.

# 6 Conclusion

## 6.1 Summary

Discriminant analysis is a widely used statistical method for classification purposes. The extension for longitudinal data, proposed by Marshall and Barón (2000) and Tomasko et al. (1999), is achieved by plugging the marginal estimates of linear mixed models into the discriminant rule. This approach expands the search space for the best classification performance by a further dimension. Not only uni- or multivariate cross-sectional measurements may yield the highest performance, also a single or multiple longitudinal data profiles come into consideration. This moves the focus to modelling issues as model selection and the need for parsimonious parametrization that are less relevant issues for cross-sectional data. Besides those two research aims, an additional goal was a software implementation in R that fulfills quality criteria as user friendliness, fast extensibility and time efficiency. All those issues of the dissertation were motivated by and adopted to biomarker data from medical diagnostics. The proposed methods were applied to classify patients as resistant or non-resistant to a given therapy based on their longitudinal biomarker measurements.

First, the longitudinal quadratic discriminant analysis (longQDA), a classification method for longitudinal data, was reviewed. It is a two-step approach in that the longitudinal measurements are modelled by linear mixed models to yield empirical means and covariances for both classes. These estimations are then plugged into the discriminant rule known from quadratic discriminant analysis for cross-sectional data. To avoid an underestimation of the classification performance, Monte Carlo cross validation (MCCV) was used, performing the first step of longQDA on the training data and the second on the test data. For a summarized assessment of the classification performance, commonly used performance measures in diagnostic medicine as the area under the ROC curve (AUC) were complemented by the Brier score (BS) with its various decompositions to overcome their shortcomings to evaluate only the discrimination. The BS and its decompositions allow a more profound evaluation as they take the probability into account with which a person is classified and are not based on the less informative discretised membership to one of the classes. The application of the methodology to univariate biomarker data (RNA profiles) from patients that are and are not resistant to HIV therapy revealed an increased performance by a longitudinal instead of a cross-sectional design.

The high flexibility of linear mixed models induced the research question for an appropriate model selection criteria in longQDA, especially for not well separated classes. Compared to analysis goals where the mean parameters of the mixed model are of main interest, in longQDA, both, the mean and the covariance parameters are important for the discriminant rule. In Chapter 3, this issue was investigated by comparing two different selection criteria in a simulation study: One strategy was to choose the mixed model which had yielded the best classification performance, also considering by which performance measure the classification was assessed. The alternative strategy was to choose the mixed model with the smallest Bayesian information criterion (BIC). We examined linear mixed models with only a random intercept as true model, one with a random intercept and a random slope and one with a random intercept and a continuous AR(1) process for the residuals. All three structures and QDA were fitted and the length of the longitudinal profiles varied between 3 and 10. The simulation results showed that the BIC performed much better than the performance measures, selecting the correct mixed model structures in the broad majority of the MCCV loops. For most of the scenarios, especially those with shorter biomarker profiles, the performance measures were very similar providing no broad selection basis. Additionally, if they differed, only few of them as AUC, BS and the biserial correlation were able to select the correct model. As expected, the choice of the correct model structure becomes more and more important for models with more than about 5 visits and/or those that assume a time-variant longitudinal data structure. Then the misspecification effects do not only occur at the individual in form of incorrectly low or high classification uncertainty but also at the global assessment level in the form of underestimated performance measures.

In Chapter 4, the multivariate extension of longQDA was proposed. The challenge consisted in finding parsimonious multivariate mixed models. The subsequent proceeding of longQDA was the same as for the univariate case. The presentation was restricted to the bivariate case with two biomarkers. In the following section containing the outlook, it will become clear why this is an advisable starting point for any multivariate longQDA. Besides a fast ad-hoc solution for independent biomarkers, two multivariate mixed model classes were proposed in this dissertation. These are multivariate random effects models and covariance pattern models with a Kronecker product structure that may cope with cross-sectional correlations as well as the additional time-dependent cross-correlations in multivariate longitudinal data. Provided the data set is transformed appropriately, the estimation of the random effects models may be carried out with statistical software written for the univariate case. For the estimation of covariance pattern models, a numerical constraint optimization without the need of computer-intensive calculations for the information matrices was proposed. Not computing the information matrices and the parsimonious parametrization of Kronecker product models led to computational advantages. In the application to a diagnostic study with seven longitudinal biomarkers, assessed for resistance to one rheumatoid arthritis medication, the multivariate random coefficients models were preferred to the Kronecker product models according to the BIC. A gain in classification performance by pairs of biomarker profiles could not be achieved, a single longitudinal biomarker was the best option. However, exemplary simulated data settings were included to illustrate various potential gains through the combination of longitudinal biomarkers.

For a wide-spread application of statistical methods, an implementation in a state-of-the-art software environment like `R` is indispensable nowadays. The `R` package `longQDA` was therefore developed. Its application and implementation were presented in Chapter 5. The package provides a general framework for executing quadratic discriminant analysis with longitudinal data, including univariate, multivariate longQDA and multivariate QDA, with real as well as with simulated data. All models for longitudinal biomarker data that are proposed in this dissertation are implemented. Based on the resampling method Monte Carlo cross validation, the estimation of the group-specific parameters is performed with the training datasets and the evaluation of the classification performance with the test datasets. The software solution was created to be used for clinical biomarker studies with a longitudinal design at Roche Diagnostics. Therefore, conceptual requirements as a data-independent implementation with a user-friendly handling, easy extensibility, a good run-time performance and a comfortable reporting determined the design. The implementation of the software follows the modern object-orientated concept with S4 classes and comprises functionality for the entire data analysis process, from the descriptive and explorative analysis of longitudinal data up to the comparison of results under different model structures and data settings. Due to the broad generality, the key ideas may also serve as a source for the implementation of other statistical methods in `R`. The usefulness of the object-oriented approach in terms of fast extensibility was exemplarily demonstrated for the multivariate version of longQDA and for the generation and analysis of simulated data. For the latter, a short computational time was achieved by using parallel computing.

## 6.2 Outlook

Our presented approach for evaluating biomarker panels is not restricted to the bivariate case. Our results indicate that for biomarker variables reflecting complex biological processes, the restrictive proportionality property of Kronecker product structures is often too simplistic. When increasing the dimension of a biomarker panel, these strong implied assumptions get more and more implausible to hold. Therefore multivariate random effects models should be favored. Fieuws and Verbeke (2006) and Fieuws et al. (2007) propose an approach for $q > 2$ which extends directly the bivariate modelling presented above. As computational problems occur more frequently when increasing the dimension of the joint covariance matrix of the random effects, their approach consists of fitting all possible bivariate models and joining them afterwards by pseudo-likelihood arguments. Biomarker-specific parameters are estimated in more than one model and are therefore averaged. Pair-specific parameters as for example the covariance between two random effects are estimated in exactly one model and thus there is no need for averaging those parameters. Standard errors are obtained via a specifically constructed covariance matrix. The matrix contains entries of the pairwise information matrices, each weighted by the resulting coefficients from the averaging step.

There is a further point to consider with a longitudinal multivariate assessment. An optimization of the performance with respect to $p$, the number of measurements over time, has different implications than an optimization with respect to $q$, the dimension of the biomarker panel. Increasing $q$ yields a much higher dimensional problem than an increase in $p$. In the first mentioned univariate case, the dimension increases from $p$ to $p+1$ but it increases directly from $p \cdot q$ to $p \cdot (q+1)$ in the multivariate case. That is, combining longitudinal biomarkers is associated with a more challenging density estimation. Due to this rapid increase in dimension, the disadvantageous curse of dimensionality may become a serious problem. This is because the volume of a space increases exponentially by a linear increase in dimension, yielding mostly empty spaces. This leads to less precise plug-in estimators or even non-estimability. Van Ness (1976) finds in a comparison of several classification methods for cross-sectional data that QDA is especially sensitive to this phenomenon.

An alternative approach to longQDA might be based on functional data analysis. James and Hastie (2001) fit B-splines to univariate longitudinal biomarker data and plug the resulting mean and covariance functions into the standard discriminant rule. Multivariate measurements might be modelled by multivariate B-splines as Brown et al. (2005) did with biomarker data of the same HIV therapy resistance study that we used. As long as the groups are not very well separated, the classification performance might depend on the appropriateness of the group-specific models. Therefore, it is advisable to check in each application the sensitivity of the performance with respect to various modelling approaches.

Müller (2005) proposes functional principal component analysis as a dimension reduction technique for longitudinal data prior to the classification. The resulting functional principal component scores, derived from all patients, are then used as explanatory variables in a functional binary regression with the group membership as dependent variable. However, applying PCA prior to classification is not regarded as the best approach. The principal components represent the highest within-group variation whereas the discrimination aims at the highest between-group variation. This does not necessarily lead in the same direction (Jolliffe, 2002). These functional data approaches might be further investigated and contrasted with longQDA in the future.

# A Report for Dataset AIDS2

In Section 5.1, we illustrated the use of the R package `longQDA` for the therapy resistance data of HIV patients (The data served also as application dataset for the univariate longQDA in Chapter 2.). Here, we include the automatically generated report of our exemplary analysis[1]. **It serves as a quick overview of all the results and was not designed with the intention to be as beautifully formatted as a final report.** It contains all the results gathered throughout the analysis and may be a starting point when preparing a report of the results. The LaTeX format enables easy editing.

[1] The original format is one-sided.

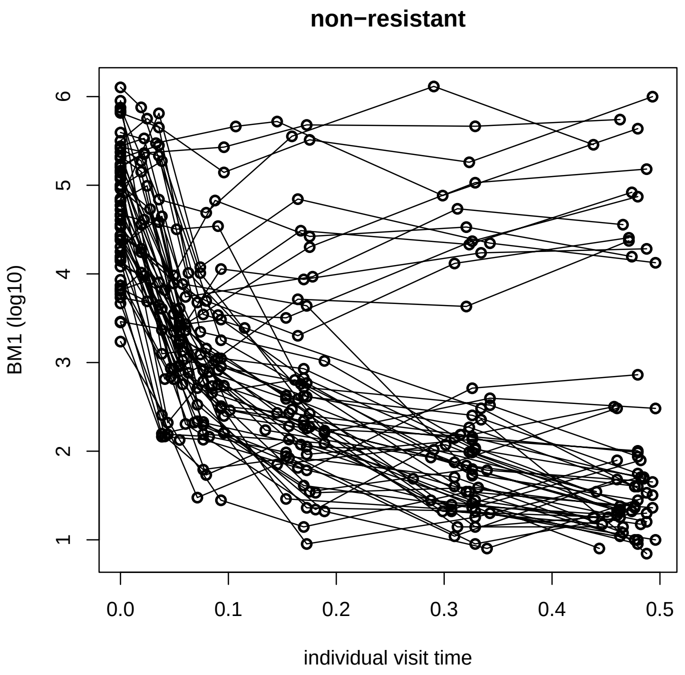
# Results of dataset AIDS2demo (complete cases)

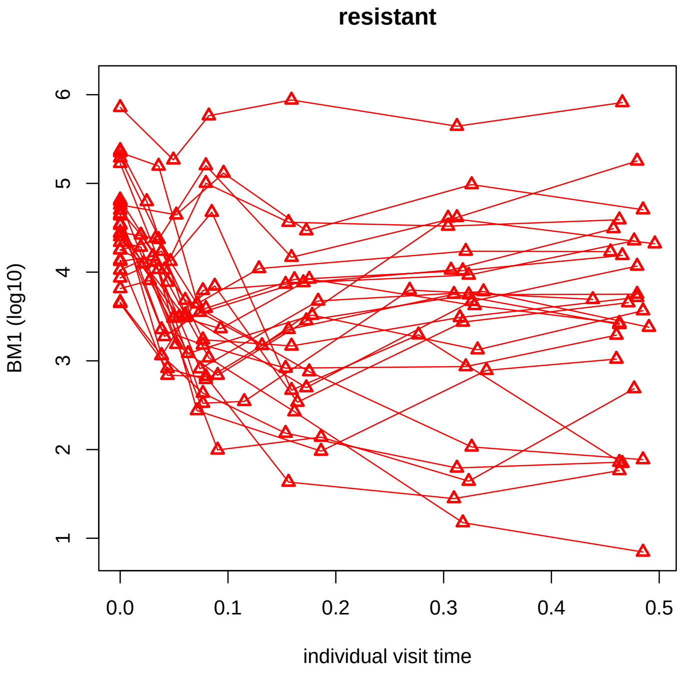
Mareike Kohlmann
DXRQB2
Roche Diagnostics GmbH

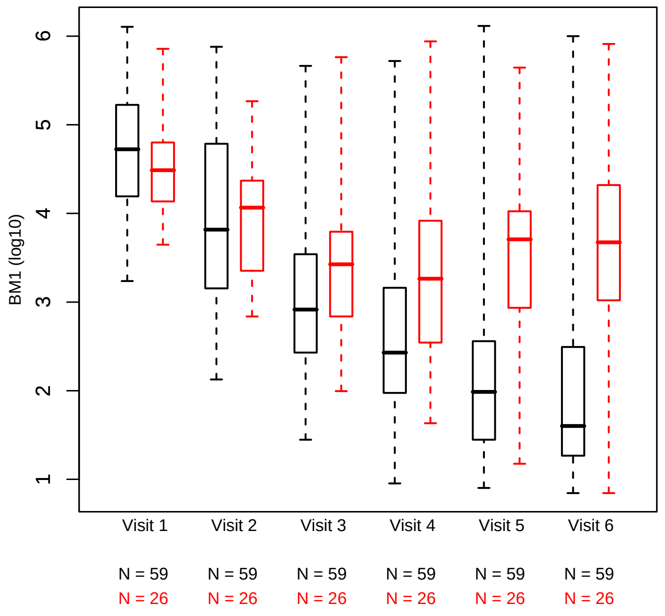
11th June 2008

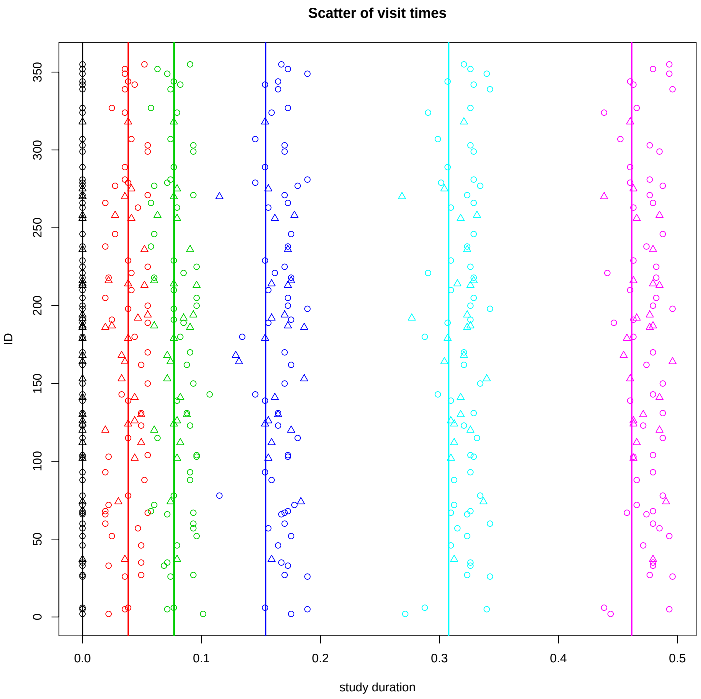
## 1 Exploratory Analysis

### 1.1 AIDS2demo (complete cases)









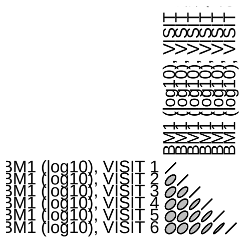
```
Summary of visit times for each visit:
$‘1‘
        non-resistant resistant
Min.                0         0
1st Qu.             0         0
Median              0         0
Mean                0         0
3rd Qu.             0         0
Max.                0         0

$‘2‘
        non-resistant resistant
Min.          0.01918   0.01918
1st Qu.       0.02603   0.03288
Median        0.03836   0.03836
Mean          0.03812   0.03793
3rd Qu.       0.04932   0.04384
Max.          0.05479   0.05479

$‘3‘
        non-resistant resistant
Min.          0.05753   0.06027
1st Qu.       0.07123   0.07397
Median        0.07945   0.07808
Mean          0.08033   0.07777
3rd Qu.       0.09315   0.08219
Max.          0.10680   0.09589

$‘4‘
        non-resistant resistant
Min.           0.1151    0.1151
1st Qu.        0.1575    0.1562
Median         0.1699    0.1603
Mean           0.1665    0.1615
3rd Qu.        0.1726    0.1726
Max.           0.1890    0.1863

$‘5‘
        non-resistant resistant
Min.           0.2712    0.2685
1st Qu.        0.3096    0.3103
Median         0.3260    0.3178
Mean           0.3199    0.3158
3rd Qu.        0.3288    0.3253
Max.           0.3425    0.3397
```

```
$‘6‘
        non-resistant resistant
Min.           0.4384    0.4384
1st Qu.        0.4630    0.4630
Median         0.4767    0.4685
Mean           0.4736    0.4712
3rd Qu.        0.4836    0.4795
Max.           0.4959    0.4959


Variance of visit times for each visit:
$‘1‘
non-resistant     resistant
            0             0

$‘2‘
non-resistant     resistant
 0.0001522672  0.0001006972

$‘3‘
non-resistant     resistant
 0.0001670600  0.0000937223

$‘4‘
non-resistant     resistant
 0.0001843052  0.0002903244

$‘5‘
non-resistant     resistant
 0.0002323249  0.0002483364

$‘6‘
non-resistant     resistant
 0.0002196071  0.0001705386
```

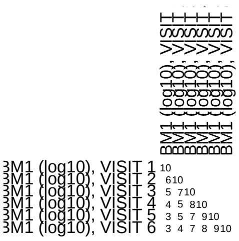
| VISIT 1 | VISIT 2 | VISIT 3 | VISIT 4 | VISIT 5 | VISIT 6 |
|---|---|---|---|---|---|
| 1.00 | 0.59 | 0.50 | 0.42 | 0.27 | 0.26 |
| 0.59 | 1.00 | 0.66 | 0.50 | 0.46 | 0.38 |
| 0.50 | 0.66 | 1.00 | 0.85 | 0.74 | 0.71 |
| 0.42 | 0.50 | 0.85 | 1.00 | 0.88 | 0.85 |
| 0.27 | 0.46 | 0.74 | 0.88 | 1.00 | 0.95 |
| 0.26 | 0.38 | 0.71 | 0.85 | 0.95 | 1.00 |

| VISIT 1 | VISIT 2 | VISIT 3 | VISIT 4 | VISIT 5 | VISIT 6 |
|---|---|---|---|---|---|
| 1.00 | 0.55 | 0.57 | 0.35 | 0.33 | 0.22 |
| 0.55 | 1.00 | 0.45 | 0.36 | 0.42 | 0.36 |
| 0.57 | 0.45 | 1.00 | 0.73 | 0.67 | 0.56 |
| 0.35 | 0.36 | 0.73 | 1.00 | 0.83 | 0.79 |
| 0.33 | 0.42 | 0.67 | 0.83 | 1.00 | 0.92 |
| 0.22 | 0.36 | 0.56 | 0.79 | 0.92 | 1.00 |

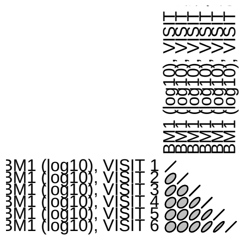
BM1 (log10), non−resistant



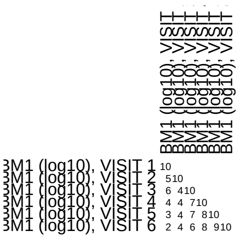
BM1 (log10), non−resistant



BM1 (log10), resistant



BM1 (log10), resistant



## 2 Analysis Setup

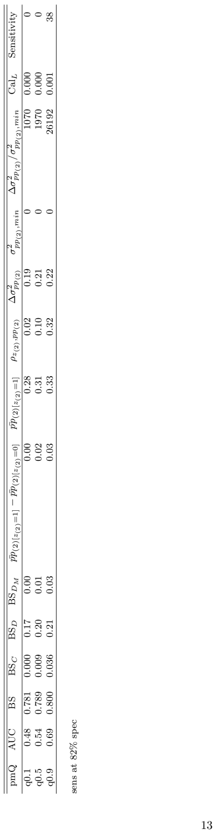
```
Formal class 'AnalysisSetup' [package "longQDA"] with 4 slots
  ..@ data                      :Formal class 'LongData' [package "longQDA"] with 17 slots
  .. .. ..@ id2rows          : int [1:85, 1:6] 1 7 13 19 25 31 37 43 49 55 ...
  .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ : chr [1:85] "2" "5" "6" "26" ...
  .. .. .. .. ..$ : chr [1:6] "1" "2" "3" "4" ...
  .. .. ..@ visitindex       : int 3
  .. .. ..@ timeindex        : int 6
  .. .. ..@ markerlabels     : Named chr "BM1 (log10)"
  .. .. .. ..- attr(*, "names")= chr "BM1"
  .. .. ..@ markerindices    : int 5
  .. .. ..@ scheduledtimes   : num [1:6] 0.0000 0.0385 0.0769 0.1538 0.3077 ...
  .. .. ..@ rawdata          :'data.frame': 510 obs. of  8 variables:
  .. .. .. ..$ PATID   : int [1:510] 2 2 2 2 2 2 5 5 5 5 ...
  .. .. .. ..$ TIMEDAY : int [1:510] 0 8 37 64 99 162 0 13 26 69 ...
  .. .. .. ..$ VISIT   : int [1:510] 1 2 3 4 5 6 1 2 3 4 ...
  .. .. .. ..$ CENSOR  : int [1:510] 1 1 1 1 1 3 1 1 1 1 ...
  .. .. .. ..$ BM1     : num [1:510] 5.24 4.61 2.46 2.27 1.69 ...
  .. .. .. ..$ SDURN   : num [1:510] 0.0000 0.0219 0.1014 0.1753 0.2712 ...
  .. .. .. ..$ GROUP   : int [1:510] 1 1 1 1 1 1 1 1 1 1 ...
  .. .. .. ..$ GROUPVAR: Factor w/ 2 levels "non-resistant",..: 1 1 1 1 1 1 1 1 1 1 ...
  .. .. ..@ name             : chr "AIDS2demo (complete cases)"
  .. .. ..@ groupindex       : int 7
  .. .. ..@ grouplabels      : chr [1:2] "non-resistant" "resistant"
  .. .. ..@ groupcodes       : num [1:2] 1 2
  .. .. ..@ groupcolors      : chr [1:2] "black" "red"
  .. .. ..@ groupsymbols     : num [1:2] 1 2
  .. .. ..@ grouppriors      : num [1:2] 0.694 0.306
  .. .. ..@ fixedgrouppriors : logi FALSE
  .. .. ..@ idindex          : int 1
  .. .. ..@ markertransformfn: chr "log10"
  ..@ MCsamples                 :List of 10
  .. ..$ :List of 2
  .. .. ..$ train: int [1:58] 2 5 6 33 35 52 57 68 72 88 ...
  .. .. ..$ test : int [1:27] 26 27 37 46 60 66 67 74 78 102 ...
  .. ..$ :List of 2
  .. .. ..$ train: int [1:58] 6 26 27 33 35 37 46 52 57 60 ...
  .. .. ..$ test : int [1:27] 2 5 66 88 104 115 123 126 139 141 ...
  .. ..$ :List of 2
  .. .. ..$ train: int [1:58] 5 6 27 33 37 46 52 57 60 66 ...
  .. .. ..$ test : int [1:27] 2 26 35 72 74 88 104 139 141 143 ...
  .. ..$ :List of 2
  .. .. ..$ train: int [1:58] 2 5 6 26 27 35 60 66 67 72 ...
  .. .. ..$ test : int [1:27] 33 37 46 52 57 68 74 123 130 131 ...
```

```
.. ..$ :List of 2
.. .. ..$ train: int [1:58] 2 5 6 26 27 33 35 37 52 57 ...
.. .. ..$ test : int [1:27] 46 72 74 115 120 139 162 164 170 179 ...
.. ..$ :List of 2
.. .. ..$ train: int [1:58] 2 6 26 27 33 37 52 57 60 66 ...
.. .. ..$ test : int [1:27] 5 35 46 88 104 112 131 143 162 187 ...
.. ..$ :List of 2
.. .. ..$ train: int [1:58] 2 5 26 27 33 35 37 52 57 66 ...
.. .. ..$ test : int [1:27] 6 46 60 67 68 74 78 88 104 115 ...
.. ..$ :List of 2
.. .. ..$ train: int [1:58] 2 5 6 26 27 33 37 52 60 66 ...
.. .. ..$ test : int [1:27] 35 46 57 68 78 102 112 153 164 168 ...
.. ..$ :List of 2
.. .. ..$ train: int [1:58] 2 6 26 27 33 35 46 52 57 60 ...
.. .. ..$ test : int [1:27] 5 37 72 78 104 112 126 162 186 189 ...
.. ..$ :List of 2
.. .. ..$ train: int [1:58] 2 5 6 26 27 35 37 46 57 60 ...
.. .. ..$ test : int [1:27] 33 52 66 74 78 88 120 131 139 143 ...
..@ paths                      :List of 2
.. ..$ :Formal class 'AnalysisPath' [package "longQDA"] with 4 slots
.. .. .. ..@ whichVisitsModel:List of 1
.. .. .. .. ..$ : num 1
.. .. .. ..@ modelfn         :Formal class 'QdaAlgo' [package "longQDA"] with 1 slots
.. .. .. .. .. ..@ reshapefn: chr "toXsec"
.. .. .. ..@ modelparams     : list()
.. .. .. ..@ biomarker       :List of 1
.. .. .. .. ..$ : chr "BM1"
.. ..$ :Formal class 'AnalysisPath' [package "longQDA"] with 4 slots
.. .. .. ..@ whichVisitsModel:List of 1
.. .. .. .. ..$ : int [1:6] 1 2 3 4 5 6
.. .. .. ..@ modelfn         :Formal class 'RisAlgo' [package "longQDA"] with 1 slots
.. .. .. .. .. ..@ reshapefn: chr "minimize"
.. .. .. ..@ modelparams     :List of 2
.. .. .. .. ..$ covstr : chr "RIS"
.. .. .. .. ..$ timestr: chr "quadratic"
.. .. .. ..@ biomarker       :List of 1
.. .. .. .. ..$ : chr "BM1"
..@ performanceMeasuresParams:List of 1
.. ..$ stepcut: num 0.01
```

## 3 Analysis

### 3.1 Path 1 Visit(s) 1 Markers BM1

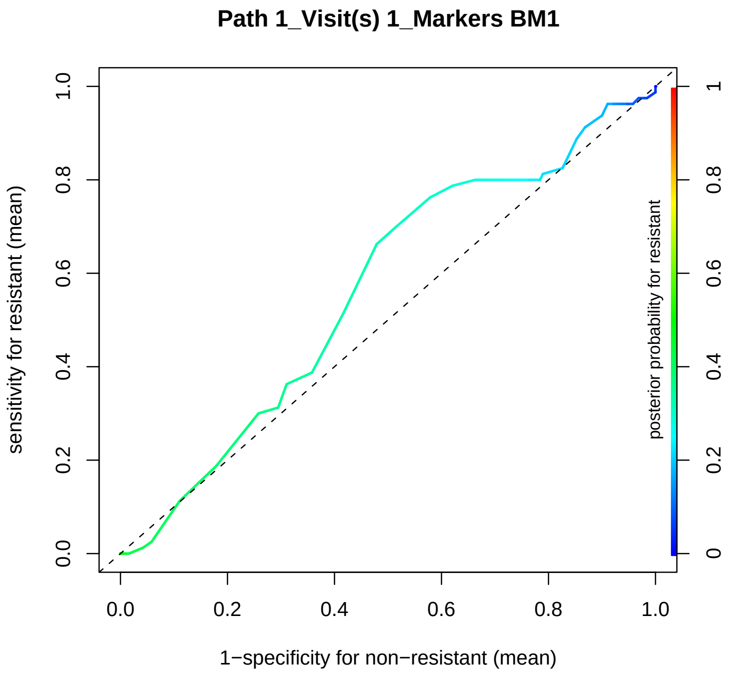
```
$group1
  mu1   V11
4.702 0.479

$group2
  mu1   V11
4.518 0.324
```

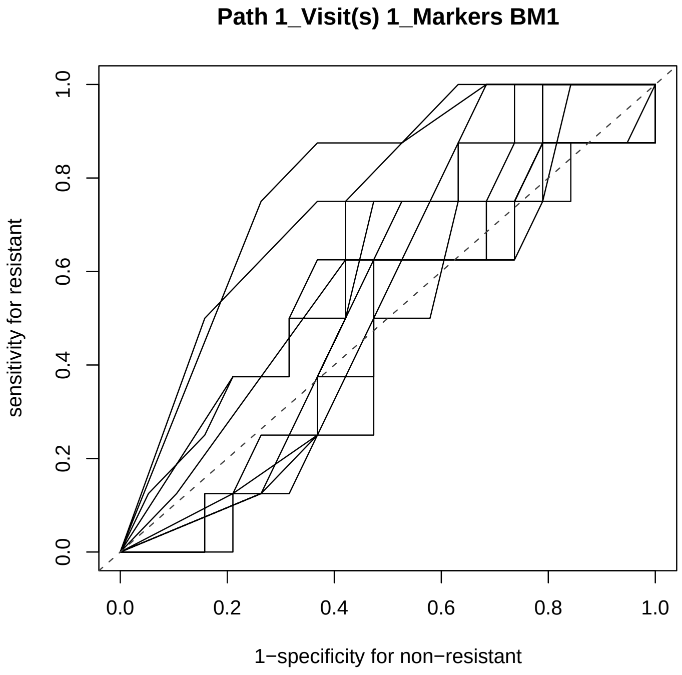


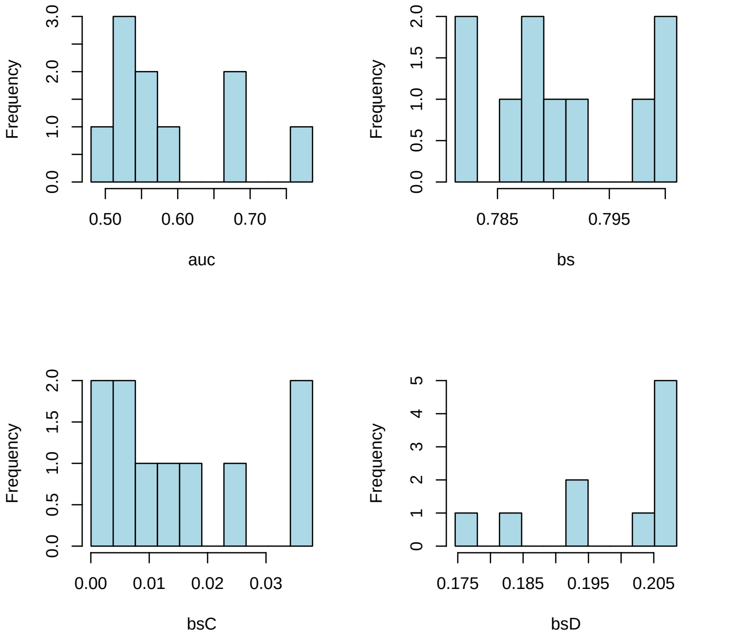
| pmQ | AUC | BS | $\mathrm{BS}_C$ | $\mathrm{BS}_D$ | $\mathrm{BS}_{D_M}$ | $\bar{pp}_{(2)[z_{(2)}=1]} - \bar{pp}_{(2)[z_{(2)}=0]}$ | $\bar{pp}_{(2)[z_{(2)}=1]}$ | $\rho_{z_{(2)},pp_{(2)}}$ | $\Delta\sigma^2_{pp_{(2)}}$ | $\sigma^2_{pp_{(2)},min}$ | $\Delta\sigma^2_{pp_{(2)}}/\sigma^2_{pp_{(2)},min}$ | $\mathrm{Cal}_L$ | Sensitivity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| q0.1 | 0.48 | 0.781 | 0.000 | 0.17 | 0.00 | 0.00 | 0.28 | 0.02 | 0.19 | 0 | 1070 | 0.000 | 0 |
| q0.5 | 0.54 | 0.789 | 0.009 | 0.20 | 0.01 | 0.02 | 0.31 | 0.10 | 0.21 | 0 | 1970 | 0.000 | 0 |
| q0.9 | 0.69 | 0.800 | 0.036 | 0.21 | 0.03 | 0.03 | 0.33 | 0.32 | 0.22 | 0 | 26192 | 0.001 | 38 |

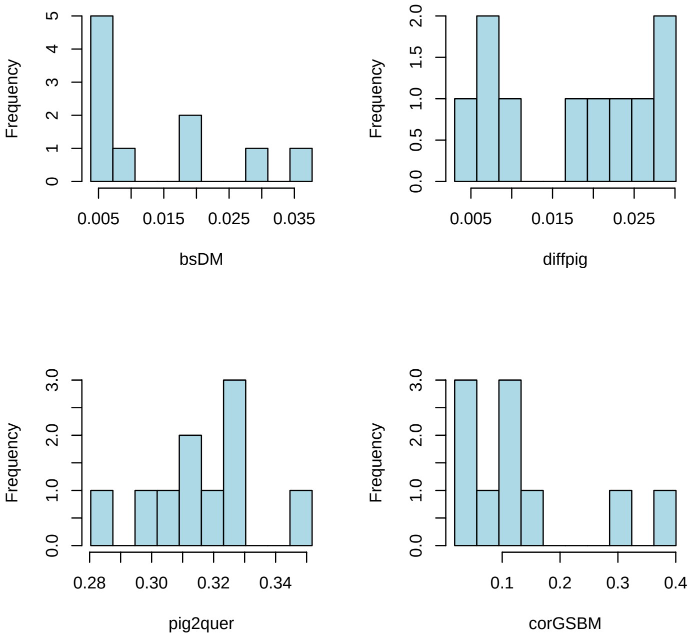
sens at 82% spec

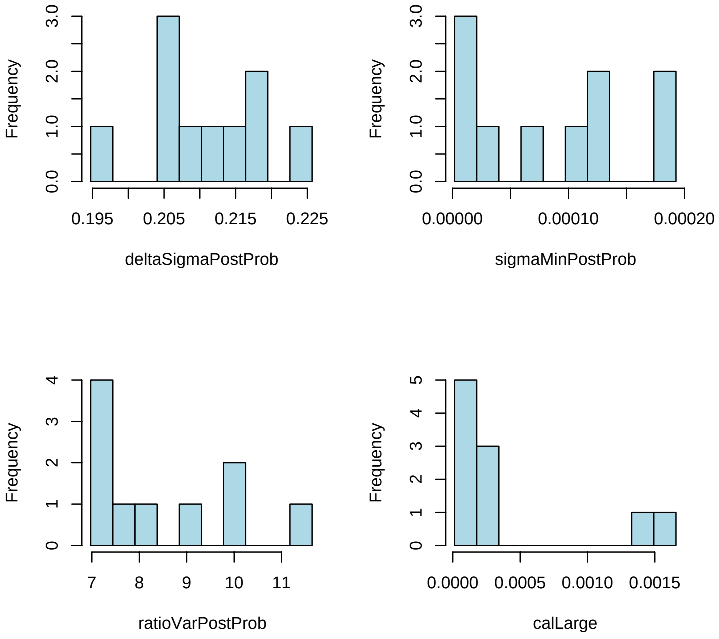
13

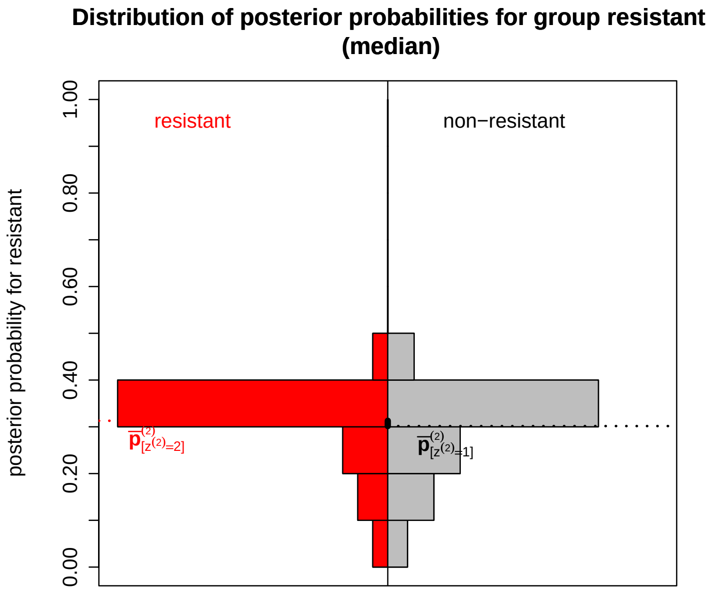


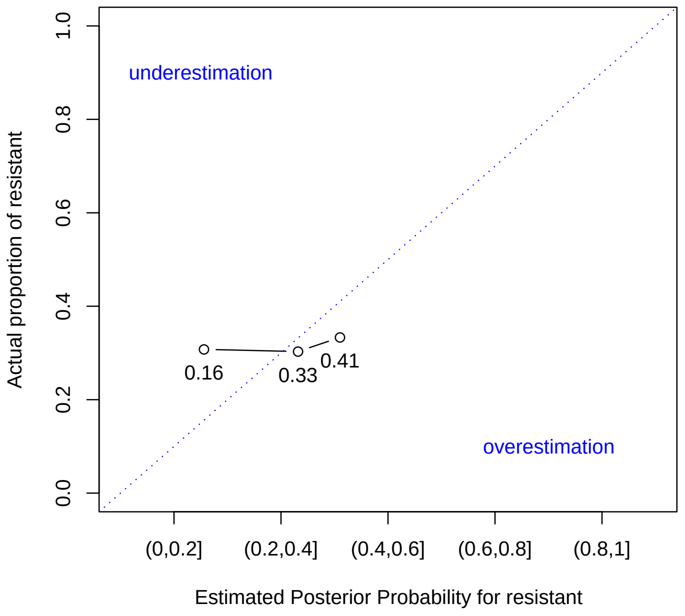





Frequency
bsDM
0.005 0.015 0.025 0.035
diffpig
0.005 0.015 0.025
pig2quer
0.28 0.30 0.32 0.34
corGSBM
0.1 0.2 0.3 0.4


Frequency
deltaSigmaPostProb
0.195 0.205 0.215 0.225
sigmaMinPostProb
0.00000 0.00010 0.00020
ratioVarPostProb
7 8 9 10 11
calLarge
0.0000 0.0005 0.0010 0.0015


Distribution of posterior probabilities for group resistant (median)
posterior probability for resistant
0.00 0.20 0.40 0.60 0.80 1.00
resistant
non−resistant
$\bar{p}^{(2)}_{[z^{(2)}=2]}$
$\bar{p}^{(2)}_{[z^{(2)}=1]}$

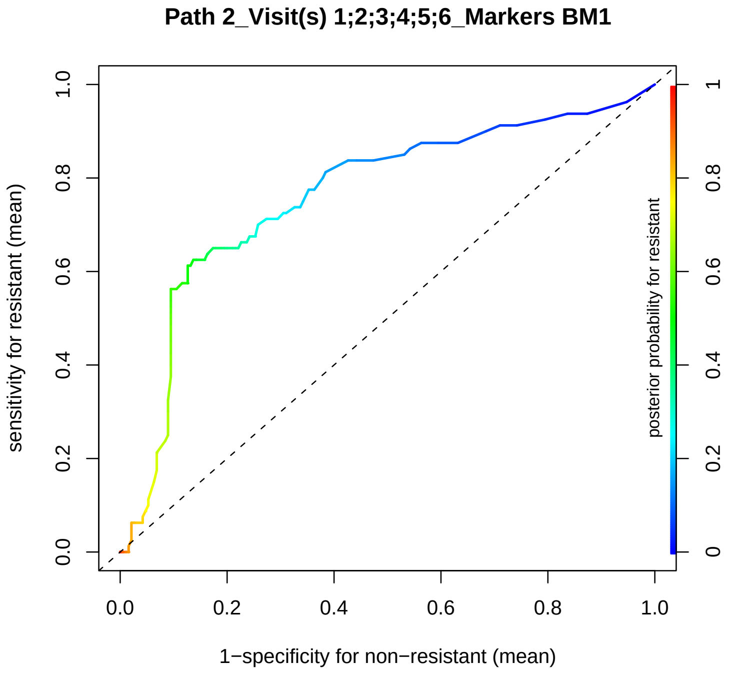

Actual proportion of resistant
0.0 0.2 0.4 0.6 0.8 1.0
underestimation
overestimation
0.16
0.33
0.41
(0,0.2] (0.2,0.4] (0.4,0.6] (0.6,0.8] (0.8,1]
Estimated Posterior Probability for resistant

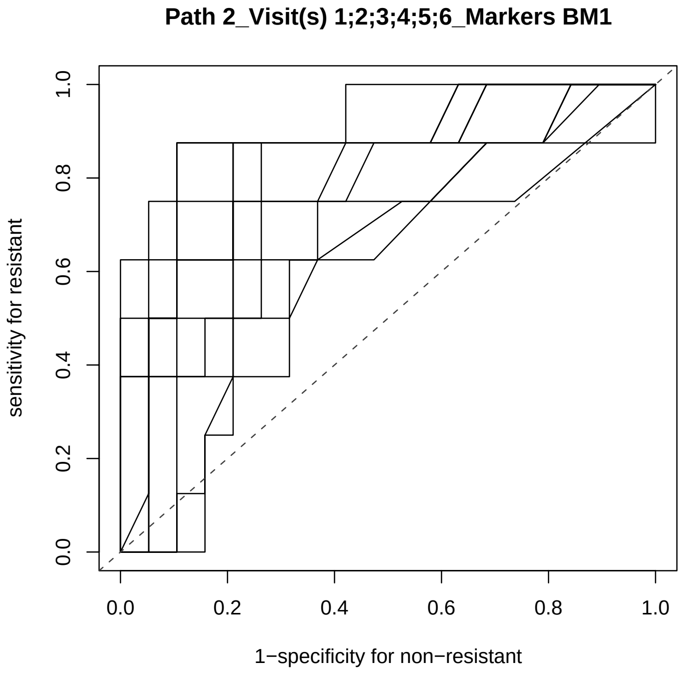
### 3.2 Path 2 Visit(s) 1;2;3;4;5;6 Markers BM1

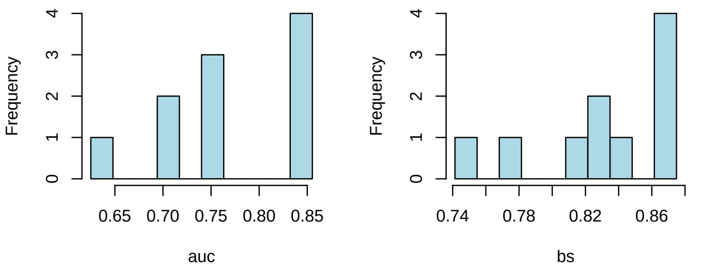
```
$group1
(Intercept)       SDURN  I(SDURN^2)        DInt      DSlope        D_IS
      4.438     -13.172      18.863       0.479       8.286       0.315
   sigQuad
      0.314

$group2
(Intercept)       SDURN  I(SDURN^2)        DInt      DSlope        D_IS
      4.270      -7.908      13.557       0.290       5.109       0.203
   sigQuad
      0.263
```

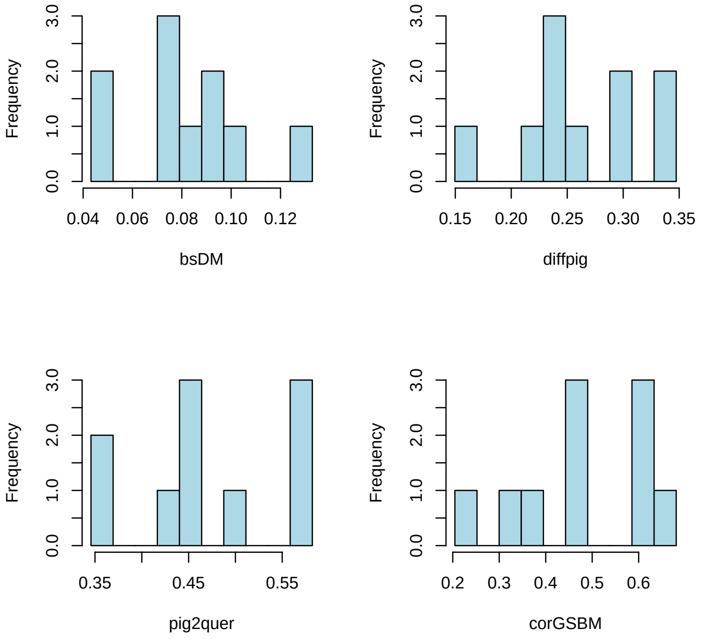
| pmQ | AUC | BS | BS$_C$ | BS$_D$ | BS$_{D_M}$ | $\bar{pp}_{(2)[z_{(2)}=1]} - \bar{pp}_{(2)[z_{(2)}=0]}$ | $\bar{pp}_{(2)[z_{(2)}=1]}$ | $\rho_{z_{(2)},pp_{(2)}}$ | $\Delta\sigma^2_{pp_{(2)}}$ | $\sigma^2_{pp_{(2)},min}$ | $\Delta\sigma^2_{pp_{(2)}}/\sigma^2_{pp_{(2)},min}$ | Cal$_L$ | Sensitivity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| q0.1 | 0.62 | 0.741 | 0.010 | 0.08 | 0.04 | 0.15 | 0.35 | 0.20 | 0.14 | 0.005 | 6 | 0.000 | 38 |
| q0.5 | 0.76 | 0.832 | 0.035 | 0.13 | 0.07 | 0.25 | 0.45 | 0.47 | 0.18 | 0.013 | 14 | 0.004 | 62 |
| q0.9 | 0.86 | 0.871 | 0.054 | 0.16 | 0.10 | 0.34 | 0.58 | 0.62 | 0.22 | 0.024 | 21 | 0.010 | 88 |

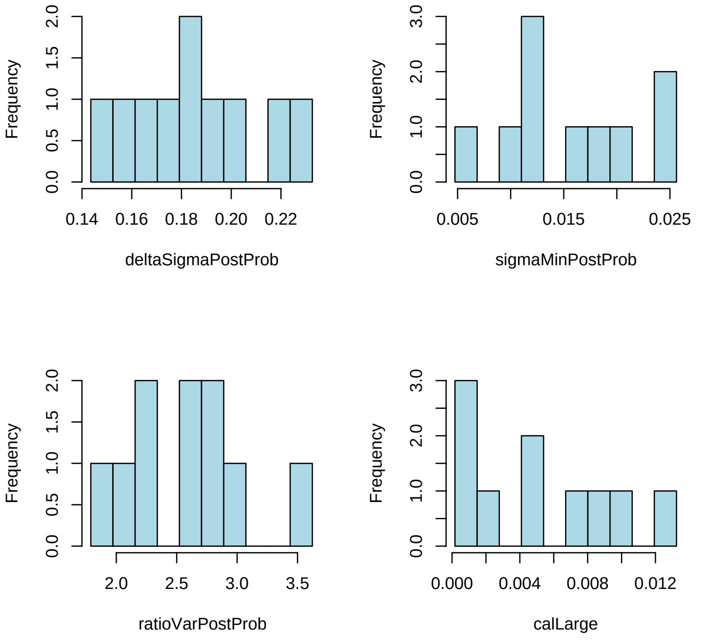
sens at 81% spec



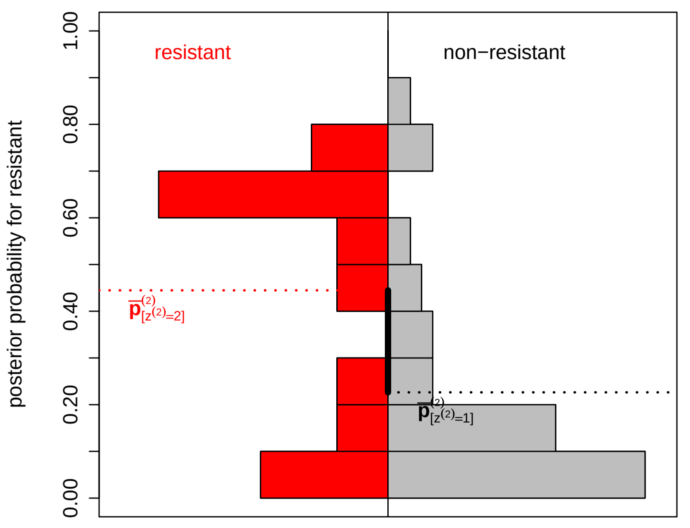


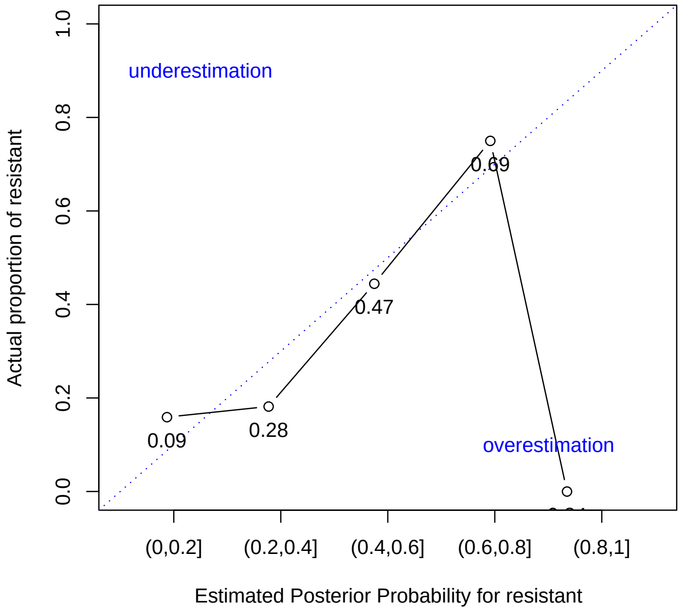






**Distribution of posterior probabilities for group resistant (median)**

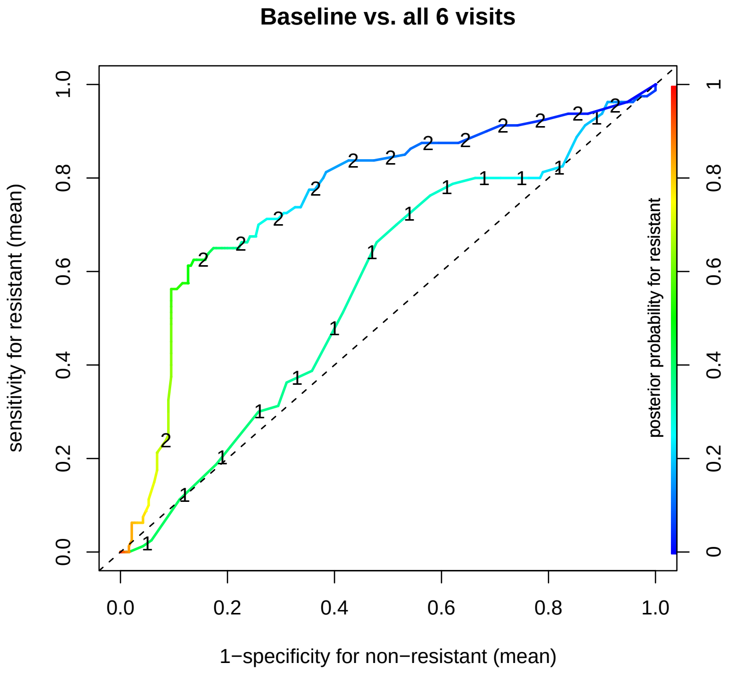


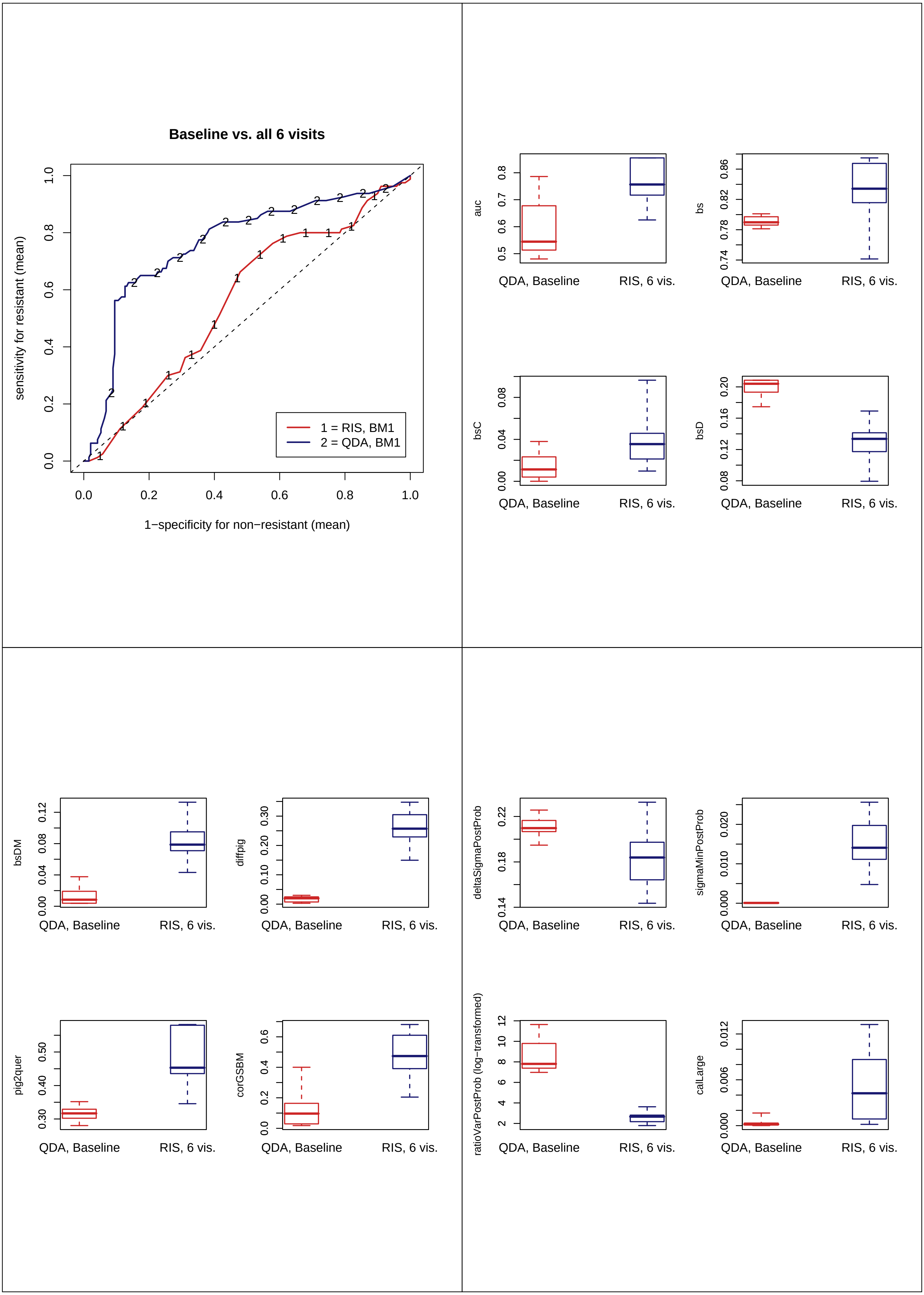


longQDA includes 6 biomarker measurements per patient.

## 4 Comparisons

### 4.1 Baseline vs. all 6 visits




Baseline vs. all 6 visits
sensitivity for resistant (mean)
1−specificity for non−resistant (mean)
0.0
0.2
0.4
0.6
0.8
1.0
1 = RIS, BM1
2 = QDA, BM1
auc
bs
bsC
bsD
bsDM
diffpig
deltaSigmaPostProb
sigmaMinPostProb
pig2quer
corGSBM
ratioVarPostProb (log−transformed)
calLarge
QDA, Baseline
RIS, 6 vis.

# B Selected Help Files of the R Package longQDA

As supplementary material to Chapter 5, the documentation for the most important functions are included in the form of help files. The following selection comprises those functions that are needed to perform a "standard" longQDA, containing the output as predefined in `createReportFiles`. Beside the class `MccvSummary` and the method `createReportFiles` of which the help files were included in Subsection 5.2.2, the classes `Report`, `LongData`, `AnalysisSetup` and the methods `analyze`, `mccvSummary` and `selectResultLeafs` are required.

---

`Report-class` *Class 'Report' Contains Paths of Output Files*

---

### Description

An object of class 'Report' contains a list with elements called "Exploratory Analysis", "Analysis Setup", "Analysis" and "Comparisons" as well as the path of the report folder where all output files are stored. Every list entry contains the paths of the output files which have been generated in the corresponding part of the analysis. Graphics are saved in pdf format, tables and text in tex format.

### Objects from the Class

Objects are created by calls of the form `Report(folder)`.

`folder`: Object of class `'character'`. Path of the folder where all analysis output is saved which is later included in the report. If the folder does not exist yet, it is automatically created. If the folder exists, its content is updated.

### Slots

`.Data`: Object of class `'list'` which contains the file names of the plots and tables to be part of the report, structured by the following list elements:

- `expl`: Output for report section "Exploratory Analysis" - further subsections possible.
- `anasetup`: Output for report section "Analysis Setup".
- `analysis`: Output for report section "Analysis" - further subsections possible.
- `comp`: Output for report section "Comparisons" - further subsections possible.

(All subsections are created within `createReportFiles` with the result that the corresponding list entries above may contain again lists (as documented above).)

`folder`: Object of class `'character'`: Path of the report folder where analysis output to be included is saved.

### Extends

Class `'list'`, from data part. Class `'vector'`, by class `'list'`, distance 2.

### Methods

**createReport** `signature(report = 'Report')`: `createReport`

**save2Report** `signature(report = 'Report')`: `save2Report`

### Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

### See Also

`save2Report` to add plots or text (comments) to the report, `createReportFiles` to generate pre-defined output for one analysis step and `createReport` to create the report in tex (and pdf) format.

### Examples

```
showClass("Report")

myreport <- Report(folder=getwd())
```

| LongData-class | *Class 'LongData' containing longitudinal data and metainformation* |
|---|---|

## Description

Contains the longitudinally structured data set as well as metainformation (e.g. group-specific colors and column indices of marker variables). For each subject, the data set contains so many rows as there are individual repeated measurements.

## Objects from the Class

Objects are created by calls of the form `LongData(rawdata, name, markerlabels, scheduledtimes, groups, priors, visitname, timename, groupname, groupcolors, groupsymbols, idname)`.

`rawdata`: Object of class `'data.frame'`. Raw data set in longitudinal structure, read in beforehand by `read.table`, for example.

`name`: Object of class `'character'`. Name of the data set, used i.a. as name of the folder where for the output of the analysis is stored. As it will also be displayed in the header of the report file in tex format, the character string should *not* contain '_'.

`markerlabels`: Object of class `'character'`. Names of the columns in `rawdata` containing the biomarker variables.

`scheduledtimes`: Object of class `'numeric'`. Vector containing scheduled times of the visits.

`groups`: Object of class `'numeric'`. Vector with levels of group variable. The group labels need to be provided as names of the vector.

`priors`: Object of class `'numeric'`. Default: `c(0,0)`. Prior probabilities for group membership. If they are not specified by the user (default), they are estimated by the group proportions present in the data. Otherwise, they are not estimated and taken as fixed throughout the analysis.

`visitname`: Object of class `'character'`. Default: `"VISIT"`. Name of the variable in the data set containing the consecutively numbered visits.

`timename`: Object of class `'character'`. Default: `"TIME"`. Name of the variable in the data set containing the continuous study time (i.e. the actual time since baseline).

`groupname`: Object of class `'character'`. Default: `"GROUP"`. Name of the variable in the dataset containing the numerically coded group membership according to the Gold Standard.

groupcolors: Object of class 'character'. Default: c("black", "red"). Colours that are used for plots to distinguish the two groups by color.

groupsymbols: Object of class 'numeric'. Default: c(1,2). Symbol numbers for plotting points to distinguish between the two groups. For possible values, refer to pch in the section "Details" of points.

idname: Object of class 'character'. Default: "PATID". Name of the variable in the data set containing the unique numerical subject identifiers.

## Slots

id2rows: Object of class 'matrix' containing the row numbers of the data set for each subject and each visit. The names of the rows are the identification numbers of the subjects, the names of the columns are the visit numbers.

visitindex: Object of class 'numeric'. Column of the data set containing the visits (see also visitname above).

timeindex: Object of class 'numeric'. Column of the data set containing the study time (see also timename above).

markerlabels: Object of class 'character'. Vector containing the names of the marker labels provided by the user. The names of the vector are the new standardized names of the markers in the data set which are "BM1", "BM2",...

markerindices: Object of class 'numeric'. Vector containing the columns of the data set containing the marker variables.

scheduledtimes: Object of class 'numeric'. Vector containing the scheduled times of the visits as provided by the user.

rawdata: Object of class 'data.frame'. Data set provided by the user in a standardized structure, e.g. coded names of the biomarker variables (see markerlabels).

name: Object of class 'character'. Name of the data set provided by the user.

groupindex: Object of class 'numeric'. Column of the data set containing the coded group membership (see also groupname above).

grouplabels: Object of class 'character'. Labels of the two groups, provided by the user.

groupcodes: Object of class 'numeric'. Coded levels of the two groups, provided by the user.

groupcolors: Object of class 'character'. V. groupcolors above.

groupsymbols: Object of class 'numeric'. V. groupsymbols above.

grouppriors: Object of class 'numeric'. V. priors above.

fixedgrouppriors: Object of class 'logical'. If TRUE, prior probabilities for each were provided by the user and are therefore treated as fixed. If FALSE, the priors are estimated by the group proportions in the data set and re-estimated whenever necessary (e.g. in completeCases).

idindex: Object of class 'numeric'. Column of the data set containing the unique subject identification numbers (see also idname above).

markertransformfn: Object of class 'character'. Names of the functions used for marker transformation, only filled after execution of transform for real data.

### Extends

Class 'Data', directly.

### Methods

autoCorr: signature(data = 'LongData'): autoCorr

boxplotLongMarkers: signature(data = 'LongData'): boxplotLongMarkers

completeCases: signature(data = 'LongData'): completeCases

createReportFiles: signature(object = 'LongData'): createReportFiles

fit: signature(algo = 'QdaAlgo', data = 'LongData'): fit,QdaAlgo, LongData-method

fit: signature(algo = 'RiAlgo', data = 'LongData'): fit,RiAlgo, LongData-method

fit: signature(algo = 'RisAlgo', data = 'LongData'): fit,RisAlgo, LongData-method

fit: signature(algo = 'Ricar1Algo', data = 'LongData'): fit,Ricar1Algo, LongData-method

minimize: signature(data = 'LongData'): minimize

plotLongMarkers: signature(data = 'LongData'): plotLongMarkers

predict2: signature(model = 'QdaModel', data = 'LongData'): predict2,QdaModel,LongData-method

predict2: signature(model = 'LongDaUnivModel', data = 'LongData'): predict2,LongDaUnivModel,LongData-method

scatterVisitTimes: signature(data = 'LongData'): scatterVisitTimes

selectIds: signature(data = 'LongData', ids = 'vector'): selectIds

selectVisitsMarkers: signature(data = 'LongData', visits = 'vector', markers = 'vector'): selectVisitsMarkers

toLongmv: signature(data = 'LongData'): toLongMv

toXsec: signature(data = 'LongData'): toXsec

transform: signature(_data = 'LongData'): transform

updateColIndices: signature(data = 'LongData'): updateColIndices

updateId2Rows: signature(data = 'LongData'): updateId2Rows

### Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

### Examples

```
showClass("LongData")

data(AIDS2)
d <- LongData(AIDS2, "AIDS2", c("BM1"), c(0,2,4,8,16,24)/52,
              c('non-resistant'=1, 'resistant'=2), timename="SDURN")
```

## AnalysisSetup-class *Class 'AnalysisSetup' Containing the Analysis Settings*

### Description

All analysis settings are contained therein to enable logging and reproducibility. In details, these are the data together with its metainformation, the Monte Carlo cross-validation (MCCV) samples, the analysis paths containing details about sub-analyses to be performed and the global precision parameters for evaluating the performance measures. To compare results of various analyses, it is recommended to use the same setup settings (the same data set, same MCCV samples and same performance measures parameters), differing only with regard to the analysis paths defining the models. This is accomplished by saving the analysis setup of one analysis and updating it for the other one by `updateAnalysisSetup`.

### Objects from the Class

Objects are created by calls of the form
`AnalysisSetup(data, nMCsamples, ratiotrain, paths, stepcut)`.

`data`: Object of class '`LongData`'. Data set and corresponding metainformation.

`nMCsamples`: Object of class '`numeric`'. Number of Monte Carlo crossvalidation samples to be used for the analysis.

`ratiotrain`: Object of class '`numeric`'. Subjects are randomized blockwise (i.e. with all their repeated measurements) to the training set with a probability of `ratiotrain` within each group. This sampling yields a ratio of the size of the training to that of the test set equal to `ratiotrain` and is performed without replacement.

paths: Object of class 'list'. The list should contain the following knot selection criteria to build up the analysis tree describing the sub-analyses to be performed. For an illustration with 2 biomarkers and 3 visits see the vignette "longQDA-Analysis".

> isLongitudinal: Object of class 'logical'. If TRUE, longitudinal data structure is accounted for in the analysis, i.e. longQDA is used.
>
> isMarkerComb: Object of class 'logical'. If TRUE, biomarkers are analyzed in combinations.
>
> markerCombLength: Object of class 'numeric' determining how many markers are combined; only needed for multivariate (long)QDA, i.e. when isMarkerComb=TRUE.
>
> isSingleVisit: Object of class 'logical'. If TRUE, data of single visits are used; only needed for classic QDA.
>
> whichVisits: Object of class 'numeric' giving the vector of visits which are included in the analysis. For longQDA, this vector should always have 1 as first entry and should be a consecutive sequence.
>
> modelparams: Object of class 'list' containing the following mixed model parameters and need to be therefore only provided by the user if isLongitudinal=TRUE.
>
> > covstr: Object of class 'character' giving the covariance structure. Can be either "RI" (random intercept), "RIS" (random intercept and linear slope) or "RICAR1" (random intercept and continuous AR(1) residual structure) for univariate longQDA.
> >
> > timestr: Object of class 'character' giving the maximum order for the fixed effect(s) of the time variable. Terms of lower order are automatically included. Can either be "linear" or "quadratic".

stepcut: Object of class 'numeric'. Default: 0.01. Precision parameter for generating the ROC curve. It gives the step length for moving the cut-off along the range of the predicted posterior probabilities. Lower values than the default gives higher precision and vice versa.

## Slots

data: Object of class 'LongData'. Data set and corresponding metainformation as provided by the user. It is not modified, only saved to support reproducible research.

MCsamples: Object of class 'list' having length nMCsamples and containing for each MC sample a list with 2 entries:

> train: Object of class 'numeric' containing the unique identification numbers of subjects sampled to the training set by MCCV.
>
> test: Object of class 'numeric' containing the unique identification numbers of subjects sampled to the test set by MCCV.

paths: Object of class 'list' containing the analysis paths of class 'AnalysisPath'. See there for details and their structure.

performanceMeasuresParams: Object of class 'list' containing the named precision parameter stepcut. (Note: The list structure was set up to allow further extension with additional parameters in the future.)

### Methods

analyze: signature(analysis = 'AnalysisSetup'): analyze

createReportFiles: signature(object = 'AnalysisSetup'): createReportFiles

updateAnalysisSetup: signature(setup = 'AnalysisSetup'): updateAnalysisSetup

### Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

### See Also

'AnalysisPath'

### Examples

```
showClass("AnalysisSetup")

data(AIDS2)
d <- LongData(AIDS2, "AIDS2", c("BM1"), c(0,2,4,8,16,24)/52,
             c('non-resistant'=1, 'resistant'=2), timename="SDURN")
d2 <- completeCases(d, visits=6)
asetup <- AnalysisSetup(d2, nMCsamples=5, ratiotrain=2/3,
               paths=list(list(isLongitudinal=FALSE, isMarkerComb=FALSE,
                               isSingleVisit=TRUE,
                               whichVisits=(1:ncol(d2@id2rows))),
                          list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                               whichVisits=(1:ncol(d2@id2rows)),
                               modelparams=list(covstr="RI",
                                                timestr="quadratic"))))
```

`analyze-method` *Performs All (long)QDA Analyses and Evaluates Classification Performance*

### Description

Every analysis comprises the following sequential steps: Data set preparation, groupwise model fitting by the appropriate algorithm on the training set, computation of the marginal estimators of the mean vector and the covariance matrix to predict the posterior probabilities by the quadratic discriminant rule on the test set and graphical and numerical evaluation of the performance measures. This is done repeatedly using each of the MCCV samples, each visit and each biomarker selection of the analysis paths defined by the user in the analysis resp. simulation setup.

For simulated data, this procedure is repeated for each of the `simreps` generated data sets. To speed up the analyses, parallel computing can be used. See `parAnalyze` for details.

### Usage

**For objects of class `'AnalysisSetup'` or `'SimulationSetup'`:**

```
analyze(analysis)
```

### Arguments

`analysis = 'AnalysisSetup'`**:** Object defining all analyses to be performed with real data.

`analysis = 'SimulationSetup'`**:** Object defining all analyses to be performed with simulated data.

### Value

Object of class `'ResultTree'` for real data resp. of class `'SimulationResult'` for simulated data.

### Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

### See Also

`'AnalysisSetup'`, `'ResultTree'`,`'SimulationSetup'`, `'SimulationResult'`

## Examples

```
### Using real data:
data(AIDS2)
d <- LongData(AIDS2, "AIDS2", c("BM1"), c(0,2,4,8,16,24)/52,
             c('non-resistant'=1, 'resistant'=2), timename="SDURN")
d2 <- completeCases(d, visits=6)
asetup <- AnalysisSetup(d2, nMCsamples=5, ratiotrain=2/3,
                        paths=list(
                              list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                                   whichVisits=(1:ncol(d2@id2rows)),
                                   modelparams=list(covstr="RI",
                                                    timestr="quadratic"))))
full.tree <- analyze(asetup)
str(full.tree, 8)

### Using simulated data (without making use of parallel computing):
risStar <- RisStarModel(group1=list(beta=c(4.0,-4.5),
                          DCovstar=matrix(c(0.40, 0.39, 0.39, 16.1),
                                   ncol=2, byrow=FALSE),
                          R=c(sigQuad=0.56)),
                        group2=list(beta=c(4.0,-1.4),
                          DCovstar=matrix(c(0.23, 0.46, 0.46, 15.8),
                                   ncol=2, byrow=FALSE),
                          R=c(sigQuad=0.39)),
                        timestr="linear")

testpaths <- list(list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                       whichVisits=(1:6),
                       modelparams=list(covstr="RIS", timestr="linear")))

sim.setup <- SimulationSetup(starModel=risStar, simreps=2, paths=testpaths,
                             nvisits=6, ngroups=2*c(95, 37),
                             grouplabels=c("non-resistant", "resistant"),
                             priors=c(292, 64)/356, randomVisits=NULL,
                             fixVisits=c(0,2,4,8,16,24)/52,
                             nMCsamples=5, ratiotrain=2/3, stepcut=0.01,
                             bmtransform="log10")

full.simtree <- analyze(sim.setup)
```

## mccvSummary-method *Summarizes Analysis Results over MCCV samples*

### Description

The lowest level of the analysis tree, consisting of one node for each MCCV sample, is summarized as one leaf node.

### Usage

**For objects of class** ’ResultTree’ **or** ’SimulationResult’**:**

```
mccvSummary(tree, analysis)
```

### Arguments

tree = ’ResultTree’**:** Tree with analysis results based on real data. The leaf nodes with the results of the MCCV samples are summarized.

tree = ’SimulationResult’**:** Tree with analysis results based on simulated data. The leaf nodes with the results of the MCCV samples are summarized.

analysis**:** Object of class ’AnalysisSetup’ containing metainformation about the data. It is needed for a correctly structured summary of the analysis tree.

### Value

Object of class ’ResultTree’ with leaf nodes of class ’MccvSummary’.

### Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

### Examples

```
### Example 1 with real data (results of class 'ResultTree')
data(AIDS2)
d <- LongData(AIDS2, "AIDS2", c("BM1"), c(0,2,4,8,16,24)/52,
            c('non-resistant'=1, 'resistant'=2), timename="SDURN")
d2 <- completeCases(d, visits=6)
asetup <- AnalysisSetup(d2, nMCsamples=5, ratiotrain=2/3,
                        paths=list(
                              list(isLongitudinal=FALSE, isMarkerComb=FALSE,
                                   isSingleVisit=TRUE,
                                   whichVisits=(1:ncol(d2@id2rows))),
                              list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                                   whichVisits=(1:ncol(d2@id2rows)),
                                   modelparams=list(covstr="RI",
                                                    timestr="quadratic"))))
tree <- analyze(asetup)
sum.tree <- mccvSummary(tree, asetup)
```

## selectResultLeafs-method *Selects Analysis Results of Specified Node Leafs*

### Description

Selects some or all of the analysis results specified by the user for comparison. In addition, the complex hierarchical structure of the result tree is removed and a flat list is returned instead. For an easy identification of the selected paths, it is recommended to provide path names.

### Usage

**For objects of class** `'ResultTree'` **or** `'SimulationResult'`**:**

```
selectResultLeafs(tree, leafnumbers=NULL, compname=paste("leafnumbers:",
ifelse(is.null(leafnumbers), "all", paste(leafnumbers,
collapse="-"))), pathnames=NULL)
```

### Arguments

`tree = 'ResultTree'`**:** A tree containing all results of the analysis based on real data in the leaf nodes.

`tree = 'SimulationResult'`**:** A tree containing all results of the analysis based on simulated data in the leaf nodes.

`leafnumbers`**:** Default: `NULL`. Vector with numbers of leaf nodes to be selected, can be displayed by `showLeafs`. If no leafnumbers are specified (default), all leafnumbers are selected.

`compname`**:** Default: `paste("leafnumbers:", ifelse(is.null(leafnumbers), "all", paste(leafnumbers, collapse="-")))`. Name of the comparison, used as name for the returned result list.

`pathnames`**:** Default: `NULL`. Names of the corresponding analysis paths of the selected leaf nodes, used as list names of the returned result list.

### Note

The result tree must have the structure as returned by `mccvSummary`. Otherwise, an error message is generated and the execution of the function is stopped.

### Value

Object of class `'ResultList'` containing only the analysis results of selected leafs.

## Author(s)

Mareike Kohlmann ⟨longQDA@web.de⟩

## Examples

```
data(AIDS2)
d <- LongData(AIDS2, "AIDS2", c("BM1"), c(0,2,4,8,16,24)/52,
             c('non-resistant'=1, 'resistant'=2), timename="SDURN")
d2 <- completeCases(d, visits=6)
asetup <- AnalysisSetup(d2, nMCsamples=5, ratiotrain=2/3,
                        paths=list(
                              list(isLongitudinal=FALSE, isMarkerComb=FALSE,
                                   isSingleVisit=TRUE,
                                   whichVisits=(1:ncol(d2@id2rows))),
                              list(isLongitudinal=TRUE, isMarkerComb=FALSE,
                                   whichVisits=(1:ncol(d2@id2rows)),
                                   modelparams=list(covstr="RI",
                                                    timestr="quadratic"))))
tree <- analyze(asetup)
summary.tree <- mccvSummary(tree, asetup)

leafs6 <- selectResultLeafs(summary.tree, leafnumbers=c(1,7),
                        compname="Baseline vs. all 6 visits",
                        pathnames=c("QDA, Baseline", "RI, 6 vis."))
```

# Bibliography

Bates, D., Maechler, M., Dai, B., 2008. *lme4: Linear mixed-effects models using S4 classes*. `http://cran.r-project.org/web/packages/lme4/`. R package version 0.99875-9.

Beckett, L. A., Tancredi, D. J., Wilson, R. S., 2004. Multivariate longitudinal models for complex change processes. *Statistics in Medicine* 23, 231–239.

Born, M., Holz, K., Kath, O., 2004. *Softwareentwicklung mit UML 2: Die "neuen" Entwurfstechniken UML 2, MOF 2 und MDA*. Addison-Wesley.

Brant, L. J., Sheng, S. L., Morrell, C. H., Verbeke, G. N., Lesaffre, E., Carter, H. B., 2003. Screening for prostate cancer by using random-effects models. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 166, 51–62.

Brown, E. R., Ibrahim, J. G., DeGruttola, V., 2005. A flexible B-spline model for multiple longitudinal biomarkers and survival. *Biometrics* 61, 64–73.

Chakraborty, H., Helms, R., Sen, P., Cohen, M., 2003. Estimating correlation by using a general linear mixed model: evaluation of the relationship between the concentration of HIV-1 RNA in blood and semen. *Statistics in Medicine* 22, 1457–1464.

Chambers, J., 1998. *Programming with Data: A Guide to the S Language*. Springer Verlag.

Chambers, J., 2006. How S4 methods work. `http://developer.r-project.org/howMethodsWork.pdf`.

De la Cruz-Mesia, R., Marshall, G., 2006. Non-linear random effects models with continuous time autoregressive errors: a Bayesian approach. *Statistics in Medicine* 25, 1471–1484.

Diamond, G. A., 1992. What price perfection? Calibration and discrimination of clinical prediction models. *Journal of Clinical Epidemiology* 45, 85–89.

Doran, H., Lockwood, J. R., 2006. Fitting value-added models in R. *Journal of Educational and Behavioral Statistics* 31, 205–230.

Dutilleul, P., 1999. The MLE algorithm for the matrix normal distribution. *Journal of Statistical Computation and Simulation* 64, 105–123.

Efron, B., Tibshirani, R. J., 1993. *An Introduction to the Bootstrap*. Vol. 57 of Monographs on Statistics and Applied Probability. CRC Press, Boca Raton.

Felson, D. T., Anderson, J. J., Boers, M., Bombardier, C., Chernoff, M., Fried, B., Furst, D., Goldsmith, C., Kieszak, S., Lightfoot, R., et al., 1993. The American College of Rheumatology preliminary core set of disease activity measures for rheumatoid arthritis clinical trials. *Arthritis & Rheumatism* 36 (6), 729–740.

Fieuws, S., Verbeke, G., 2004. Joint modelling of multivariate longitudinal profiles: pitfalls of the random-effects approach. *Statistics in Medicine* 23, 3093–3104.

Fieuws, S., Verbeke, G., 2006. Pairwise fitting of mixed models for the joint modeling of multivariate longitudinal profiles. *Biometrics* 62, 424–431.

Fieuws, S., Verbeke, G., Molenberghs, G., 2007. Random-effects models for multivariate repeated measures. *Statistical Methods in Medical Research* 16, 387–397.

Fridman, V., 2005. *LineStats 1.1*. DoWhatever Software, `http://www.linestats.com-about.com`.

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

Galecki, A. T., 1994. General-class of covariance-structures for 2 or more repeated factors in longitudinal data-analysis. *Communications in Statistics-Theory and Methods* 23, 3105–3119.

Gerds, T. A., Cai, T., Schumacher, M., 2008. The performance of risk prediction models. *Biometrical Journal* 50, 457–479.

Gneiting, T., Raftery, A. E., 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102, 359–378.

Goedman, R., Grothendieck, G., Højsgaard, S., Pinkus, A., 2008. *Ryacas: R interface to the yacas computer algebra system*. `http://cran.r-project.org/web/packages/Ryacas/`. R package version 0.2-7.

Goldstein, H., 2003. *Multilevel Statistical Models*, 3rd Edition. Arnold, London.

Gradstein, M., 1986. Maximal correlation between normal and dichotomous variables. *Journal of Educational Statistics* 11, 259–261.

Greven, S., 2007. *Non-Standard Problems in Inference for Additive and Linear Mixed Models*, Dissertation, Ludwig-Maximilians-Universität, München. Cuvillier Verlag, Göttingen.

Gueorguieva, R., 2001. A multivariate generalized linear mixed model for joint modelling of clustered outcomes in the exponential family. *Statistical Modelling* 1, 177–193.

Gurka, M. J., 2006. Selecting the best linear mixed model under REML. *The American Statistician* 60, 19–26.

Gurney, J. W., Swensen, S. J., 1995. Solitary pulmonary nodules: determining the likelihood of malignancy with neural network analysis. *Radiology* 196, 823–829.

Hamlett, A., Ryan, L., Serrano-Trespalacios, P., Wolfinger, R., 2003. Mixed models for assessing correlation in the presence of replication. *Journal of the Air & Waste Management Association* 53, 442–450.

Hand, D. J., 1997. *Construction and assessment of classification rules*. Wiley, New York.

Hand, D. J., 2001. Measuring diagnostic accuracy of statistical prediction rules. *Statistica Neerlandica* 55, 3–16.

Harrell, F. E., 2006. How to avoid overfitting in training optimization with respect to area under ROC. `http://sci4um.com/about33215.html`.

Harville, D., 1977. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association* 72, 320–338.

Hippokrates, 300 B.C. The book of prognostics. Translated by F. Adams, `http://classics.mit.edu/Hippocrates/prognost.1.1.html`.

Hogan, J. W., Roy, J., Korkontzelou, C., 2004. Tutorial in biostatistics: Handling drop-out in longitudinal studies. *Statistics in Medicine* 23, 1455–1497.

Ihaka, R., Gentleman, R., 1996. R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5, 299–314.

Ikeda, M., Ishigaki, T., Yamauchi, K., 2002. Relationship between Brier score and area under the binormal ROC curve. *Computer Methods and Programs in Biomedicine* 67, 187–194.

James, G. M., Hastie, T. J., 2001. Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, 533–550.

Johnson, R. A., Wichern, D., 2002. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Englewood Cliffs, NY.

Jolliffe, I. T., 2002. *Principal Component Analysis*, 2nd Edition. Springer, NY.

Kohlmann, M., 2005. Classifying Disease Progression by Longitudinal Biomarkers in the Absence of a Gold Standard. Master Thesis, Ludwig-Maximilians-Universität, München.

Littell, R. C., Pendergast, J., Natarajan, R., 2000. Modelling covariance structure in the analysis of repeated measures data. *Statistics in Medicine* 19, 1793–1819.

Lockwood, J. R., Doran, H., McCaffrey, D., 2003. Using R for estimating longitudinal student achievement models. R *News* 3, 17–23.

Lu, N., Zimmerman, D., 2005. The likelihood ratio test for a separable covariance matrix. *Statistics and Probability Letters* 73, 449–457.

Maechler, M., Gentleman, R., 2008. *classGraph: Construct Graphs of S4 Class Hierarchies*. `http://cran.r-project.org/web/packages/classGraph/`. R package version 0.7-1.

Marshall, G., Barón, A. E., 2000. Linear discriminant models for unbalanced longitudinal data. *Statistics in Medicine* 19, 1969–1981.

May, S., DeGruttola, V., 2007. Nonparametric tests for two-group comparisons of dependent observations obtained at varying time points. *Biometrics* 63, 194–200.

McLachlan, G. J., 1992. *Discriminant analysis and statistical pattern recognition*. Wiley, New York.

Mickey, R. M., Shema, S. J., Vacek, P. M., Bell, D. Y., 1994. Analysis of multiple outcome variables measured longitudinally. *Computational Statistics & Data Analysis* 17, 17–33.

Mitchell, M. W., Genton, M. G., Gumpertz, M. L., 2006. A likelihood ratio test for separability of covariances. *Journal of Multivariate Analysis* 97, 1025–1043.

Müller, H. G., 2005. Functional Modelling and Classification of Longitudinal Data. *Scandinavian Journal of Statistics* 32, 223–240.

Munoz, A., Carey, V., Schouten, J. P., Segal, M., Rosner, B., 1992. A parametric family of correlation structures for the analysis of longitudinal data. *Biometrics* 48, 733–742.

Murphy, A. H., 1973. A new vector partition of the probability score. *Journal of Applied Meteorology* 12, 595–600.

O'Brien, L. M., Fitzmaurice, G. M., 2005. Regression models for the analysis of longitudinal gaussian data from multiple sources. *Statistics in Medicine* 24, 1725–1744.

Patterson, H. D., Thompson, R., 1971. Recovery of inter-block information when block sizes are unequal. *Biometrika* 58, 545–554.

Pepe, M. S., 2003. *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press, New York.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., R Development Core Team, 2008. *nlme: Linear and Nonlinear Mixed Effects Models*. `http://cran.r-project.org/web/packages/nlme/`. R package version 3.1-86.

R Development Core Team, 2008. R: *A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0. `http://www.R-project.org`.

R Foundation for Statistical Computing, 2008. Ruml. `http://www.r-project.org/SoC08/Ruml.pdf`.

Roy, A., Khattree, R., 2003. Tests for mean and covariance structures relevant in repeated measures based discriminant analysis. *Journal of Applied Statistical Science* 12, 91–104.

Sanders, F., 1963. On subjective probability forecasting. *Journal of Applied Meteorology* 2, 191–201.

Schafer, J. L., Yucel, R. M., 2002. Computational strategies for multivariate linear mixed-effects models with missing values. *Journal of Computational and Graphical Statistics* 11, 437–457.

Schmidberger, M., Morgan, M., Eddelbuettel, D., Yu, H., Tierney, L., Mansmann, U., 2009. State-of-the-art in Parallel Computing with R. Technical Report No. 47, Department of Statistics, University of Munich, `http://epub.ub.uni-muenchen.de/8991/1/parallelR_techRep.pdf`.

Shah, A., Laird, N., Schoenfeld, D., 1997. Random-effects model for multiple characteristics with possibly missing data. *Journal of the American Statistical Association* 92, 775–779.

Shao, J., 1993. Linear model selection by cross-validation. *Journal of the American Statistical Association* 88, 486–494.

Singer, J. D., Willett, J. B., 2003. *Applied Longitudinal Data Analysis*. Oxford University Press, New York.

SparksSystems, 2008. *Enterprise Architect*.

Spiegelhalter, D. J., 1986. Probabilistic prediction in patient management and clinical trials. *Statistics in Medicine* 5, 421–433.

Taylor, J., 2005. Scale parameter modelling of the t-distribution. Dissertation, University of Adelaide, `http://hdl.handle.net/2440/37745`.

Taylor, J. M. G., Cumberland, W. G., Sy, J. P., 1994. A stochastic model for analysis of longitudinal aids data. *Journal of the American Statistical Association* 89, 727–736.

Thiébaut, R., Jacqmin-Gadda, H., Babiker, A., Commenges, D., 2005. Joint modelling of bivariate longitudinal data with informative dropout and left-censoring, with application to the evolution of CD4+ cell count and HIV RNA viral load in response to treatment of HIV infection. *Statistics in Medicine* 24, 65–82.

Thiébaut, R., Jacqmin-Gadda, H., Chêne, G., Leport, C., Commenges, D., 2002. Bivariate linear mixed models using SAS proc MIXED. *Computer Methods and Programs in Biomedicine* 69, 249–256.

Timm, N. H., 2002. *Applied Multivariate Analysis*. Springer, New York.

Tomasko, L., Helms, R. W., Snapinn, S. M., 1999. A discriminant analysis extension to mixed models. *Statistics in Medicine* 18, 1249–1260.

Van Ness, J., 1976. On the effects of dimension in discriminant analysis for unequal covariance populations. *Technometrics* 18, 119–127.

Venables, W., Ripley, B., Hornik, K., Gebhardt, A., 2008. *VR: Bundle of MASS, class, nnet, spatial*. `http://cran.r-project.org/web/packages/VR/`. R package version of MASS 7.2-40.

Verbeke, G., Molenberghs, G., 2000. *Linear Mixed Models for Longitudinal Data*. Springer, New York.

Verbyla, A., 1990. A conditional derivation of residual maximum likelihood. *Australian & New Zealand Journal of Statistics* 32, 227–230.

Weiss, R. E., 2005. *Modeling Longitudinal Data*. Springer, New York.

Wernecke, K. D., Kalb, G., Schink, T., Wegner, B., 2004. A mixed model approach to discriminant analysis with longitudinal data. *Biometrical Journal* 46, 246–254.

Werner, K., Jansson, M., Stoica, P., 2008. On estimation of covariance matrices with Kronecker product structure. *IEEE Transactions on Signal Processing* 56, 478–491.

Yates, J., 1982. External correspondence: Decompositions of the mean probability score. *Organizational Behavior and Human Performance* 30, 132–156.

yWorks, 2008. *yEd Graph Editor*. `http://www.yworks.com/en/products_yed_about.html`.

Zhou, X. H., Obuchowski, N. A., McClish, D. K., 2002. *Statistical Methods in Diagnostic Medicine*. Wiley & Sons, New York.

Zolg, J. W., Langen, H., 2004. How industry is approaching the search for new diagnostic markers and biomarkers. *Molecular and Cellular Proteomics* 3, 345–354.

# Lebenslauf

**Mareike Kohlmann**

geboren am 21.6.1980 in Bad Mergentheim

## Ausbildung

| | |
|---|---|
| 1990 - 1999 | Deutschorden-Gymnasium in Bad Mergentheim |
| 10/1999 - 05/2005 | Statistik-Studium mit Anwendungsgebiet Medizinische Biometrie an der Ludwig-Maximilians-Universität München: |
| 10/2003 | Bachelor of Science |
| 05/2005 | Master of Science |
| 07/2002 - 06/2003 | Statistik-Studium an der University of Auckland, Neuseeland: |
| 06/2003 | Postgraduate Diploma in Science |
| seit 07/2005 | Promotionsstudium in Statistik zum Dr. rer. nat. an der Ludwig-Maximilians-Universität München |

## Auszeichnungen

| | |
|---|---|
| 03/2006 | Bernd-Streitberg-Preis der Internationalen Biometrischen Gesellschaft Deutsche Region für die Master Thesis |

## Praktische Tätigkeiten

| | |
|---|---|
| 09/2000 - 02/2001 | Werkstudentin beim Biometrischen Zentrum für Therapiestudien GmbH, München |
| 03/2001 - 02/2002 | Studentische Hilfskraft am Sylvia Lawry Centre for Multiple Sclerosis Research und am Institut für medizinische Statistik und Epidemiologie der Technischen Universität München |
| 04/2004 - 06/2004 | Statistische Beratung für das Zentrum für naturheilkundliche Forschung der Technischen Universität München |
| 07/2004 - 08/2008 | Projektarbeit für Roche Diagnostics GmbH, Penzberg |
| 07/2005 - 08/2008 | wissenschaftliche Mitarbeiterin am Institut für Statistik, Ludwig-Maximilians-Universität München |
| 09/2008 - 04/2010 | Senior Statistical Consultant bei STAT-UP, München |
| seit 05/2010 | Research Scientist bei BASF SE, Ludwigshafen |